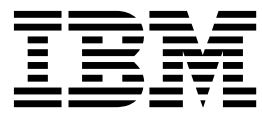IBM Security Identity Manager
Version 6.0.0.18
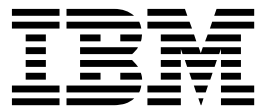
Configuration Topics

IBM

IBM Security Identity Manager
Version 6.0.0.18

Configuration Topics

IBM

Table of contents

Table list

Chapter 1. User interface customization overview

Many customers want a simple user interface for their employees to interact with IBM® Security Identity Manager to do basic management and provisioning functions. IBM Security Identity Manager provides multiple user interfaces that are customizable and provide the basic IBM Security Identity Manager functions that are needed by both users and administrators.

Interface customization options that IBM Security Identity Manager provides give customers the control and flexibility to manage how IBM Security Identity Manager functions are presented to their employees. With these options, customers can integrate a self-service user interface, a service center interface, and an administrative console interface into their intranet website and maintain a common corporate appearance.

Self-service user interface customization

The IBM Security Identity Manager self-service user interface is highly customizable. You can integrate a common corporate appearance while they maintain the flexibility to do self-care identity management tasks integral to their roles and responsibilities.

You can define and customize the self-service interface in two ways, by using the built-in console framework and by directly modifying files that are installed within IBM Security Identity Manager:

- Built-in console features
  - Access control items (ACIs)
  - Views
- Modifiable files
  - Properties files
  - Cascading style sheet (CSS) files
  - A subset of Java™ server pages (JSP) files
  - Image files

Back up any modifiable files for recovery purposes before you make customization changes to IBM Security Identity Manager.

Configuration files and descriptions

Configuration files define the appearance of the IBM Security Identity Manager self-service user interface.

The following tables list the file names and describe their roles in the customization of IBM Security Identity Manager.

1

Table 1. Property configuration files and descriptions

File Name	File Description
SelfServiceUI.properties	• Controls the layout of the user interface (banner, footer, navigation bar, toolbar), the number of pages that display, and the number of search results returned. • Configures the items available in the "Search By" box for user search in the self-service interface. • Enables direct access to the Expired Password change screen and bypass the self-service login page under certain conditions. The property key that allows these actions is ui.directExpiredChangePasswordEnabled.
SelfServiceScreenText.properties	Provides the text on the self-service user interface.
SelfServiceScreenText_*language*.properties	Provides the language-specific text on the self-service user interface. By default this file is SelfServiceScreenText_en.properties, which contains the English language bundle.
SelfServiceHomePage.properties	Defines the sections of the self-service user interface home page and the order in which they occur.
SelfServiceHelp.properties	Defines the links to html help pages on the self-service user interface. The html files are in the WAS_PROFILE_HOME\installedApps*node_name*\ITIM.ear\itim_self_service_help.war directory. You can redirect help by modifying the information in this file.
SelfServiceScreenTextKeys.properties	Provides label keys on the self-service user interface. This file can be used to assist with customization of screen text by providing a template to develop labels and instructions. The file contains labels that are set to the key name. For instance, password_title=password_title. For customization and development purposes. you can copy this file to SelfServiceScreenText_*language*.properties, where *language* is a language suffix that is not installed. You can then switch your browser locale from your current language to the unused language. Restart the web application to navigate through the pages and see the label keys instead of the value text. By switching your browser locale, you can then toggle between keys and values. When customization is complete, you can then copy and rename the file to the language suffix you want to use, for example SelfServiceScreenText_en.properties, to finalize changes.

Table 2. Java server pages (JSP) configuration files and descriptions

File Name	File Description
loginBanner.jsp	Contains the content of the banner on the self-service login page.
loginFooter.jsp	Contains the content of the footer on the self-service login page.

Table 2. Java server pages (JSP) configuration files and descriptions (continued)

File Name	File Description
loginToolbar.jsp	Contains the content of the toolbar on the self-service login page.
Home.jsp	Contains the content of the self-service home page.
banner.jsp	Contains the content of the self-service banner.
footer.jsp	Contains the content of the self-service footer.
nav.jsp	Contains the content of the self-service navigation bar.
toolbar.jsp	Contains the content of the self-service toolbar.

Table 3. Cascading style sheet (CSS) configuration files and descriptions

File Name	File Description
calendar.css	CSS file that contains the styles that are used for calendar widgets.
customForm.css	CSS file that contains the styles that are used to lay out custom forms for left to right language orientation.
customForm_rtl.css	CSS file that contains the styles that are used to lay out custom forms for right to left language orientation.
dateWidget_ltr.css	CSS file that contains the styles that are used for date widgets for left to right language orientation.
dateWidget_rtl.css	CSS file that contains the styles that are used for date widgets for right to left language orientation.
enduser.css	CSS file that contains main CSS styles for left to right language orientation.
enduser_rtl.css	CSS file that contains main CSS styles for right to left language orientation.
time.css	CSS file that contains the styles that are used for time widgets.
widgets.css	CSS file that contains the styles that are used for other widgets for left to right language orientation.
widgets_rtl.css	CSS file that contains the styles that are used for other widgets for right to left language orientation.

Backing up and restoring self-service user interface configuration files

Before you begin customization of the self-service user interface, back up all configuration files in IBM Security Identity Manager for later recovery purposes.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Log in to each computer that is running Security Identity Manager. Back up the following files:
- In the *WAS_PROFILE_HOME*\installedApps*node_name*\ITIM.ear\ itim_self_service.war\custom directory:
  – banner.jsp

- calendar.css
- customForm.css
- customForm_rtl.css
- dateWidget_ltr.css
- dateWidget_rtl.css
- enduser.css
- enduser_rtl.css
- footer.jsp
- Home.jsp
- loginBanner.jsp
- loginFooter.jsp
- loginToolbar.jsp
- nav.jsp
- time.css
- toolbar.jsp
- widgets.css
- widgets_rtl.css

Note: Default files are also available in the *ISIM_HOME*\data\defaults directory.

- In the *ISIM_HOME*\data directory:
  - SelfServiceHelp.properties
  - SelfServiceHomePage.properties
  - SelfServiceScreenText.properties
  - SelfServiceUI.properties
  - SelfServiceScreenTextKeys.properties

About this task

Any changes made to properties files require you restart the Security Identity Manager application. For instance, upon recovering any properties files, complete these steps:

Procedure

1. Using the WebSphere® administration console, click the **Applications** group in the left frame, and then click the **Enterprise Applications** link.
2. Select the check box next to the Security Identity Manager application and click the **Stop** button.
3. After the application stops, select the check box next to the Security Identity Manager application and click the **Start** button.
4. Verify that the recovery is complete by logging in to the self-service user interface.

User interface elements affected by view definitions

Defined views affect the visibility of task panels and other elements within the self-service interface.

View definition elements

View definitions can have the following effects on the self-service user interface:

Home page

> Adapts to the user's views by showing only the tasks and task panels on the home page that the user is granted. If the user is not allowed to view any tasks in a section, then the task panel also does not appear on the home page.
>
> Some task views, such as the Request Account task, have advanced views. To clarify, Request Account is a single task. If the **Request Account Advanced** view is granted, or if both the **Request Account** and **Request Account Advanced** views are granted, the user has a single **Request Account** task on the home page. The main Request Account page displays a search page in which the user can search for a service on which they can request an account. If only the standard Request Account view is granted, and not the advanced view, then the **Request Account** task appears on the home page. The main Request Account page displays a table that lists the services that the user can request an account on, instead of a search page.
>
> If the user can do both Change and View tasks for an account or profile, it combines them into a single task. For example, the task appears as **View or Change Account**.
>
> Some tasks might not appear if they are not enabled by the system administrator. For example, **Change Forgotten Password Information**requires the enablement of challenge response.
>
> The **Action Needed** task is only available if there are pending to-do items or challenge response information is not configured.
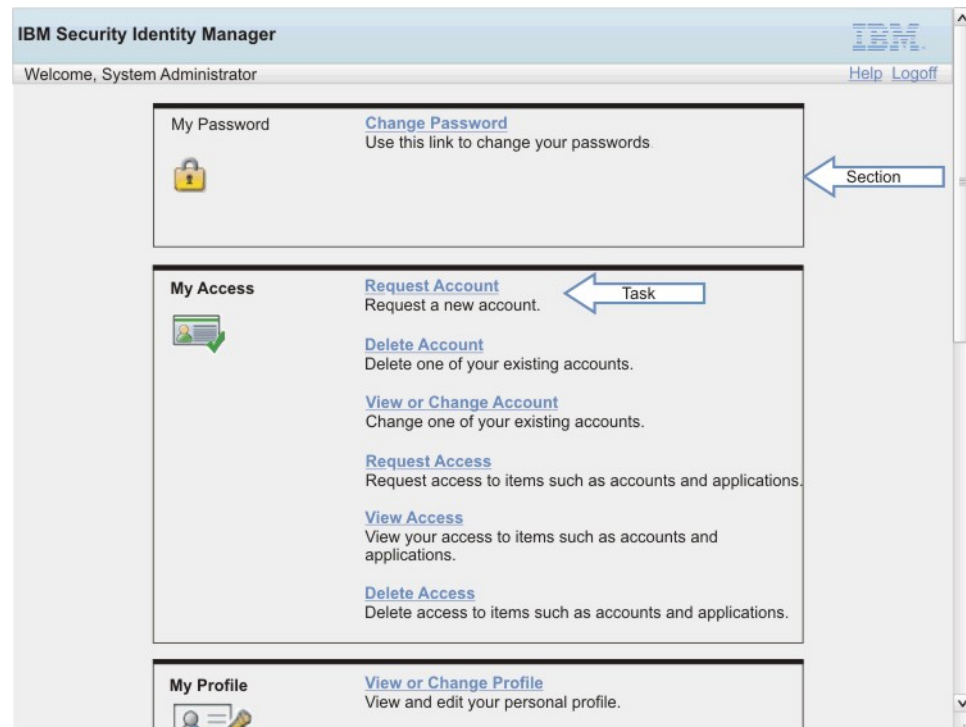
Figure 1. Home page elements

Related tasks

> Related task sections are displayed in many areas of the self-service application, for example when a request is submitted. View definitions can filter some or all of the sections from being shown based on the view definition permissions. For example, if the user does not have regular

access to **View My Requests**, then it is filtered from the **Related Tasks** task panel.
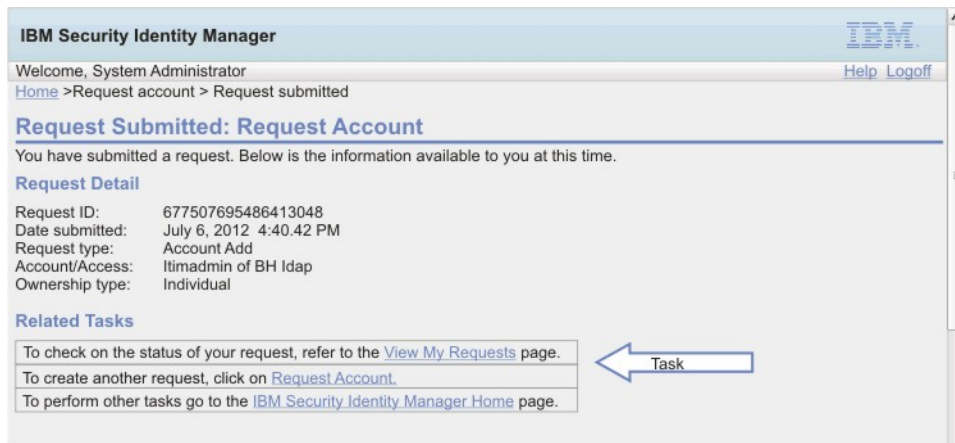
Figure 2. Related task panel element

Panel instruction text

The instruction text on certain screens can contain links to the **View My Requests** task. A different instruction message is displayed without the task link if the user is not granted the **View My Requests** task in a view definition.
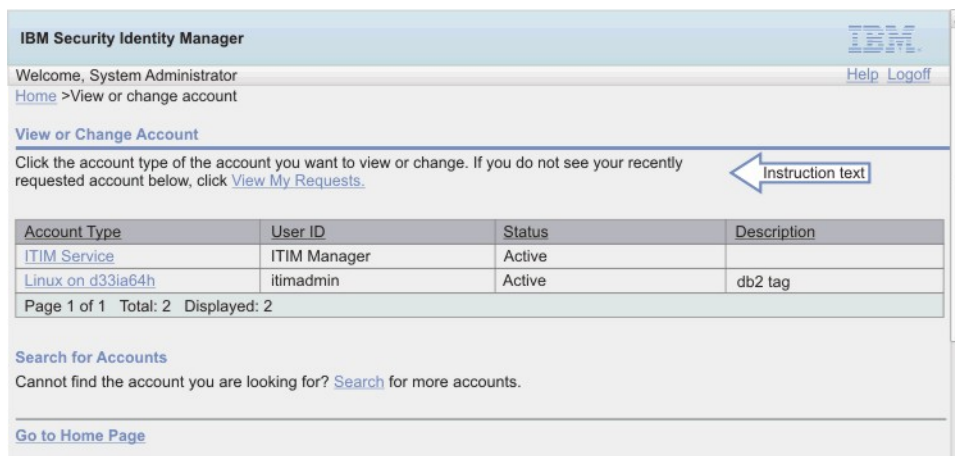
Figure 3. Instruction text panel element

Customizing labels, description, and other screen text

You can change the majority of the text displayed in the self-service user interface with customization.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Not every label can be customized by the user. Only the labels that have an entry present in the SelfServiceScreenTextKeys.properties file can be customized.

The following screen text items can be customized:

- Titles
- Subsection titles
- Subsection descriptions
- Field labels
- Table column headers and footers
- Button text

The following figure shows the visual representation of these screen text items.
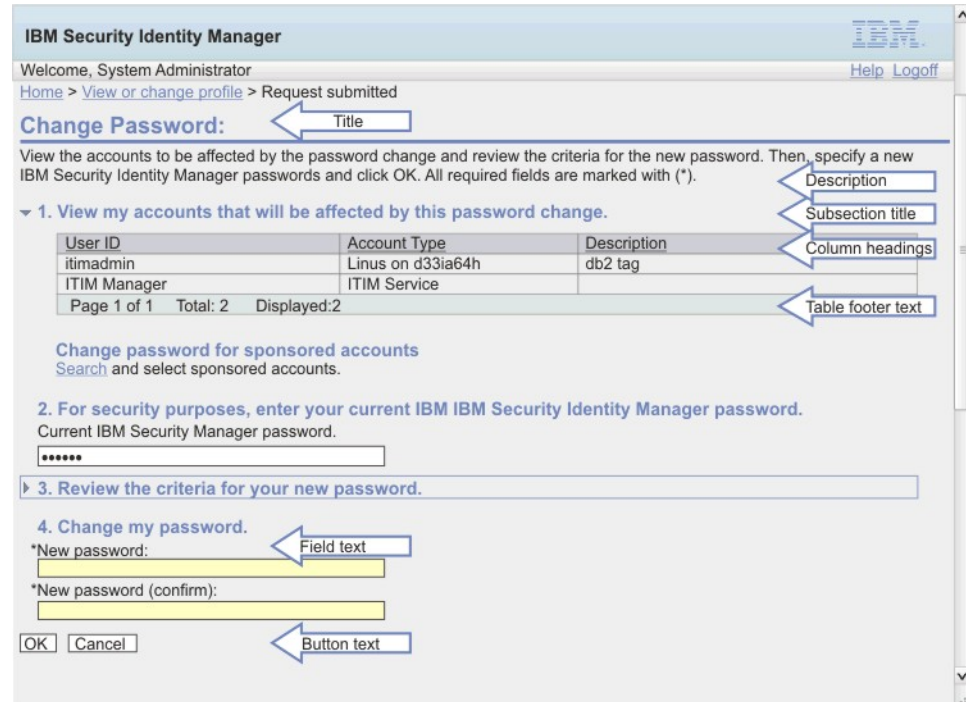
Figure 4. Screen text

Text that cannot be replaced includes error messages and text in the help content that you access by clicking on the help link. However, it is possible to redirect help requests to a different URL.

To customize screen text, complete the following steps:

Procedure

1. Make a backup copy of the SelfServiceScreenText.properties and SelfServiceScreenTextKeys.properties files. If you have installed a language pack, back up any other language pack files you plan to modify, including the SelfServiceScreenText_en.properties file. SelfServiceScreenText.properties is the default file used if no other matching language is found.
2. Edit the properties files. Modify the values of the screen text fields and save the files. Note that any changes you make to the SelfServiceScreenText.properties file should also be made to the SelfServiceScreenText_en.properties files to maintain consistency.
3. Restart the IBM Security Identity Manager application to make the changes effective.

Customizing website layout

You can change the layout in the self-service user interface with customization.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

High-level layout elements can be enabled and disabled from display in the self-service user interface with settings in the `SelfServiceUI.properties` file. The default layout contains a banner, toolbar, and footer.

Turning on and off page elements can give various layout options. The only required page element is the content element, which contains the tasks and task pages.

To show or hide a page element, change the ui.layout.show*name* property in the `SelfServiceUI.properties` file. For instance, ui.layout.showBanner controls the display of the banner section. Setting a property to true indicates that the element is included in the page. A setting of false indicates that the element is not included in the page.

Any change to the `SelfServiceUI.properties` file requires a restart of the IBM Security Identity Manager application in WebSphere to make the change effective.

The following figures show a visual representation of different layout elements and options.
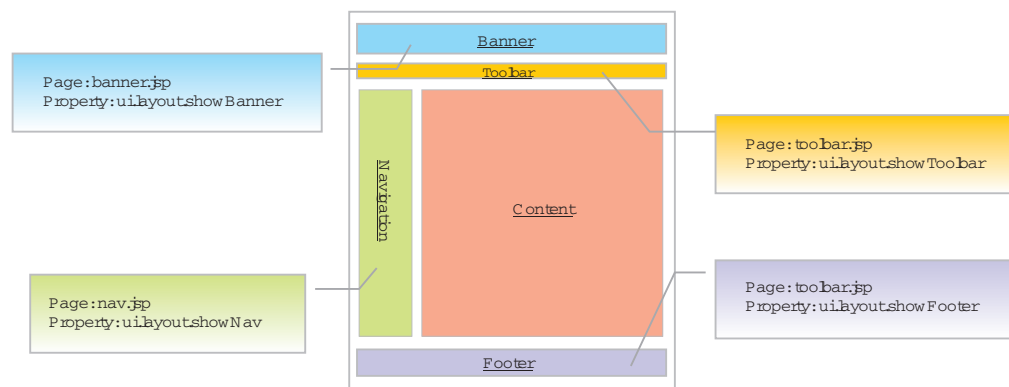
Figure 5. Layout elements
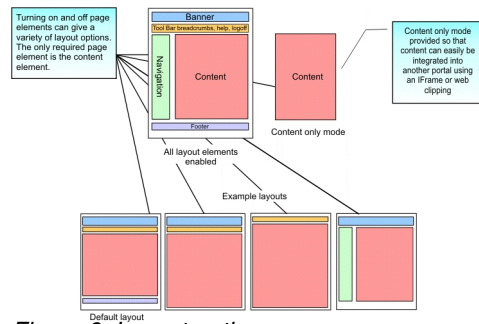
Figure 6. Layout options

The following table displays a list of properties and their details.

Table 4. Layout properties and details

Property	Description
ui.layout.showBanner	Controls the banner section. The default banner contains IBM and product images.
ui.layout.showFooter	Controls the footer section. The default footer contains the product copyright.
ui.layout.showToolbar	Controls the toolbar section. The default toolbar contains the welcome message, help link, logoff link, and breadcrumbs.
ui.layout.showNav	Controls the Navigation bar. **Note:** No default content is included for the navigation bar.

To customize the layout, complete the following steps:

Procedure

1. Make a backup copy of the SelfServiceUI.properties file in the *ISIM_HOME*\data directory.
2. Edit the SelfServiceUI.properties file. Modify the values of the screen text fields and save the file.
3. Restart the Security Identity Manager application to make the changes effective.

Customizing banner, footer, toolbar, and navigation bar content

You can change the appearance of the self-service user interface by customizing the banner, footer, toolbar, and navigation bar.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Content in the *WAS_PROFILE_HOME*\installedApps*node_name*\ITIM.ear\ itim_self_service.war\custom directory can be replaced or modified to alter the appearance of the self-service user interface. You can replace or modify the banner, footer, toolbar, and navigation bar.

The layout elements are JSP fragments that are included in the layout of the web page when the JSP is rendered.

The following table displays a list of layout elements and their corresponding files, which are in the *WAS_PROFILE_HOME*\installedApps*node_name*\ITIM.ear\ itim_self_service.war\custom directory.

Table 5. Layout elements and file names

Layout element	File name
Banner	banner.jsp
Footer	footer.jsp
Toolbar	toolbar.jsp
Navigation bar	nav.jsp

To modify these files, complete these steps:

Procedure

1. Make backup copies of the files and store the files you want to modify in a temporary directory.
2. Edit the files in the temporary directory and copy the updated files back into the deployed WebSphere directory. No restart of the IBM Security Identity Manager application is required for these changes to take effect.

What to do next

The default version of these files is shipped with the product archive. Be sure to back up the custom version of the files you created so that your changes are not lost.

Request parameters and content examples for use in customizing user interface content

This section describes the request parameters that you can use in JSP files to customize content.

Request parameter values

To support dynamic content such as breadcrumbs, help links and user IDs, a few request parameters are available. The following table shows these properties, their possible values, and a description.

Table 6. Request parameters, values, and descriptions

Property name	Value	Description
loggedIn	true or false	Flag that indicates whether the user is logged in.
usercn	The common name of the owner of the logged in account	**Note:** This value is only set if the user is logged in.
langOrientation	ltr or rtl	Indicates the language direction of the current locale, either left to right, or right to left.

Table 6. Request parameters, values, and descriptions (continued)

Property name	Value	Description
helpUrl	/itim/self/ Help.do?helpId=*example_url*	URL to the help web page with the *helpId* parameter set for the current page.
helpLink	Example: home_help_url	The *helpId* for the current page. The value *home_help_url* maps to the corresponding key in the `SelfServiceHelp.properties` file.
breadcrumbs	*example_message_key1* *example_message_key2* *example_message_key3*	A list of message keys that correspond to entries in the `SelfServiceScreenText.properties` file.
breadcrumbLinks	*pathname1* *pathname2* *empty_string*	A list of links that is the same length as the breadcrumbs list.

Examples of request parameters in toolbar.jsp

The default file `toolbar.jsp` contains the logic to display the welcome message and help links. This logic can be moved into the other layout elements; for example, the welcome message might be provided in the banner.

Displaying the welcome message

The following code checks to see whether the user's common name is set. If so, it translates the welcome message and substitutes the name into the message.

Note: The self-service user interface message labels and keys are defined in the `SelfServiceScreenText.properties` file.

```
<%-- If the Users Common Name is not empty display it. Note this value is not
     set until the user is logged in --%>

<c:if test="${!empty usercn and loggedIn == true}">
    <%--Translate the Welcome, Common Name message passing in the name --%>
    <fmt:message key="toolbar_username" >
        <fmt:param><c:out value="${usercn}"/></fmt:param>
    </fmt:message>
</c:if>
</div>
<%-- end user info -- %>
```

Displaying help links

The following code adds the Help link to the page. The helpUrl is retrieved from help attributes, and the help label is translated for display.

```
<%-- Add Help Link to the page --%>

<a id="helpLink" href="javascript:launchHelp('<c:out value='${helpUrl}')">
    <fmt:message key="toolbar_help"/></a>
```

Supporting logoff

The Logoff link can only be displayed if the user is logged in. The following code tests to see whether the **loggedIn** request parameter is true. If so, the code translates the label for the logoff link and includes the link in the page.

```
<%-- If the user is logged in display the logoff link --%>
<c:if test="${loggedIn == true}">
   <a id="logofflink" href="/itim/self/Login/Logoff.do">
       <fmt:message key="toolbar_logoff"/></a>
</c:if>
```

Displaying breadcrumbs

The following code adds the breadcrumbs attribute to the page. The breadcrumbs attribute contains the list of label keys for the breadcrumbs attribute. The breadcrumbLinks contain URL information for each breadcrumb label. A value of null or empty for the breadcrumbLinks indicates that the breadcrumb is not linkable.

```
<%-- If the breadcrumbs label keys are not empty then display --%>

<c:if test="${!empty breadcrumbs}">
    <c:forEach items="${breadcrumbs}" var="breadcrumb" varStatus="status">

        <c:if test="${status.index > 0}">
             &gt; 
        </c:if>

        <c:choose>
            <%-- If the action link is not empty for the current label then
                 create a link for the breadcrumb --%>

            <c:when test="${!empty breadcrumbLinks{status.index}}">
                <html:link action="${breadcrumbLinks{status.index}}">
                <fmt:message key="${breadcrumb}"/></html:link>
            </c:when>
            <%-- If the action link is empty then just translate the
                 label for the breadcrumb --%>
            <c:otherwise>
                <fmt:message key="${breadcrumb}"/>
            </c:otherwise>
        <c:choose>

    </c:forEach>

</c:if>
```

Customizing the self-service home page

You can change the home page in the self-service user interface with customization.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

The home page refers to the main page that gets loaded in the content layout element after a user logs in to the self-service user interface.

Section and task definitions tie defined views to tasks, and group tasks into sections, also called task pages. These section and task definitions are defined in the SelfServiceHomePage.properties file in the *ISIM_HOME*\data directory.

The home page layout element is a JSP fragment that is included in the layout of the web page. This layout information is stored in the Home.jsp file in the *WAS_PROFILE_HOME*\installedApps*node_name*\ITIM.ear\itim_self_service.war\custom directory.

You can add tasks and sections to the home page by updating the SelfServiceHomePage.properties file. The comments in the file explain the file format. You can alter the content without modifying the jsp file.

To customize the home page, complete these steps:

Procedure

1. Make a backup copy of the SelfServiceHomePage.properties file in the *ISIM_HOME*\data directory.
2. Make a backup copy of the Home.jsp file in the *WAS_PROFILE_HOME*\installedApps*node_name*\ITIM.ear\itim_self_service.war\custom directory.
3. Edit the SelfServiceHomePage.properties file. Modify the values and save the file.
4. Copy the Home.jsp file to another directory, then modify the file in that directory and copy the updated file back into the *WAS_PROFILE_HOME*\installedApps*node_name*\ITIM.ear\itim_self_service.war\custom directory. The default version of these files is shipped with the product archive. Be sure to back up the custom version of the files you created so your customizations are not lost.
5. Restart the IBM Security Identity Manager application in WebSphere to make the changes effective.

Request parameters and content examples for use in customizing the home page content

This section describes the request parameters that you can use in JSP files to customize home page content.

Home page form parameters

To support dynamic home page content such as sections, action-needed sections, tasks, a Java bean is available as a request parameter called **HomePageForm**. The home page Java bean contains a handful of methods that can be used to access information about sections and tasks.

Table 7. Home page request parameters, values, and descriptions

Property name	Value	Description
sections	List of Section Java beans	A list of sections the current user can view.
sectionToTaskMap	Map of sections to their corresponding tasks	A map that links a specified section Java bean to a task Java bean.

Table 7. Home page request parameters, values, and descriptions (continued)

Property name	Value	Description
actionNeededSection	Section Java bean, or null	A section Java bean that contains the pending actions for the current user. A null is used if no pending actions exist for the current user.

The following properties are available for the section Java bean:

Table 8. Section Java bean request parameters, values, and descriptions

Property name	Value	Description
titleKey	Title message key for the section	The message key for the section title.
iconUrl	Icon URL, or null	The URL path for the icon to be used for the section. A null is used to indicate that no icon is used.
iconAltTextKey	Text key	Text key to be used as the alternate text for the icon of the section.
tasks	List of task Java beans	A list of tasks that can be displayed in the section

The following properties are available for the task Java bean:

Table 9. Task Java bean request parameters, values, and descriptions

Property name	Value	Description
urlPath	URL	A URL path to this task.
urlKey	Text key	The text key to be used for the link to this task.
descriptionKey	Text key	Text key to be used as the description of this task.

Examples of request parameters in home.jsp

The following code obtains the **HomePageForm** Java bean and iterates through the available sections and tasks and creates links to each available task.

```
<c:set var="pageConfig" value="${HomePageForm}" scope="page" />
<c:forEach items="${pageConfig.sections}" var="section">
    <%-- Process each section here --%>
    <c:forEach items="${pageConfig.sectionToTaskMap[section]}" var="task">
    <%-- Process each section here --%>
        <a href="/itim/self/<c:out value="${task.urlPath}"/>"
           title="<fmt:message key="${task.urlKey}" />">
           <fmt:message key="${task.urlKey}" />
        </a>
        <fmt:message key="${task.descriptionKey}" />
    </c:forEach>
</c:forEach>
```

Customizing style sheets

You can change the appearance of the self-service user interface by customizing Cascading Style Sheets (CSS).

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Cascading Style Sheets (CSS) are used to style the appearance of the self-service user interface. You can edit the style sheets to modify the fonts, colors, and other styles associated with the self-service user interface. This section describes the location of the style sheets, and key styles to edit to customize the user interface to match the look and feel of your website.

The default deployed CSS files are compressed and optimized with bandwidth in mind for scalability. The non-optimized versions (with whitespace/formatting intact) can be found in the `ISIM_HOME`\defaults\custom directory. The CSS files stored in the `WAS_PROFILE_HOME`\installedApps\`node_name`\ITIM.ear\ itim_self_service.war\custom directory are unsuitable for editing. Copy the default files stored in the `ISIM_HOME`\defaults\custom directory to another directory. Edit the style sheets and then copy your changed files to the `WAS_PROFILE_HOME`\installedApps\`node_name`\ITIM.ear\itim_self_service.war\ custom directory.

The following table shows the CSS files that can be modified to adjust the self-service user interface.

Table 10. Cascading Style Sheet file names

CSS file name	Description
end_user.css	CSS file that contains main CSS styles for left to right language orientation.
end_user_rtl.css	CSS file that contains main CSS styles for right to left language orientation.
widgets.css	CSS file that contains styles used for widgets, such as those contained in Profile, Account, and RFI forms, for left to right language orientation. **Note:** Editing this file takes more advanced CSS skills.
widgets_rtl.css	CSS file that contains styles used for widgets, such as those contained in Profile, Account, and RFI forms, for right to left language orientation. **Note:** Editing this file takes more advanced CSS skills.
dateWidget_ltr.css	CSS file that contains styles used for date widgets, such as those contained in Profile, Account, and RFI forms, for left to right language orientation. **Note:** Editing this file takes more advanced CSS skills.

Table 10. Cascading Style Sheet file names (continued)

CSS file name	Description
dateWidget_rtl.css	CSS file that contains styles used for date widgets, such as those contained in Profile, Account, and RFI forms, for right to left language orientation. **Note:** Editing this file takes more advanced CSS skills.
time.css	CSS file that contains styles used for time widgets, such as those contained in Profile, Account, and RFI forms. **Note:** Editing this file takes more advanced CSS skills.
customForm.css	CSS file that contains styles used for layout forms, such as those contained in Profile, Account, and RFI forms, for left to right language orientation. **Note:** Editing this file takes more advanced CSS skills.
customForms_rtl.css	CSS file that contains styles used for layout forms, such as those contained in Profile, Account, and RFI forms, for right to left language orientation. **Note:** Editing this file takes more advanced CSS skills.

The following figures provide a visual representation of page elements for which style changes can apply.
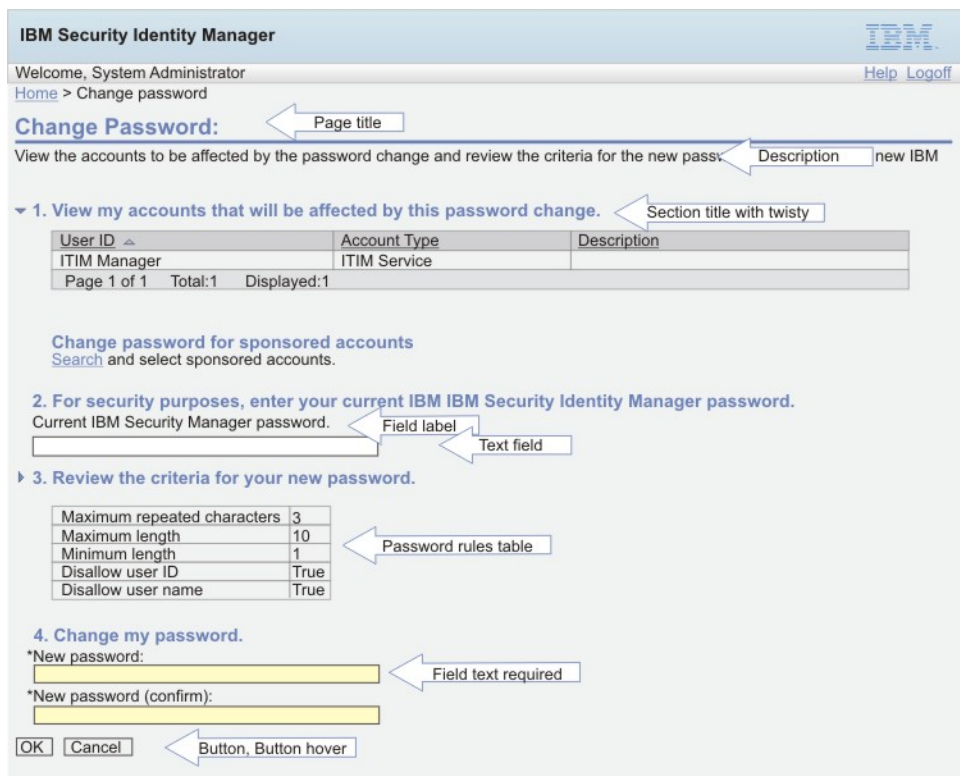
Figure 7. Page elements for style changes
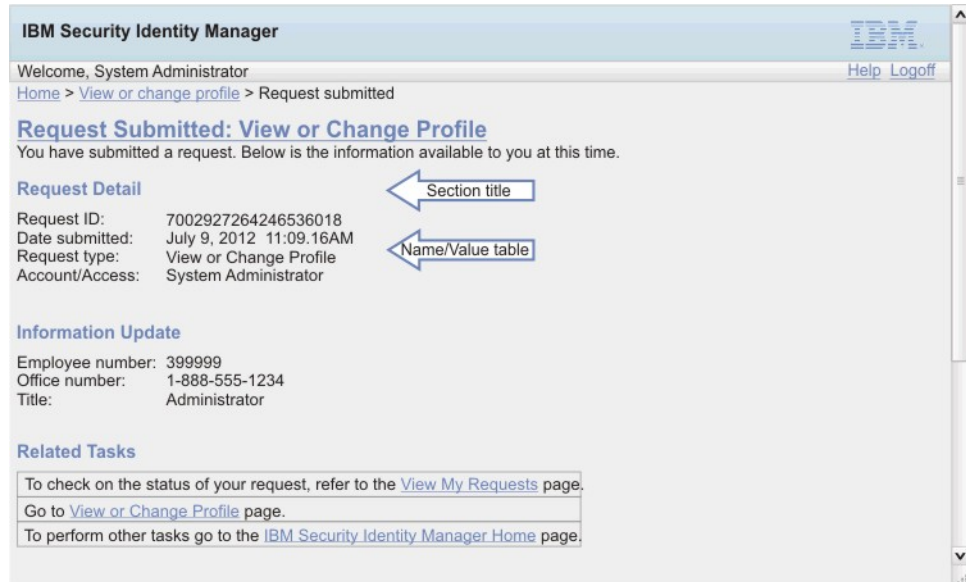
Figure 8. Page elements for style changes (continued)
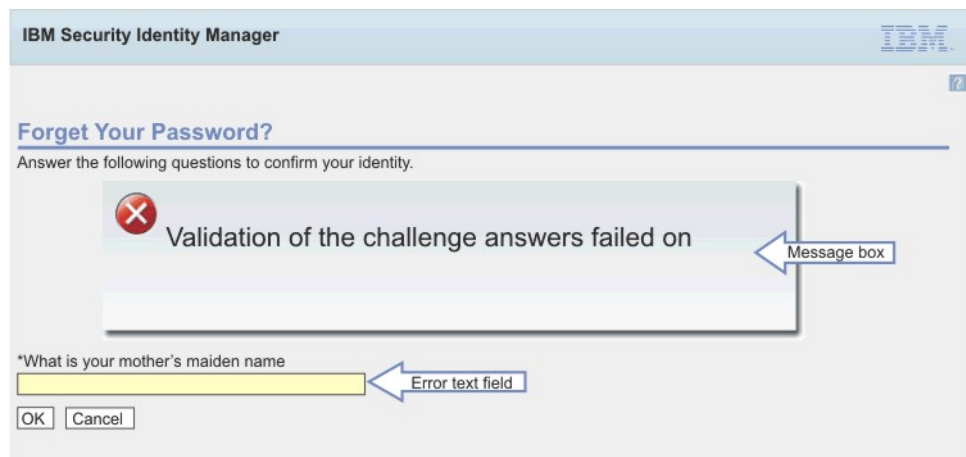
Figure 9. Page elements for style changes

The following table provides a reference for the main CSS styles.

Table 11. CSS styles reference

Element	Example	Main style selector	Description
Page Title	**Page Title**	Type selector: h1	Element used for all page titles.
Section title	**Subsection Title**	Type selector: h2	Section titles for pages that do not contain a twisty.

Table 11. CSS styles reference (continued)

Element	Example	Main style selector	Description
Section title (twisty)	**Twisty Title**	Type selector: h3	Section titles on pages which contain twisty sections. The titles are intended to allow space for the twisty image.
Breadcrumbs	Home > View or change profile	Type selector: #breadcrumbs	The breadcrumbs navigation trail shown on the top left above the page title.
Button, Button Hover, Disabled Button	Button Button hover Button disabled	Class selectors: • .button • .button_hover • .button_disabled	These button styles cover the majority of buttons in the user interface. The hover style is used when a mouse hovers over the button
Inline button, Inline button hover	Button inline Button inline hover	Class selectors: • .button_inline • .button_inline_hover	Used for a subset of buttons with special layout requirements.
Page/section descriptions	This is a description.	Class selector: .description	Page and section descriptions. The description is contained in a <div> block. Therefore, you could add borders, colors, etc. if desired.
Field labels	Field label	Type selector: label	Field labels on forms.
Text field	Text field (white field background default)	Class selector: input.textField_std	Standard text fields.
Required text field	Required text field (yellow field background default)	Class selector: input.textField_required	Required text fields.
Error text field	Error text field (red field border default)	Class selector: input.textField_error	Text fields in an error state.
Warning text field	Warning text field (yellow field border default)	Class selector: input.textField_warning	Text fields in a warning state.

Table 11. CSS styles reference (continued)

Element	Example	Main style selector	Description
Field/value tables	```Field Name1 Field value1Field Name2 Field value2Multi-valued Field3 Item 1 Item 2 Item 3 Item 4Multi-valued Field3 Item 1 Item 2```	Class selector: table.nameValueTable	Field value tables are used through out the user interface to display a field name and one or more corresponding values. For example, the Information section of the request submitted pages use name value tables. The selector is shown for the table. Additional selectors exist that style the rows, cells, multi-value lists, and name columns for this table.
Password rules table	◆ Rule1 Value1 AccountInfo1 ◆ Rule2 Value2 AccountInfo2	Class selectors: • .pwRulesTable • .pwRulesTable .ruleCol • .pwRulesTable .valueCol • .pwRulesTable .accountInfoCol • .button_inline_hover	The password rules table is used to style the password rules sections through out the user interface. The table consists of three columns; a rule column, a value column, and an account information column.
Message box	CTGIMU651E A required field does not have a specified value. Close Message	div.messageBoxComposite	The message box composite is the main CSS selector for the message box. Additional selectors exist to specify the image / link / and message layout.

To customize the style sheets, complete these steps:

Procedure

1. Make a backup copy of the CSS files in the `WAS_PROFILE_HOME`\installedApps\ `node_name`\ITIM.ear\itim_self_service.war\custom directory.
2. Copy the CSS files from the `ISIM_HOME`\defaults\custom directory to another directory, then modify the files in that directory and copy the updated files to the `WAS_PROFILE_HOME`\installedApps\`node_name`\ITIM.ear\ itim_self_service.war\custom directory. Be sure to back up the custom version of the files you created so your customizations are not lost.

Merging style sheet customization from a previous version

After upgrading from a previous version of IBM Tivoli® Identity Manager, you must reapply any customizations you made to the cascading style sheet for the self-service user interface.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

You need access to the file system on which IBM Tivoli Identity Manager is deployed.

You must have a working knowledge of Security Identity Manager and cascading style sheets (CSS).

About this task

Customizations, including view definitions, that are defined through the administrative console are preserved during an upgrade. Updates to SelfServiceScreenText.properties are automatically merged as well.

However, after the upgrade program completes, the deployed self-service cascading style sheet (CSS) is restored to factory defaults. First merge the updated CSS values into your customized CSS skin created for your previous version of the product. Then reapply your customized files to the deployed self-service war.

Note: During the upgrade, ITIM.ear file is backed up from WebSphere application server to `ISIM_HOME`/data/backup/ITIM.ear directory. You can view the itim_self_service.war/custom directory for a copy of the CSS skin that was deployed before the upgrade.

To merge CSS customizations, make the following additions and modifications to your original IBM Tivoli Identity Manager CSS files.

Note: If modifications for right-to-left (RTL) CSS files (for example, enduser_rtl.css) were made, merge the modifications by using a text comparison tool. Make the equivalent changes to the enduser_rtl.css file as the enduser.css file, but adjust for the right-to-left layout.

Procedure

1. Open your existing CSS file with an editor.

   This file can be found in the directory *ISIM_HOME*/data/backup/ITIM.ear

   Note: For a separate system upgrade, copy the files from the deployed ITIM.ear file.

2. Add the appropriate changes based on your migration path. See "CSS updates."

   For migration from Version 5.0 to Version 6.0, add the CSS changes made in both Version 5.1 and Version 6.0.

   For migration from Version 5.1 to Version 6.0, add the CSS changes made in Version 6.0 only.

3. Copy the updated CSS files to the Self Service user interface custom directory `itim_self_service.war/custom`.

Results

These modifications take effect immediately, and no restart of the Security Identity Manager application is required.

CSS updates

CSS changes added in 5.1:

enduser.css

     Description: Added twistie for h2 headings.

     Add the following text:

```
a.twistie_open h2{
 margin-left:0px;
 background-repeat: no-repeat;
 background-position: left;
 padding-left: 15px;
 background-image: url("/itim/self/images/twistie_open.gif");
}

a.twistie_closed h2{
 margin-left:0px;
 background-repeat: no-repeat;
 background-position: left;
 padding-left: 15px;
 background-image: url("/itim/self/images/twistie_closed.gif");
}
```

     Description: Changed Review Activities instructions to be a twistie.

     Add the following text:

```
/* Review Activity Styles */
#instructionDetailTwistieDiv {
 white-space: expression("pre"); /* IE */
 white-space: -moz-pre-wrap; /* Firefox */
 word-wrap: break-word;
}
/* End Review Activity Styles */
```

     Description: Added CSS Styles for User Recertification.

     Add the following text:

```
/* Recertification items table styles */
table.recertItemsTable {
 width: auto;
}

table.recertItemsTable th {
 padding: .2em 1em .2em 1em;
 background-color: #C0C0C0;
 white-space: nowrap;
 text-align: left;
}

table.recertItemsTable td {
 padding: .2em 1em .2em 1em;
 border: 1px solid #C0C0C0;
}

table.recertItemsTable tr.recertItemRow td {
 border-bottom-style: none;
}

table.recertItemsTable tr.recertSubItemRow td {
 border-top-style: none;
 border-bottom-style: none;
}

table.recertItemsTable tr.altRow {
 background-color: #F6F6F6;
}

table.recertItemsTable .selectAllOptions {
 display: inline;
 padding: 0 .5em 0 .5em;
 font-weight: normal;
}

table.recertItemsTable .selectAllOptions a {
 padding: 0 .3em 0 .3em;
 color:#1375D7;
 font-weight: normal;
}

table.recertItemsTable .recertItemSelectAllOptions {
 display: inline;
 padding: 0 .5em 0 .5em;
 font-weight: normal;
 font-size: .8em;
}

table.recertItemsTable .recertItemSelectAllOptions a {
 padding: 0 .3em 0 .3em;
}

table.recertItemsTable a.recertExpandCollapseLink {
 margin-right: .2em;
}

table.recertItemsTable a.recertExpandCollapseLink img {
 border: none;
 vertical-align: bottom;
}

table.recertItemsTable div.recertItem {
 display: inline;
 margin-bottom: 2px;
}
```

```
table.recertItemsTable td.recertItemImpact {
 text-align: center;
}

table.recertItemsTable div.recertItemDescription {
 max-width: 300px;
 font-size: .8em;
}

table.recertItemsTable div.recertItemImpactedBy {
 display: inline;
 margin-bottom: 2px;
}

table.recertItemsTable td.recertItemActionRecertify {
 width: expression("0%"); /* IE */
 width: 1px; /* Firefox */
 white-space: nowrap;
 padding-right: 0;
 border-right: none;
}

table.recertItemsTable td.recertItemActionRecertifyErrorNone {
 width: expression("0%"); /* IE */
 width: 1px; /* Firefox */
 white-space: nowrap;
 padding: .2em 0 .2em 13px;
 border-right: none;
}

table.recertItemsTable td.recertItemActionRecertifyErrorExists {
 width: expression("0%"); /* IE */
 width: 1px; /* Firefox */
 white-space: nowrap;
 padding: .2em 0 .2em 5px;
 border-right: none;
}

table.recertItemsTable td.recertItemActionReject {
 width: 0%;
 white-space: nowrap;
 padding-left: 0;
 border-left: none;
 border-right: none;
}

table.recertItemsTable td.recertItemActionBlank {
 height: 24px;
}

table.recertItemsTable label.recertItemAction {
 display: inline;
}

table.recertItemsTable td.recertItemSelectAll {
 width: 0%;
 white-space: nowrap;
 padding-left: 0;
 border-left: none;
}

table.recertItemsTable .recertSubItem {
    font-size: 1em;
    margin: 0 0 0 1em;
}

table.recertItemsTable div.recertItemDecision {
```

```
 display: block;
 margin-bottom: 2px;
 margin-top: 5px;
}
/* End recertification items table styles */

.simpleLink:link, .simpleLink:visited {
 font-weight: normal;
}


.requiredInstruction {
 font-size: .8em;
 margin: 1em 0 0 1em;
 background-image: url("/itim/self/images/required_field.gif");
 background-repeat: no-repeat;
 background-position: center left;
 padding-left: 12px;
}
```

CSS changes added in 6.0:

enduser_extra.css

Import enduser_extra.css

Add the following text:

```
@import "enduser_extra.css";
```

Description : Background color of self console has been changed to whitesmoke color.

Add the following style in body tag selector

```
background-color: #F5F5F5;
```

Description : Background color of banner has been changed to light blue.

Update the following style in banner id selector:

```
background-color: black;
```

to

```
background-color: #c8e0f8;
```

Description : Various changes in login screen.

Update the loginContainer id selector using the following style:

```
#loginContainer{
  width:619px;
  margin:20px auto;
  margin-left: auto ;
  margin-right: auto ;
    background-position:left top;
   background-repeat:no-repeat;
  background-color:#FFF;
  padding:0;
  border: solid 1px #bbbbbb;
  font-family:Arial,Verdana,Helvetica,Tahoma,sans-serif;
  font-size:12px;
  color:#555555;
  overflow: hidden;
  text-align: left;
  }
```

Description : Layout changes for product login image

Add the following style in loginImage id selector:

```
margin-left: 40px;
margin-top: -30px;
```

Description : Layout changes and font size changes for product version

Update content in loginVersion id selector:

```
margin-left: 110px;
font-size:10px;
```

Description : Layout changes and font size changes for login content

Update content in loginContent id selector

```
margin-left: 40px ;
margin-right: 20px ;
font-size:14px;
```

Description : Styling for new help link in login screen

Add the following style:

```
#loginToolbar {
  margin-right: 20px ;
  }
```

Description : Styling for message box

Add the following style:

```
#messageBox {
  margin-right:80px;
  font-size:14px;
  }
```

Description : Add an extra tag selector h2i to the existing group selector declaration box for h1, h2, h3. Also, add corresponding style for h2i tag selector.

Add the following style:

```
h2i {
  font-size:120%;
  border-bottom-style: none;
  border-bottom-width: 2px;
  margin-bottom: 0px;
  margin-left: 15px;
  }
```

Description : Added hand cursor for anchor.

Add the following style in pseudo class a:LINK, a:VISITED:

```
cursor: hand;
```

Description : Added a new class **descriptioni**.

Add the following style:

```
.descriptioni {
  display: block;
    margin-bottom: 20px;
  margin-left: 15px;
  }
```

Description : Added new style for tables.

Add the following style:
```
span.tableLayout {
  display:inline-block;
  min-width:80%;
  margin : 10px 10px 10px 0 ;
  }
```

Description : Updated width for table column header.

Add the following style in **thead th** selector:
```
width: auto;
```

Description : Added a new class dataTable

Add the following style:
```
.dataTable {
  width:100%;
  margin: 0px;
  }
```

Description : Added a new class customHeader

Add the following style:
```
customHeader {
  text-align: left;
  border-style: solid;
  background-color: #E6E6E6;
   border-width:1px 1px 1px 1px;
  border-color:#C8C8C8 #C8C8C8 #737373 #C8C8C8;
  width: auto;
  }
```

Description : Added a new class customHeaderTable

Add the following style:
```
.customHeaderTable {
  border-top-style:hidden;
  }
```

Description : Updated the width of anchors in column header

Add the following style in **thead th a:LINK, thead th a:VISITED** selector:
```
width: auto;
```

Description : Added style for account tables

Add the following style:
```
table #global_table_accounttype {
   width: 20%;
  }
```

```
table #global_table_userid_10 {
  width: 10%;
  }


table #global_table_description_30 {
width: 30%;
}
```

Description : Added a new class viewRequestsCustomHeaderStyle

Add the following style:
```
.viewRequestsCustomHeaderStyle{
  text-align: left;
  padding: 5px;
  vertical-align: middle;
  }
```

Description : Added style for custom header labels

Add the following style:
```
div.viewRequestsCustomHeaderStyle label{
  display: inline;
  font-weight:bold;
  }
```

Description : Added new style for recertItemOwnershipType

Add the following style:
```
table.recertItemsTable div.recertItemOwnershipType {
   max-width: 300px;
  font-size: 1em;
  }
```

Description : Added styles for table cells

Add the following style:
```
div.tableCellContent {
  white-space:nowrap;
  overflow:hidden;
  width:25em;
  text-overflow:ellipsis;"
  }
```

Add an extra tag class tfootTd to the existing group selector declaration
box for tfoot th selector. Also, add an extra tag class label to the existing
group selector declaration box for label selector.

Description : Removed following styles that are not used anymore.

Remove the following styles:
```
th.reviewActivitiesCustomHeader {
  text-align: left;
  border-style: solid;
  border-width:1px 1px 1px 1px;
  background-color: #E6E6E6;
  border-color:#FFFFFF #C8C8C8 #737373 #FFFFFF;
  }


.simpleLink:link, .simpleLink:visited {
  font-weight: normal;
  }
```

```
                        .label_accessibility {
                         display: none;
                        }

                       .requiredInstruction {
                          font-size: .8em;
                          margin: 1em 0 0 1em;
                          background-image: url("/itim/self/images/required_field.gif");
                          background-repeat: no-repeat;
                          background-position: center left;
                          padding-left: 12px;
                        }
```

What to do next

These modifications take effect immediately. A restart of the Security Identity Manager application is not required.

Redirecting help content

You can redirect help requests to your own website to deliver custom help content.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Editing the out-of-the box help content shipped with the self-service user interface is not supported. But it is possible to redirect help requests to your own website to deliver custom help content in line with your corporate appearance.

The `SelfServiceHelp.properties` file specifies the base URL that help requests are sent to. These files are in the *ISIM_HOME*\data directory.

The following table shows the property and property description for self-service help.

Table 12. Self-service help properties and description

Property	Description
helpBaseUrl	Specifies the base URL to send help requests to. A blank value indicates that help goes to the default URL for the self-service user interface.
Help Id mappings: helpId = relative page URL	The help mappings section maps IDs from specific pages to a relative URL sent to the help server.

The Help URL is the combination of the helpBaseUrl + locale + relativeHelppageURL

For example:
```
helpBaseUrl=http://myserver:80
locale = en_US
```

Locale is determined by resolving the SelfServiceScreenText.properties resource bundle for the current logged in user and with the associated locale.

```
loginId/relativeURL = login_help_url=ui/ui_eui_login.html
```

Therefore, the final URL = http://myserver:80/en_US/ui/ui_eui_login.html.

To redirect help, complete these steps:

Procedure

1. Make a backup copy of the `SelfServiceHelp.properties` file in the `ISIM_HOME`\data directory.
2. Change the helpBaseUrl property in the `SelfServiceHelp.properties` file.
3. Update helpId mappings to use the relative URLs for your server.
4. Add pages to your server for the appropriate locales.
5. Restart the IBM Security Identity Manager application in WebSphere to make the changes effective.

Configuration of direct access to self-service tasks

Many pages in the interface can be directly accessed from other HTML pages or integrated with a company intranet portal.

The user must first authenticate by either logging in through the Login page or through a single sign-on. When a user attempts to access a page for which direct access is supported, the following occur:

- If the page is defined by a configured view, the page is displayed.
- If the page is not in a configured view, an error page is displayed instead of the requested page.

Note: Direct access to the **Approve and Review Requests** task is supported even if it is not enabled in a configured view. Also, depending on group membership, more than one view configuration might apply. If at least one view configuration that applies to a user includes the task that the user is attempting to access, the page is displayed.

The following table displays tasks and URLs that are supported for direct access, and that you can link to from your company intranet portal.

Table 13. Direct-access tasks and URLs

Task	URL
Logon Page	http://*server_name*/itim/self
Change Password	http://*server_name*/itim/self/PasswordChange.do
Change Forgotten Password Information	http://*server_name*/itim/self/changeForgottenPasswordInformation.do
Expired Password (bypass the Login page)	http://*server_name*/itim/self/Login/DirectExpiredPasswordChange.do?expiredUserId=*userID* **Note:** This solution works only if single sign-on is not enabled and the ui.directExpiredChangePasswordEnabled property is set to true in SelfServiceUI.properties file.
Request Access	http://*server_name*/itim/self/RequestAccess.do
Request Access (for a specific access request)	http://*server_name*/itim/self/RequestAccess.do?accessDN=*accessDN*

Table 13. Direct-access tasks and URLs (continued)

Task	URL
View Access	http://*server_name*/itim/self/ViewAccess.do
Delete Access	http://*server_name*/itim/self/DeleteAccess.do
Delete Access Confirmation (for a specific access deletion)	http://*server_name*/itim/self/DeleteAccess.do?accessDN=*accessDN*
Request Account	http://*server_name*/itim/self/RequestAccounts.do
Request Account (directly access the request account form for a specific service)	http://*server_name*/itim/self/RequestAccounts.do?serviceDN=*serviceDN*
View Account	• http://*server_name*/itim/self/ViewAccount.do (multiple accounts view) • http://*server_name*/itim/self/ViewAccount.do?userID=*userID*&serviceDN=*serviceDN* (specific service account)
View or Change Account	http://*server_name*/itim/self/ViewChangeAccount.do
Change Account	• http://*server_name*/itim/self/ChangeAccount.do (multiple accounts view) • http://*server_name*/itim/self/ChangeAccount.do?userID=*userID*&serviceDN=*serviceDN* (specific service account)
Delete Account	http://*server_name*/itim/self/DeleteAccount.do
Delete Account Confirmation	http://*server_name*/itim/self/DeleteAccount.do?userID=*userID*&serviceDN=*serviceDN* (specific service account)
View Profile	http://*server_name*/itim/self/ViewProfile.do
Change Profile	http://*server_name*/itim/self/ChangeProfile.do
View or Change Profile	http://*server_name*/itim/self/ViewChangeProfile.do
View My Requests	• http://*server_name*/itim/self/ViewRequests.do (multiple requests view) • http://*server_name*/itim/self/ViewRequests.do?request=*requestID* (specific request view)
Approve and Review Requests	• http://*server_name*/itim/self/ReviewActivities.do (multiple activity view) • http://*server_name*/itim/self/ReviewActivities.do?activity=*activityID* (specific activity view)
Delegate Activities	http://*server_name*/itim/self/delegateActivities.do
Check out Credential (available only if shared access module is installed and configured)	http://*server_name*/itim/self/CheckoutSharedAccount.do
Check in Credential (available only if shared access module is installed and configured)	http://*server_name*/itim/self/CheckinSharedAccount.do

Table 13. Direct-access tasks and URLs (continued)

Task	URL
View Password (available only if shared access module is installed and configured)	http://*server_name*/itim/self/ViewPassword.do

Customizing person search capability

You can enable person search capability in the self-service user interface.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Person search capability is a powerful feature that you can use to select only people that match certain search criteria. Person search filters a wide range of search attributes.

The names of attributes take the form of ui.usersearch.attr.*attribute_name*=*attribute_name* in cases where *attribute_name* is common to all person and business partner person profiles. The *attribute_name* is a value that maps to that profile attribute. For example, ui.usersearch.attr.cn=cn searches by common name.

Some single attributes can map to multiple attributes if the profiles vary. In this case, the names of attributes take the form of ui.usersearch.attr.*attribute_name*=*profile1.attribute_name1*,*profile2.attribute_name1*

For example, ui.usersearch.attr.telephone=Person.mobile,BPPerson.telephonenumber would map the mobile number for the person profile and the telephone number for the business partner person profile.

The translated value of the attribute name is displayed in the search by attribute box. Do not specify attributes that cannot be searched by using plain text. For example, audio, photo, and other similar items.

To enable person search capability for the self-service user interface, complete these tasks:

Procedure

1. Make a backup copy of the SelfServiceUI.properties file in the *ISIM_HOME*\data directory.
2. Add or remove attributes in the SelfServiceUI.properties file under the **User Search** configuration section.
3. Restart the IBM Security Identity Manager application in WebSphere to make the changes effective.

Administrative console user interface customization

You can customize the administrative console user interface.

The IBM Security Identity Manager administrative console user interface is customizable. Customers can integrate a common corporate appearance and maintain the flexibility to do administrative identity tasks integral to their roles and responsibilities.

You can define and customize the administrative console interface in two ways, by using the built-in console framework and by directly modifying files installed within IBM Security Identity Manager:

- Built-in console features
  - Access control items (ACIs)
  - Views
- Modifiable files
  - Properties files
  - Image files

Back up any modifiable files for recovery purposes you make customization changes to IBM Security Identity Manager.

Configuration files and their descriptions

Configuration files define the appearance of the IBM Security Identity Manager administrative console user interface.

The following table lists the file names and describe their roles in the customization of IBM Security Identity Manager.

Table 14. Configuration property files and descriptions

File Name	File Description
ui.properties	Controls the appearance of the header, footer, and home page, and configures the title, number of pages that are displayed, and the number of search results returned.
helpmapping.properties	Controls the redirection and mapping of administrative console html help.

Backing up and restoring administrative console user interface configuration files

Before you begin customization of the administrative console user interface, back up all configuration files in IBM Security Identity Manager for later recovery purposes.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Create a directory named `custom` in the `WAS_PROFILE_HOME`\installedApps\ `node_name`\ITIM.ear\itim_console.war directory and store any new customization files in that custom directory.

Log in to each computer that is running Security Identity Manager and back up the following files:

- In the *ISIM_HOME*\data directory:
  - ui.properties
  - helpmappings.properties

About this task

Any changes made to properties files require you restart the Security Identity Manager application. For instance, upon recovering any properties files, complete these steps:

Procedure

1. Using the WebSphere administration console, click the **Applications** group in the left frame, and then click the **Enterprise Applications** link.
2. Select the check box next to the Security Identity Manager application, and click the **Stop** button.
3. After the application stops, select the check box next to the Security Identity Manager application, and click the **Start** button.
4. Verify that the recovery is complete by logging in to the administrative console user interface.

Customizing banner content

You can change the appearance of the administrative console user interface by customizing the banner.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

You can add or modify banner content to alter the appearance of the administrative console user interface.

The default banner area is defined in two files, a JSP file named banner.jsp and a properties file named ui.properties. The banner area consists of four parts:

- Banner launch link
- Banner launch logo
- Banner logo
- Banner background image

When customizing the banner, adjust the dimensions (width and height) of the components in the banner.jsp. Adjust these dimensions so that the custom logo image is sized properly without any distortion. Also ensure that the entire banner frame is not distorted.

You can change the banner launch link and logo by modifying the ui.properties file. If you want to modify the background image and banner logo, you must create a file to display your banner. This file can be either an HTML or a JSP banner file.

The following property keys in the `ui.properties` file define the banner launch link and banner launch logo. They also define the URL to the banner background image and logo.

Table 15. Banner property keys

Property key	Default value	Description
`enrole.ui.customerLogo.image`	`ibm_banner.gif`	Launch link logo, located in the `WAS_PROFILE_HOME\` `installedApps\`*`node_name`*`\` `ITIM.ear\itim_console.war\` `html\images` directory. You can also specify a URL pointing to the image file or put this file in the `WAS_PROFILE_HOME\` `installedApps\`*`node_name`*`\` `ITIM.ear\itim_console.war\` `custom` directory. If this directory does not exist, you must create it. Prefix the path name with `/itim/console/custom` in the ui.properties file. Specifying no value results in the default `ibm_banner.gif` file being displayed.
`enrole.ui.customerLogo.url`	`www.ibm.com`	Launch link URL. This value can be specified with or without the HTTP prefix. For instance, you can use `www.ibm.com` or `http://www.ibm.com` to specify the launch link URL.
`ui.banner.URL`	This value is left blank by default and displays the default banner area.	The HTML or JSP file that provides the banner logo, background image, and launch link and logo. You can enter either a URL or put this file in the `WAS_PROFILE_HOME\` `installedApps\`*`node_name`*`\` `ITIM.ear\itim_console.war\` `custom` directory If this directory does not exist, you must create it. Prefix the path name with `/itim/console/custom` in the `ui.properties` file.
`ui.banner.height`	48	Enter the pixel height of the banner.

To modify these files, complete these steps:

Procedure

1. Make backup copies of the files and store the files you want to modify in a temporary directory.
2. Edit the files in the temporary directory and copy the updated files back into the deployed WebSphere directory. You must restart the IBM Security Identity Manager application for these changes to take effect.

What to do next

Be sure to back up the custom version of the files you have created so your customizations are not lost.

Customizing footer content

You can change the appearance of the administrative console user interface by customizing the footer.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

You can add or modify footer content to alter the appearance of the administrative console user interface.

The default footer area is defined in the `ui.properties` file.

The following property keys in the `ui.properties` file define the footer and specify its visibility and height.

Table 16. Footer property keys

Property key	Default value	Description
ui.footer.isVisible	no	Specifies whether the footer is visible. By default the footer is disabled.
ui.footer.URL	This value is left blank by default.	Specifies the location of the HTML or JSP file that provides the footer. You can enter a URL. Alternatively, put this file in the `WAS_PROFILE_HOME\` `installedApps\node_name\` `ITIM.ear\itim_console.war\` `custom` directory (if this directory does not exist, you must create it), and prefix the path name with `/itim/console/custom` in the `ui.properties` file.
ui.footer.height	50	Enter the pixel height of the footer.

To modify these files, complete these steps:

Procedure

1. Make a backup copy of the `ui.properties` file and store the file in a temporary directory.

2. Edit the file in the temporary directory and copy the updated file back into the deployed WebSphere directory. You must restart the IBM Security Identity Manager application for these changes to take effect.

What to do next

Be sure to back up the custom version of the file you created so your customizations are not lost.

Customizing the administrative console home page

You can change the home page in the administrative console user interface with customization.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

The home page refers to the main page that gets loaded after a user logs in to the administrative console user interface.

Section and task definitions tie defined views to tasks, and group tasks into sections, also called task pages. These section and task definitions are defined in a properties file in the *ISIM_HOME*\data directory.

You can code direct links to tasks from the home page to administrative functions. Use JSP to generate dynamic HTML so administrative functions are limited to users with the appropriate authority.

To customize the home page, complete these steps:

Procedure
1. Make a backup copy of the ui.properties file in the *ISIM_HOME*\data directory.
2. Edit the ui.properties file. Modify the ui.homepage.path key, and save the file. Enter a URL of the HTML or JSP file that you are using for a home page. Alternatively, put this file in the *WAS_PROFILE_HOME*\installedApps*node_name*\ ITIM.ear\itim_console.war\custom directory (if this directory does not exist, you must create it), and prefix the file name with /itim/console/custom.
3. Restart the IBM Security Identity Manager application in WebSphere to make the changes effective.

Direct-access URL links to administrative console tasks
This section provides the direct URL access links to tasks in the administrative console user interface.

The following table displays the links to tasks that are supported for direct access, and that you can link to from the home page.

Table 17. Direct access tasks and links

Task	URL
Change Password	\Change Password\
Manage Roles	\Manage Roles\
Manage Organization Structure	\Manage Organization Structure\
Manage Users	\Manage Users\
Manage Services	\Manage Services\
Manage Identity Policies	\Manage Identity Policies\
Manage Password Policies	\Manage Password Policies\
Manage Adoption Rules	\Manage Adoption Rules\
Manage Recertification Policies	\Manage Recertification Policies\
Manage Provisioning Policies	\Manage Provisioning Policies\
Manage Service Selection Policies	\Manage Service Selection Policies\
Manage Account Request Workflows	\Manage Account Request Workflows\
Manage Access Request Workflows	\Manage Access Request Workflows\
Manage Groups	\Manage Groups\
Manage Access Control Items	\Manage Access Control Items\
Manage Views	\Manage Views\
Set Security Properties	\Set Security Properties\
Configure Forgotten Password Settings	\Configure Forgotten Password Settings\
Request Reports	\Request Reports\
Service Reports	\Service Reports\

Table 17. Direct access tasks and links (continued)

Task	URL
Audit and Security Reports	\Audit and Security Reports\
Custom Reports	\Custom Reports\
Report Properties	\Report Properties\
Configure Replication Schema	\Configure Replication Schema\
Design Reports	\Design Reports\
Manage Service Types	\Manage Service Types\
Design Forms	\Design Forms\
Set Workflow Notification Properties	\Set Workflow Notification Properties\
Configure Post Office	\Configure Post Office\
Manage Entities	\Manage Entities\
Manage Operations	\Manage Operations\
Manage Lifecycle Rules	\Manage Lifecycle Rules\
Manage Access Types	\Manage Access Types\
Configure Policy Join Behaviors	\Configure Policy Join Behaviors\
Configure Global Policy Enforcement	\Configure Global Policy Enforcement\
Import Data	\Import Data\
Export Data	\Export Data\
View Pending Requests by User	\View Pending Requests by User\
View All Requests by User	\View All Requests by User\
View Pending Requests by Service	\View Pending Requests by Service\
View All Requests by Service	\View All Requests by Service\

Table 17. Direct access tasks and links (continued)

Task	URL
View All Requests	View All Requests
View Activities	View Activities
View Activities by User	View Activities by User
Manage Delegation Schedules	Manage Delegation Schedules
About	About
Define Forgotten Password Questions	Define Forgotten Password Questions

Customizing the title bar

You can change the title bar shown in the web browser when you log in to the IBM Security Identity Manager administrative console.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To customize the title bar, complete these steps:

Procedure

1. Make a backup copy of the `ui.properties` file and store the file in a temporary directory.
2. Edit the `ui.titlebar.text` property with the title you want to use, and save the file. The default value is blank and displays the text IBM Security Identity Manager.
3. Copy the updated file back into the deployed WebSphere directory. You must restart the Security Identity Manager application for these changes to take effect.

What to do next

Be sure to back up the custom version of the files you created so your customizations are not lost.

Redirecting help content

You can redirect help requests to your own website to deliver custom help content for the administrative console user interface.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Editing the ready to use help content that is shipped with the administrative console user interface is not supported. But it is possible to redirect help requests to your own website to deliver custom help content.

The `helpmappings.properties` file specifies the base URL that help requests are sent to. These files are in the *ISIM_HOME*\data directory.

The following table shows the property and property description for help.

Table 18. Administrative help properties and description

Property	Description
url.contexthelp	Specifies the base URL to send help requests to. A blank value indicates that help goes to the default URL for the administrative console user interface.
Help ID mappings: helpID = relative page URL	The help mappings section maps IDs from specific pages to a relative URL sent to the help server.

The Help URL is the combination of the url.contexthelp + locale + relativeHelppageURL

For example:
```
url.contexthelp=http://myserver:80
locale = en_US
```

Note: Locale is determined by matching the current logged in user's browser settings with the currently installed IBM Security Identity Manager language packs.
```
loginID/relativeURL = login_help_url=ui/ui_eui_login.html
```

Therefore, the final URL is `http://myserver:80/en_US/ui/ui_eui_login.html`.

To redirect help, complete these steps:

Procedure

1. Make a backup copy of the `helpmappings.properties` file in the *ISIM_HOME*\data directory.
2. Change the `url.contexthelp` property in the `helpmappings.properties` file. Customers must not change the helpIDs. They are what the Security Identity Manager user interface panels use to find the appropriate help.
3. Update helpID mappings to use the relative URLs for your server.
4. Add pages to your server for the appropriate locales.
5. Restart the **ITIM** application in WebSphere for these changes to take effect.

Customizing the number of items displayed on pages

You can change the number of items displayed on pages.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

The following property keys in the `ui.properties` file define the number of list items displayed on the page and maximum number of search items that are returned.

Table 19. Panel parameters, default values, and descriptions

Property	Default value	Description
enrole.ui.pageSize	50	Specifies the number of list items displayed on a page.
enrole.ui.maxSearchResults	1000	Specifies the maximum number of search items returned.

Note: These changes can affect memory usage if set to excessive values.

To change page parameters, complete these steps:

Procedure

1. Make a backup copy of the `ui.properties` file in the *ISIM_HOME*\data directory.
2. Restart the IBM Security Identity Manager application in WebSphere to make the changes effective.

What to do next

Be sure to back up the custom version of the file so your customizations are not lost.

Configuring the Justification field in the user interface

You can add the **Justification** field to the user interface. You can also configure the **Justification** field to be a required field.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

By default, the **Justification** field is not displayed in the user interface. You can configure the properties so that the **Justification** field is displayed in the user interface. You can also configure the **Justification** field to be a required field.

The following table shows the properties, default values, and description of the parameters that are related to the **Justification** field. These properties settings affect the administrative console and self service user interface that contain the **Justification** field.

Table 20. Properties, default values, and descriptions

Property	Properties File	Default Value	Description
ui.displayJustification	ui.properties	false	Specifies whether the **Justification** field is displayed in the user interface.
enrole.justificationRequired	enRole.properties	false	Specifies whether the **Justification** field is displayed in the user interface as a required field.

To change the properties for the **Justification** field, complete these steps:

Procedure

1. Make a backup copy of the `ui.properties` and `enRole.properties` files in the *ISIM_HOME*\data directory.
2. Edit the files in a temporary directory and copy the updated file back into the *ISIM_HOME*\data directory

Option	Description
To configure the Justification field to be displayed but not required	In the `ui.properties` file, set `ui.displayJustification=true`.
To configure the Justification field to be displayed *and* to be required, you modify the	In the `enRole.properties` file, set `enrole.justificationRequired=true`. **Note:** You do not need to modify the `ui.properties` file. If the `enrole.justificationRequired` property is set to `true`, then the **Justification** field is displayed as a required field regardless of the setting for the `ui.displayJustification` property in the `ui.properties` file.

3. Refresh the IBM Security Identity Manager application in your web browser. The **Justification** field is displayed in the user interface pages for which it applies.

What to do next

Be sure to back up the custom version of the files so that your changes are not lost.

Related reference:

Required field properties
These properties are used to configure whether fields in the user interface are required to be completed by the user.

ui.properties
The ui.properties file specifies attributes that affect the operation and display of the Security Identity Manager graphical user interface.

Identity Service Center user interface customization

The Identity Service Center user interface is highly customizable. You can change most of the screen text, icons, graphics, help file content, and layout. You can also change the contents of many user interface elements, such as the home page, user cards, and access cards.

You can customize the Identity Service Center user interface in the following ways:
- Copying and modifying the customizable files that are installed with IBM Security Identity Manager.
- Replacing the icons and graphics.

When the customized files are placed in the appropriate location, the IBM Security Identity Manager server can find and use them.

Without customization, you can use the Identity Service Center user interface to achieve goals such as these:
- Request access to applications
- View your requests

Location of Identity Service Center customizable files

As an administrator, if you want to customize the Identity Service Center, you must know where to find the files that IBM provides. You must also know where to put the customized versions of those files.

Customizable files are maintained in folders under the WebSphere Application Server configuration directory. The exact location depends on whether you are using a single server or are using a managed-clustered WebSphere Application Server configuration. Check with your WebSphere Application Server administrator to determine which configuration you are using.
- For a single-server configuration, the customizable files are under the profile for the application server.

  WAS_HOME/profiles/*app-server*/config/cells/*cell_name*/itim/custom/ui
- For a managed-cluster configuration, the customizable files are under the profile for the deployment manager. The files are pushed to application servers during synchronization.

  WAS_HOME/profiles/*dmgr*/config/cells/*cell_name*/itim/custom/ui

The exact location depends on the use of a managed-clustered application server configuration. For a managed-cluster configuration, the customizable files are under the profile for the deployment manager. The files are pushed to application servers during synchronization.

Note: During the installation or maintenance upgrade, the customizable files that are provided by IBM are copied to an original folder under the .../itim/custom/ui folder. These files can be copied or used as a reference during the customization process. The files in the original folder are for reference only and are installed as Read-Only files. These files are only copies of the files that are used at run time. Changes to these files do not affect the Identity Service Center user interface.

See "Customizing Identity Service Center files" on page 47 for information about how to customize these files that IBM provides.

Maintenance upgrades to IBM Security Identity Manager add or replace files in the `original` folder so that the files are always the most recent versions.

The customizable files that are provided by IBM are organized into folders under the `.../itim/custom/ui/original` folder.

- Translatable files that contain screen text are under the `.../itim/custom/ui/original/nls` folder.
- Configuration files that contain non-translatable configuration properties are in the `.../itim/custom/ui/original/config` folder.
- Icons, graphics, and other image files are under the `.../itim/custom/ui/original/images` folder.
- HTML templates that contain no translatable text are in the `.../itim/custom/ui/original/template` folder.

The following table lists the customizable files that are provided by IBM.

Table 21. Types and locations of customizable files

Locations of customizable files	Descriptions

`nls/AdditionalInformation.properties`
`nls/advanceSearchDialog.properties`
`nls/BigCard.properties`
`nls/CardCustomValue.properties`
`nls/CardGrid.properties`
`nls/Category Display.properties`
`nls/common.properties`
`nls/CriteriaTextBox.properties`
`nls/DateTimeWidget.properties`
`nls/DesignatedMaeesgeArea.properties`
`nls/DualList.properties`
`nls/ExpiredPassword.properties`
`nls/filteringCardSelect.properties`
`nls/HCard.properties`
`nls/headerLabel.properties`
`nls/LoginHour.properties`
`nls/LoginPageCopyrightContent.properties`
`nls/LoginPageInfoContent.properties`
`nls/logMessages.properties`
`nls/operators.properties`
`nls/Picker.properties`
`nls/PickerPage.properties`
`nls/RequestAccess.properties`
`nls/RequestListCard.properties`
`nls/RequestStatusDetails.properties`
`nls/RequestStatusList.properties`
`nls/SearchCustomAttributes.properties`
`nls/SearchCustomValue.properties`
`nls/tmsMessagesUI.properties`
`nls/UILanguages.properties`
`nls/UMask.properties`
`nls/validatorMessage.properties`
	These files contain screen text that is translated into multiple languages to support globalization. Each of these files has a version for all of the supported locales.
	
	If you customize any of these files, you must also customize the locale-specific versions of the files. You must customize the files for all of the languages that you plan to support in your environment.

`config/Access.json`
`config/ActionDefinition.json`
`config/HeaderMenu.json`
`config/Homepage.json`
`config/Person.json`
`config/Search.json`
`config/UIconfig.properties`
`config/UIHelp.properties`
	These files contain configuration information that does not require language translation.

Table 21. Types and locations of customizable files (continued)

Locations of customizable files	Descriptions
images/approved.png images/companyLogo.gif images/favicon.ico images/getDetailsButton.png images/getDetailsButton_rtl.png images/identity.png images/more.png images/notprovisioned.png images/pending.png images/provisioned.png images/rejected.png images/access/iconAccessRoleAccess.gif images/access/iconAccessServiceAccess.gif images/access/iconApplicationAccess.gif images/access/iconDefaultAccess.gif images/access/iconMailGroup.gif images/access/iconProvideAccountInfo.png images/access/iconProvideAccountInfoRtl.png images/access/iconServiceAccess.gif images/access/iconSharedFolderAccess.gif images/homepage/network_nav.png images/homepage/RequestAccess.png images/homepage/ViewRequests.png images/status/request/fulfilled.png images/status/request/notfulfilled.png images/status/requests/partiallyfulfilled.png images/status/requests/pending.png	These files are the icons, images, and graphics that are displayed throughout the user interface. Subfolders are used to group related images together.
html/Login.html	This file is an HTML template that does not contain any text that requires language translation.
nls/html/LoginPageCopyrightContent.html nls/html/LoginPageInfoContent.html	These files are HTML files that might contain text that is translated into multiple languages to support globalization. Each of these files has a version for all of the supported locales. If you customize any of these files, you must also customize the locale-specific versions of the files. You must customize the files for all of the languages that you plan to support in your environment.
.../itim/custom/ui/css/Custom.css	This file is the custom stylesheet that you can modify to customize the style of login page, home page, and headers.
.../itim/custom/ui/images/	This location is where you can place your customized icons and images.

Table 21. Types and locations of customizable files (continued)

Locations of customizable files	Descriptions

`.../itim/custom/ui/original/nls`
 	
ISIM_HOME`/data/CustomLabels.properties`
 	
`I18NMESSAGES schema table`
	These locations are where you can customize labels.

`.../itim/custom/ui/template`	The files at this location contain templates for advanced customization of login page and headers.

Customizing Identity Service Center files

As a site administrator, you might want to customize Identity Service Center to meet your specific business needs. Customization involves either copying and modifying files that are provided by IBM or creating your own custom files to use in place of the IBM files. You must ensure that your new and modified custom files are placed in the correct location. Otherwise, the files cannot be found and used by Identity Service Center.

Before you begin

Customizing the Identity Service Center user interface requires access to files and folders under the WebSphere Application Server configuration folder of your IBM Security Identity Manager runtime environment. See "Location of Identity Service Center customizable files" on page 43 for the exact location of the files and folders to which you need access. To obtain access to the necessary files and folders, contact your system administrator.

About this task

The Identity Service Center can be customized in many ways. To customize a particular aspect of the Identity Service Center such as the login page, or the home page, see the appropriate customization instructions. Most customization tasks involve changing or providing replacements for one or more of the customizable files that are provided by IBM. This procedure describes how to create a custom file and where to place the custom file so that it can be used by the IBM Security Identity Manager server.

Note: In single-server WebSphere Application Server environments, IBM Security Identity Manager uses the custom file immediately after it is copied to its correct location in the `.../itim/custom/ui` folder. In managed-cluster environments, the custom file is used as soon as the configuration is synchronized to the application servers in the cluster. For significant customization, create and test the customized files when there are no active users of the Identity Service Center.

Many of the customizable files contain text that is translated into multiple languages for globalization. For customizable globalization files, a default file exists that is not locale-specific, such as `common.properties`. There are also locale-specific files for each supported language such as `common_fr.properties` for French and `common_ar.properties` for Arabic. The instructions in the following sections describe how to customize the default version of a globalization file only. If you choose to customize globalization files, follow the same instructions to customize the locale-specific versions of the files. You must customize the files for each language that you intend to support in your environment.

Procedure

1. To replace an image file that IBM provides such as an icon or other graphic, complete the following steps.

   a. In the `.../itim/custom/ui/original/` folder, locate the image that you want to replace. For example, `.../itim/custom/ui/original/images/identity.png`.

   b. Create your custom image. Use the same file name and type as the file that IBM provides. For example, `identity.png`.

   c. Copy the custom image to the `.../itim/custom/ui` folder. Use the same relative path and file name as the file that IBM provides in the `.../itim/custom/ui/original/` folder. For example, `.../itim/custom/ui/images/identity.png`

2. To create a custom version of an IBM text file, complete the following steps.

   a. In the `.../itim/custom/ui/original/` folder, identify the text file that you want to customize. For example, `.../itim/custom/ui/original/config/UIconfig.properties`.

   b. Copy the file to the `.../itim/custom/ui/` folder. Ensure that you keep the same relative path and file name. For example, `.../itim/custom/ui/config/UIconfig.properties`.

   c. Change the file permissions on the copied file so that it can be modified. The files that IBM provides are installed with Read-Only file permissions.

   d. Modify the copied file for your customization requirements.

What to do next

Start the Identity Service Center in a browser to verify that the customized file is being used. You can verify the file immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center.

Merging new and existing customized configuration files

To work with new features after you upgrade, you must merge the content of new configuration files with any existing, customized Identity Service Center configuration files. If you do not merge the new and existing customized configuration files, you get errors when you work with the upgraded version of Identity Service Center.

Before you begin

- See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for the location of the Identity Service Center customizable files and how to customize the files.

- If you did not previously customize the tertiary attribute in the `config/Access.Json`, you can see that the tertiary attribute values are modified from `"tertiary": [ "accessCategory", "entityProfile" ]` to new default value `"tertiary": [ "additionalInformation" ]`.

- If you did not previously customize the `config/HeaderMenu.json`, you can see that the new attribute `menuIconItem` is introduced. The tasks **View Access**, **Request Access**, and **Edit and Delete Access** are grouped under the **Manage Access** toolbar menu.

About this task

If you previously customized the configuration files that contain non-translatable configuration properties, complete the following steps.

Procedure

1. Merge the content of the new configuration files into the existing customized configuration files. For example, if you previously customized the file `.../itim/custom/ui/config/Person.json`, then merge it to the new`.../itim/custom/ui/original/config/Person.json` file.

2. Replace the existing customized configuration files in the directory `.../itim/custom/ui/config` with the merged files.

Results

You can now work with the new features by restoring the previous customization that you made.

User interface elements that are affected by view definitions

Defined views affect the visibility of the tasks that are displayed in the page header menus and on the home page of the Identity Service Center.

Page header menus

The page header menus of the Identity Service Center adapt to the user's views by showing only the tasks that are granted to the user. These tasks can be arranged in groups, with each group displayed as a drop-down menu on the page header. If the user is not granted any tasks in a group, the menu for that task group is not displayed on the page header. In the following scenarios, a task itself is displayed on the page header instead of a drop-down menu.

- If there is only one task in the group.
- If the user is granted only one task in the group.

Some tasks might not be displayed in the page header menus. The system administrator can choose not to include tasks in the header menus.
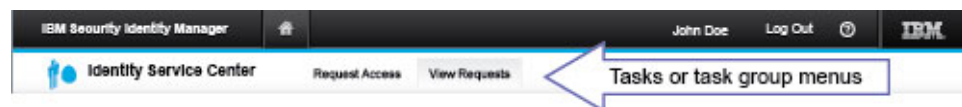
Home page

The content on the home page of the Identity Service Center user interface adapts to the user's views. The home page displays only the tasks that are granted to the user. Some tasks might not be displayed on the home page. The system administrator can choose not to display tasks on the home page.
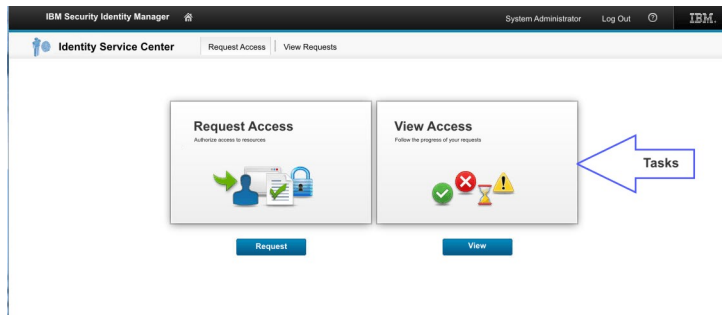
Note: If the user is not granted any tasks, the home page does not display after the user logs in to the Identity Service Center. An error message is displayed and the user must log out.

Enabling Identity Service Center as the default user interface

To enable the Identity Service Center as the default user interface, you must complete the configuration steps. The configuration steps provide the mechanism to hide the view of the self-service user interface.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Procedure

1. In the *ISIM_HOME*/data/ui.properties file, set the property ui.defaultui.redirectSelfToISC to true.
2. Optional: Set the language at these two locations because the languages set in both the Identity Service Center and self-service user interfaces might be different at few occasions.
   - On the **Language** menu in the Identity Service Center login page
   - In the browser
3. Optional: To ensure that users have access to the same functions in the Identity Service Center that they have in self-service user interface, manually configure the views. Configuring the view is applicable for each view that provides access to self-service function. See View management.
4. Optional: Customize the self-service user interface to hide the headers so that a user does not have both an Identity Service Center header and a self-service interface header. See "Self-service user interface customization" on page 1.
5. Optional: Configure to enable the forgotten password link on Identity Service Center login page.
   a. Log on to IBM Security Identity Manager administrative console.
   b. Select **Set System Security** > **Configure Forgotten Password Settings**.
   c. Select **Enable forgotten password authentication**.

   Note: A user must set answers for the security questions in the self-service user interface. If the forgotten password link is enabled, you must create a custom task for the forgotten password challenge behavior so that a user can update

the forgotten password information.
For more information about the forgotten password, see Forgotten password settings.

6. Optional: If you want to view the status of the requests that are placed through custom tasks that launch the self-service user interface, you must create a new custom task that points to the relevant view request status.

Login page customization

Users can enter their user names and passwords in the Login page to authenticate with the IBM Security Identity Manager Server and access the Identity Service Center.

The Login page can be customized in various ways to meet the needs of your business. They are as follows:

- Change the logo to your company or product image. The image dimensions are resized to fit the allocated space.
- Modify or replace the site information on the right side of the Login page.
- Modify the text for the **Help?** and **Learn More** hyperlinks or remove the hyperlinks from the Login page.
- Change the copyright information at the bottom of the Login page.
- Customize the validation messages for the Login page.

Customizing the Login page

You can customize the authentication Login page to meet the requirements of your organization. For example, you can customize your company or product logo. You can also define more details, or change the labels of fields on the page.

Before you begin

You must have read or write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details about where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

You can customize the appearance and content of the Login page to meet your needs.

Procedure

1. Open any text editor to edit a property file.
2. Optional: Make a custom copy of the `nls/LoginText.properties` file.
3. Optional: Edit the `nls/LoginText.properties` file to customize the various fields on the Login page. The `nls/LoginText.properties` file contains a set of properties that define the text strings that are displayed in the various fields of the Login page. You can customize the Login page text by changing the text that is associated with these properties, as follows:

    LOGO_ALT_TEXT
    Specify custom text to display if the logo image is missing.

    HELP_TAG
    Specify a custom label for the **Help** hyperlink.

LEARN_MORE
> Specify custom text for the **Learn More** hyperlink.

LOG_IN Specify a custom label for the **Log in** button.

PASSWORD
> Specify a custom label for the **Password** field.

PRODUCT_NAME
> Specify a custom title for the product name.

USER_ID
> Specify a custom label for the **User ID** field.

INVALID_USERNAME
> Specify a custom message that is displayed when an invalid user name is entered.

INVALID_PASSWORD
> Specify a custom message that is displayed when an invalid password is entered.

LANGUAGE_LABEL
> Specify the label for the language selection field.

4. Optional: Customize the company or product image on the Login page. You can customize the image on the login page by using either of the following ways:

   - Create a custom image file in `GIF` format with the file name `companyLogo.gif`. Place the image file in the `/images` folder of your customizable files.
   - Create a custom image file with any file name, such as `someImage.jpg`. Place the image file in the `/images` folder of your customizable files. Then, create a custom copy of the `config/UIconfig.properties` file and modify the `LOGO_IMAGE` property to specify the name of the image file such as `someImage.png`.

5. Optional: Modify the copyright information at the bottom of the Login page. The copyright information that is provided by IBM is delivered as an HTML template, `nls/html/LoginPageCopyrightContent.html`, and a properties file, `nls/LoginPageCopyrightContent.properties`. The properties file contains the text that is substituted into the HTML template. To customize the copyright information on the Login page, you can take one of the following actions:

   - Continue to use the HTML template that is provided by IBM. Create a custom copy of the template substitutions `nls/LoginPageCopyrightContent.properties` file. Then, modify the `FOOTER_TEXT` property in the `nls/LoginPageCopyrightContent.properties` file to the text you want to be displayed at the bottom of the Login page.
   - Create a custom version of the `nls/html/LoginPageCopyrightContent.html` template and modify it to use whatever HTML formatting and template substitutions that you want. Then, create a custom copy of the `nls/LoginPageCopyrightContent.properties` file and add or modify the properties in it to provide the necessary substitutions for your custom HTML template.
   - If you do not want to use an HTML template with substitutions, you can create a custom copy of the `nls/html/LoginPageCopyrightContent.html` template file. Then, replace the template substitution references with the actual copyright text that you want to display.

6. Optional: Modify the site information on the right side of the Login page. The site information that is provided by IBM is delivered as an HTML template,

`nls/html/LoginPageInfoContent.html`, and a properties file, `nls/LoginPageInfoContent.properties`. The properties file contains text that is substituted into the HTML template. To customize the site information on the Login page, you can take one of the following actions:

- Continue to use the HTML template that is provided by IBM. Create a custom copy of the template substitutions `nls/LoginPageInfoContent.properties` file. Then, modify the properties in the `nls/LoginPageInfoContent.properties` file to provide the text that you want to be displayed on the right side of the Login page.
- Create a custom version of the `nls/html/LoginPageInfoContent.html` template and modify it to use whatever HTML formatting and template substitutions that you want. Then, create a custom copy of the `nls/LoginPageInfoContent.properties` file and modify or replace the properties in it to provide the necessary substitutions for your custom HTML template.
- If you do not want to use an HTML template with substitutions, you can create a custom copy of the `nls/html/LoginPageInfoContent.html` template file. Then, replace the template substitution references with the actual site information text that you want to display.

7. Save and close each of the custom files when you are finished customizing for the Login page.
8. Start the Login page again.

Results

The Login page now displays the customization that you made.

What to do next

Use the custom Login page for your requirements. You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center.

Advanced customization of login page and headers

Use templates to add or customize static elements on the login page and headers.

You must have read or write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details about where these files are located. Contact your system administrator if you do not have the necessary permissions.

The following templates are provided for customization in the `WAS_HOME`/profiles/`app_server`/config/cells/`cell_name`/itim/custom/ui/template folder:

- `Login.html`
- `Header.html`
- `ForgotPasswordHeader.html`

You can add or customize static elements (that do not involve scripting), such as links, images, and icons. For example, you can add a custom help desk link or a custom image to the login page or headers.

For customizing icons in the template files, complete the following steps:

1. In the `.../itim/custom/ui/original/images` folder, locate the image that you want to customize. For example, `.../itim/custom/ui/original/images/identity.png`.

2. Copy the image that you want to customize to the `.../itim/custom/ui/images` folder.

3. Use the absolute paths while you add an image directly. For example, `./custom/ui/images/identity.png`.

If you want to style the new static element, use the custom stylesheet. See "Customizing cascading styles" on page 62.

After you customize the file, validate the `html` to ensure that the syntax is correct.

The following example shows the original `Header.html` template before customization:

```
<div>
 <div class="idxHeaderPrimary afterLogin">
  <div class="idxHeaderPrimaryInner"
   data-dojo-attach-point="primaryBannerNode">
   <ul class="idxHeaderPrimaryActionsLeading"
    data-dojo-attach-point="_leadingGlobalActionsNode">

    <div class="shadowUpper"></div>

   </ul>

   <ul class="idxHeaderPrimaryActionsTrailing"
    data-dojo-attach-point="_trailingGlobalActionsNode">
   </ul>

  </div>

 </div>
 <div class="itimSSNavHeader">

  </div>
</div>
```

In the following example, links are added to the customized header template.

```
<a href="http://www.google.com" class="linkTest" target="_blank">User Agreement</a>

<div>
 <div class="idxHeaderPrimary afterLogin">
  <div class="idxHeaderPrimaryInner"
   data-dojo-attach-point="primaryBannerNode">
   <ul class="idxHeaderPrimaryActionsLeading"
    data-dojo-attach-point="_leadingGlobalActionsNode">

    <div class="shadowUpper"></div>

   </ul>
   <ul> <a href="http://localhost:9080/itim/self" class="selfcare_link" target="_blank">SelfCare</a>
       </ul>    <ul class="idxHeaderPrimaryActionsTrailing"
    data-dojo-attach-point="_trailingGlobalActionsNode">
   </ul>

  </div>

 </div>
```

```
<div class="itimSSNavHeader">
  <ul> <a href="http://localhost:9080/itim/console" class="console_link" target="_blank">AdminUI</a>
        </ul>
  </div>
</div>
```

Customizing the page header

The Identity Service Center user interface has a header at the top of the page. The header provides menus that are used to navigate to tasks that the user is authorized to perform. The page header can be customized in various ways to meet the needs of your organization.

Before you begin

You must have read and write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details about where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

The Identity Service Center page header is divided into two areas. The upper portion is the primary header. It displays the product name and the current user, as well as the **Home** and **Log Out** shortcuts, the header logo, and the **Help** menu. The lower portion is the secondary header. It displays a product image, the name of the active task, and menus of tasks the user is authorized to perform. You can customize the appearance of the page header to suit your needs.

The task menus on the page header adapt to the user's authorized views so that only tasks the user is allowed to perform are shown. You can customize the organization of tasks in the page header menus.

The IBM Security Identity Manager administrator console is used to manage view definitions by:
- Assigning tasks to views.
- Associating groups with those views.
- Managing the members of the groups.

IBM provides a set of ready-to-use Service Center tasks. You can also create custom tasks to launch your own web applications from the Identity Service Center user interface. Both the tasks that are provided by IBM and your custom tasks can be displayed in the page header menus. However, the method of customizing the appearance and organization of tasks is different for each of these types of tasks.

Procedure

1. Optional: Customize the product name in the primary area of the page header or the home page name in the secondary area of the page header:
   a. If you did not already do so, make a custom copy of the nls/headerLabel.properties file.
   b. Open the custom copy of the nls/headerLabel.properties file in a text editor.
   c. Modify these properties to suit your needs.

      identityManager
           Specify the custom text to display for the product name field.

SVCENTER_HOMEPAGE
> Specify the custom text to display for the home page name.

2. Optional: You can customize the logo that is displayed in the primary area of the page header by using either of these methods.

   - Create a custom image file in PNG format with the same name as the image provided by IBM, `headerLogo.png`. Place the image file in the `images` folder of your customizable files. The custom image is used in place of the `headerLogo.png` image that is provided by IBM.

   - If you did not already do so, make a custom copy of the `config/UIconfig.properties` file. Create a custom image in any image format with any file name, for example `customLogo.jpg`. Place the image file in the `images` folder of your customizable files. Edit the custom copy of the `config/UIconfig.properties` file. Change the value of the **HEADER_LOGO_IMAGE** property to specify the name of your custom image, such as `customLogo.jpg`. Save the file.

3. Optional: You can customize the alternate text for the header logo in the primary area of the page header. The alternate text is displayed when the user's browser is set to not show images. Screen readers for visually impaired users also read the alternate text to indicate what the image represents.

   a. If you did not already do so, make a custom copy of the `nls/headerLabel.properties` file.

   b. Open the custom copy of the `nls/headerLabel.properties` file in a text editor.

   c. Modify the value of the **headerLogoAltText** property to define the alternate text for the header logo.

4. Optional: You can customize the image that is displayed in the secondary area of the page header, by using either of the following ways:

   - Create a custom image file in PNG format with the same name as the image provided by IBM, `identity.png`. Place the image file in the `images` folder of your customizable files. The custom image is used in place of the `identity.png` image that is provided by IBM.

   - If you did not already do so, make a custom copy of the `config/HeaderMenu.json` file. Create a custom image file in any image format and with any file name, for example `customIcon.jpg`. Place the image file in the `images` folder of your customizable files. Edit the custom copy of the `config/HeaderMenu.json` file. Change the value of the **secondaryIcon** field to specify the location and name of your custom image, such as `custom/ui/images/customIcon.jpg`. Save the file.

5. Optional: Customize the appearance and organization of the tasks that are provided by IBM in the page header menus.

   The `config/HeaderMenu.json` file defines the appearance and organization of the page header menus. The contents of this file are maintained in JavaScript Object Notation (JSON) format, which is a way of representing structured data. The text labels for the drop-down menus and the task names in the menus are defined in the `nls/headerLabel.properties` file.

   The **secondaryNavigation** section of the `config/HeaderMenu.json` file contains a **menus** subsection.

```
"secondaryNavigation": {
       . . .
       "menus": [
           . . .
       ]
    }
```

Each area in this **menus** subsection describes one of the menus in the secondary area of the page header.

```
{
  "labelKey": "manageAccess",
  "icon": "custom/ui/images/header/tab_RequestAccess.png",
  "menuItemIcon":"/itim/ui/custom/ui/images/header/dd_requestAccess.png",
  "menuItems": [
        {
             "actionId": "SVCENTER_REQUEST_ACCESS"
        }
    ]
},
```

labelKey

Specifies the name of the property in the `nls/headerLabel.properties` file whose value is displayed for the menu. To customize the labels for the menus and tasks, if you did not previously complete this task, make a custom copy of the `nls/headerLabel.properties` file. Find the corresponding property from the `config/HeaderMenu.json`, such as **manageAccess**, in the custom `nls/headerLabel.properties` file. Change the value.

Note: If a menu contains only a single task, the **labelKey** for the menu is not used. That task is displayed on the page header instead of a drop-down menu with a single menu item.

icon Specifies an icon that is displayed at the left corner of the task header.

menuItemIcon

Specifies an icon that is displayed for each of the tasks in the menu.

menuItems

Defines the list of tasks that are displayed in the menu.

actionId

Specifies the name of the property in the `nls/headerLabel.properties` file whose value is displayed for the task. To customize the label for a task, if you did not previously complete this task, make a custom copy of the `nls/headerLabel.properties` file. Find the corresponding property from the `config/HeaderMenu.json`, such as **SVCENTER_REQUEST_ACCESS**, in the custom `nls/headerLabel.properties` file. Change the value.

To change the organization of the menus and the tasks in each menu:

a. If you did not previously complete this task, make a custom copy of the `config/HeaderMenu.json` file.

b. Open the custom copy of the `config/HeaderMenu.json` file in a text editor.

c. Edit the **menus** subsection of the **secondaryNavigation** section of the file.

d. Move the menu sections so that they are in the order that you want them to be displayed on the page header.

e. Add or move tasks in the **menuItems** subsection so that they are in the order that you want them to be displayed in the drop-down menu.

f. Save the file.

6. Optional: Customize the appearance and organization of custom tasks in the page header menus. The IBM Security Identity Manager administration console can be used to create custom tasks to launch your own web applications. To create these custom tasks, the administrator specifies parameters that define the appearance of the task, such as:

- Label
- Description
- Page header menu in which the custom task is displayed

For information about creating custom tasks, see View management.

Results

The appearance of the page header is changed to reflect your customizations.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment.

In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center. Log in to the Identity Service Center and verify that the home page reflects the customizations that you made.

Customizing the home page

The home page of the Identity Service Center displays a list of tasks that the user is permitted to perform. The home page can be customized in various ways to meet the needs of your organization.

Before you begin

You must have read and write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details of where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

The IBM Security Identity Manager administrator console is used to manage view definitions by:
- Assigning tasks to views.
- Associating groups with those views.
- Managing the members of the groups.

IBM provides an initial set of ready-to-use Service Center tasks. You can also create custom tasks to launch your own web applications from the Identity Service Center user interface. Both the tasks that are provided by IBM and your custom tasks can be displayed on the home page. However, the method of customizing of the appearance and organization of tasks is different for each of these types of tasks. For information about adding custom tasks, see View management.

Each task on the home page is represented by a card. The card provides information about the task, such as a task name, a description, and an image. You can customize the appearance and organization of these tasks on the home page to suit your needs.

The tasks on the home page adapt to the user's authorized views so that only tasks the user is allowed to perform are shown.

The `config/Homepage.json` file defines the appearance and organization of the tasks that IBM provides on the home page. The contents of this file are maintained in JavaScript Object Notation (JSON) format, which is a way of representing structured data. Each section in the `config/Homepage.json` file is enclosed in braces and defines the appearance of one task on the home page. For example, the following section of the `config/Homepage.json` file defines the **Request Access** task.

```
{
     "actionId":"SVCENTER_REQUEST_ACCESS",
     "btnLabel": "SVCENTER_REQUEST_ACCESS_BUTTON",
     "desc":   "SVCENTER_REQUEST_ACCESS_DESC",
     "img": "./custom/ui/images/homepage/requestOthersAccess.png"
  },
  {
     "actionId":"SVCENTER_REQUEST_ACCESS_FOR_MYSELF",
     "btnLabel": "SVCENTER_REQUEST_ACCESS_BUTTON",
     "desc":   "SVCENTER_REQUEST_ACCESS_FOR_MYSELF_DESC",
     "img": "./custom/ui/images/homepage/requestMyAccess.png"
  },
```

actionId
> Specifies the name of the property in the `nls/headerLabel.properties` file whose value is displayed for the task name.

btnLabel
> Specifies the name of the property in the `nls/headerLabel.properties` file whose value is displayed for the label of the button for the task.

desc Specifies the name of the property in the `nls/headerLabel.properties` file whose value is displayed for the description of the task.

img Specifies the location of the image that is displayed for the task.

The steps in this procedure use the **Request Access** section of the `config/Homepage.json` file as an example. The actual property names in the `nls/headerLabel.properties` file depend on the section of the `config/Homepage.json` file that you are customizing.

Procedure

1. Optional: Customize the text for a task that IBM provides on the home page.
   a. If you have not previously completed this task, make a custom copy of the `nls/headerLabel.properties` file.
   b. Open the custom copy of the `nls/headerLabel.properties` file in a text editor.
   c. Optional: To customize the text for the task name, find the property that matches **actionId** field, such as **SVCENTER_REQUEST_ACCESS**. Change the value.
   d. Optional: To customize the text for the task button, find the property that matches **btnLabel** field, such as **SVCENTER_REQUEST_ACCESS_BUTTON**. Change the value.
   e. Optional: To customize the text for the task description, find the property that matches **desc** field, such as **SVCENTER_REQUEST_ACCESS_DESC**. Change the value.
   f. Save the file.
2. Optional: Customize the image that is displayed for the task that IBM provides. You can use either of two methods.

- Create a custom image file in PNG format with the same name as the image provided by IBM, such as `requestAccess.png` in the previous example. Place the image file in the `images/homepage` folder of your customizable files. The custom image is used in place of the `requestAccess.png` image that is provided by IBM.
- If you have not previously completed this task, make a custom copy of the `config/Homepage.json` file. Create a custom image file in any image format and with any file name, for example `customImage.jpg`. Place the image file in the `images/homepage` folder of your customizable files. Edit the custom copy of the `config/Homepage.json` file. Change the value of the **img** field for the task to specify the location and name of your custom image `custom/ui/images/homepage/customImage.jpg`. Save the file.

3. Optional: Change the organization of the tasks that IBM provides on the home page.
   a. If you have not previously completed this task, make a custom copy of the `config/Homepage.json` file.
   b. Open the custom copy of the `config/Homepage.json` file in a text editor.
   c. Move the sections so that they are in the order that you want them to be displayed on the home page.
   d. Save the file.

4. Optional: Customize the appearance and organization of custom tasks on the page header menus. The IBM Security Identity Manager administration console can be used to create custom tasks to launch your own web applications. To create these custom tasks, the administrator specifies parameters that define the appearance of the task, such as:
   - Label
   - Description
   - Icon

   For information about creating custom tasks, see View management.

Results

The appearance of the home page is changed to reflect your customizations.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center. Log in to the Identity Service Center and verify that the home page reflects the customizations that you made.

Customizing the scope of user lists for tasks

You can customize the definition of a task to limit the list of users that are displayed in the task. The definition can limit the list to include only the users that are relevant for the current Identity Service Center user.

Before you begin

You must have read and write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details of where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

The home page and page header menus of the Identity Service Center display tasks that the user is allowed to perform. Some tasks, such as the Request Access task, involve the selection of one or more users from a list. For some organizations, this list of users can be large.

You can customize the definition of tasks so that the user list shows only the users that are relevant for the current Identity Service Center user. For example, you might want the list of users to be restricted to only those users in the department that is managed by the current user.

The `config/ActionDefinition.json` file defines how tasks are launched when the user selects them. The contents of this file are maintained in JavaScript Object Notation (JSON) format, which is a way of representing structured data. Each section in this file defines the launch information for one task, as shown here for the Request Access task.

```
  "SVCENTER_REQUEST_ACCESS": {
      "actionType": "CreateFlow",
      "urlHash":"requestAccess",
      "properties": {
          "widgetPath":
              "com/ibm/isim/ui/util/uiflow/requestaccess/RequestAccessFlow",
          "widgetArgs": { "personFilterId": "" }
      }
  },
```

The **properties** section contains a **widgetArgs** field that defines a list of JavaScript variables that are passed to the task when it is launched. The value of the **personFilterId** variable specifies the `filterId`. The `filterId` is configured in the `custom/rest/searchfilter.json` file. This filter is used by the task when it looks for users that are relevant to the current Identity Service Center user. The value can be customized to suit the needs of your organization by modifying the attribute **baseFilter** for the configured `filterId` in the `custom/rest/searchfilter.json` file. For example, see Filter configuration for REST search services.

Procedure

1. Create a custom copy of the `config/ActionDefinition.json` file.
2. Locate the section of this file that describes the launch information for the task to be customized, such as **SVCENTER_REQUEST_ACCESS**.
3. Modify the value of the **personFilterId** variable of the **widgetArgs** field in the properties section to specify the `filterId` for the user list in the task. See Filter configuration for REST search services.
4. Save and close the file.

Results

When the task with the customized user scope is launched, the list of users is restricted to only those users that match the specified filter. Only those users are displayed on the Select user page.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere product documentation at http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp. Log in to the Identity Service Center and verify that the user scope reflects the customizations that you made.

Customization of styles, icons, and labels

Identity Service Center user interface is highly customizable. You can change icons, fonts, color, and labels.

When the customized files are placed in the appropriate location, the IBM Security Identity Manager server can find and use them. To customize the Identity Service Center, you must know where to find the files that IBM® provides. See "Location of Identity Service Center customizable files" on page 43.

You can customize Identity Service Center user interface in the following ways:
* Copying and modifying the customizable files that are installed with IBM Security Identity Manager.
* Replacing icons and graphics. See "Customizing icons or images" on page 64.
* Applying customized styles. See "Customizing cascading styles" and "Sample CSS file for customizing styles" on page 65.
* Modifying labels. See "Customizing labels" on page 63.

Customizing cascading styles

You can customize the styles for the login page, home page, and headers. To apply customized styles, you must create a cascading stylesheet (CSS) file that contains your customized styles and store it in the appropriate location.

About this task

By default, the `css` folder is provided by IBM Security Identity Manager at `.../itim/custom/ui/original`. You must create a `css` file that contains customized style data. Refer to the sample `css` file that is available in the `.../itim/custom/ui/original/css` folder. See "Sample CSS file for customizing styles" on page 65.

This procedure describes how to apply the customized styles and where to place the `css` file so that it can be used by the IBM Security Identity Manager server.

Restriction: You can use the custom `css` to change the look and feel of existing elements, such as their font, color, size, and so forth. However, you cannot add elements to the user interface or modify the contents of the user interface.

The following list includes several properties of the user interface elements that you can customize:

- Background color, text color, gradient color, border color, and so forth
- Image size, text size, element size, and so forth
- Element alignment and position
- Hide unwanted elements
- Add a custom link in the header

Procedure

1. Create a `css` file named `Custom.css`

   Note: The `css` file name must be `Custom.css`.

2. Open the `css` file in a text editor, and define any new styles that you want to apply to the product user interface.

3. Store the `Custom.css` in the `.../itim/custom/ui/css` folder.

What to do next

Start the Identity Service Center in a browser to verify that the customized styles are being used.

Customizing labels

Labels and text that are displayed in the Identity Service Center user interface are customizable. The label and text customization task involves providing replacements for one or more of the customizable files that are provided by IBM Security Identity Manager.

Before you begin

Create the backup copy of original files and then replace the customized labels or text.

About this task

This procedure describes how to replace the customizable labels or text and where to place the customized file so that it can be used by the IBM Security Identity Manager server.

You can customize labels that exist at the following locations:
- `.../itim/custom/ui/original/nls` folder
- *ISIM_HOME*`/data/CustomLabels.properties`
- `I18NMESSAGES` schema table

If the labels do not exist at these specified locations, they cannot be customized.

Procedure

1. In the `.../itim/custom/ui/original/nls` folder, locate the label or text that you want to replace from one or more property files. For example, `viewProfile=View Profile` from the `.../itim/custom/ui/original/nls/headerLabel.properties`.

2. Create your custom label or text. For example, `viewProfile=View User Profile`.

3. Copy the custom label or text to the `.../itim/custom/ui/nls` folder. Use the same relative path and file name as the file that IBM provides in the `.../itim/custom/ui/original/` folder. For example, `viewProfile=View User Profile` in the `.../itim/custom/ui/nls/headerLabel.properties`.

4. Customize labels in the *ISIM_HOME*/data/CustomLabels.properties file.
   a. Edit the CustomLabels.properties file.
   b. Change the value of the property to the text that you want to display for the label.
   c. Restart the IBM Security Identity Manager application server after you modify the CustomLabels.properties file.
5. Customize labels in the I18NMESSAGES schema tables.
   a. Connect to the IBM Security Identity manager DB2 database, for example, isimdb.
   b. Go to *DB_name* > **Schemas** > *ITIMUSER (your ISIM DB user name)* > **Tables** > **I18NMESSAGES**.
   c. Open the table in edit mode.
   d. Locate the label that you want to customize under the **Message** column and update the attribute. For example, you can edit Gecos (comments) label to TestLinux Gecos (comments) for Linux Profile.
   e. Save the changes.
   f. Open the account form of Linux in the Identity Service Center. The updated label is displayed with the customization.
   g. Restart the IBM Security Identity Manager application server for the changes to take effect.

What to do next

Start the Identity Service Center in a browser to verify that the customized label is being used.

Customizing icons or images

All icons or images that are used in the Identity Service Center are customizable. The icon customization task involves providing replacements for one or more of the customizable files that are provided by IBM Security Identity Manager.

Before you begin

- Create the backup copy of original files before you customize the images.
- For the customized image, use the same resolution and format as the original graphic image.

About this task

This procedure describes how to replace the customizable icon file and where to place the customized icon file so that it can be used by the IBM Security Identity Manager server.

Procedure

1. In the .../itim/custom/ui/original/images folder, locate the image that you want to customize. For example, .../itim/custom/ui/original/images/identity.png.
2. Copy the image that you want to customize to the .../itim/custom/ui/images folder. Use the same relative path as the file that IBM Security Identity Manager provides in the .../itim/custom/ui/original/ folder. For example, .../itim/custom/ui/images/identity.png
3. Modify the copied image or create your custom image. You have two options:

- You can use the same file name and type as the file that IBM Security Identity Manager provides. For example, `identity.png`.
- You can use a different name for the image file. If you use a different file name, then you must also modify the custom cascading stylesheet (CSS) to point to the image file. For more information, see "Customizing cascading styles" on page 62.

What to do next

Start the Identity Service Center in a browser to verify that the customized file is being used.

Sample CSS file for customizing styles

To apply customized style to Identity Service Center, you must create a cascading stylesheet (CSS) file. This topic provides information about four sample CSS file.

Custom.css

This sample `Custom.css` file is the main `css` file that imports other `css` files.

```
@import "LoginPage.css";
@import "HomePage.css";
@import "Header.css";
```

LoginPage.css

This sample file contains style data for the login page.

```
/*
This page contains Sample css file that provides information to customize styles in the Identity Service Center login page.
You must create a new Custom.css file in the .../itim/custom/ui/css directory.
For a trial, import this Sample.css file in the Custom.css file that you created to view the effect of the customized styles in the Identity Service Center login page.
*/

.oneui .loginForm .page{
 max-height: auto;
}

/*
To change product logo size edit below class:
*/
.page .product_logo{
 height: 40px;
 width: 120px;
}

/*
To change Header title properties edit below class:
*/
.loginColumn .headertTitle{
 font-size:1.67em;
 font-weight:normal;
 font-color:#222222;
 margin-bottom: 10px;
}

/*
To change language field properties edit below class:
*/
.languageField{
 padding:10px 0px !important;
 float: right;

}

.dj_ie9 .languageField{
 margin: 50px !important;
}

.dijitRtl .languageField {
    float: left;
}

/*
To change Login form properties edit below class:
*/
.oneui .loginForm {
   text-align: center;
   background: #A9A9F5;
   background: -moz-linear-gradient(center top, #e8e8e8 0%, #b9b9b9 100%);
   background: -webkit-linear-gradient(top, #e8e8e8 0%, #b9b9b9 100%);
   background: -ms-linear-gradient(top, #e8e8e8 0%,#b9b9b9 100%);
   background: linear-gradient(to bottom, #e8e8e8 0%,#b9b9b9 100%);
   filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#e8e8e8', endColorstr='#b9b9b9',GradientType=0 );
   box-shadow: 2px 0px 4px #000 inset;
}

/*
To change Login Page properties edit below class:
*/
.loginForm .page{
 display: inline-block;
 text-align: left;
 margin: auto;
 padding: 20px;
 /* max-height: 48em; */
```

```
    border: medium solid #999999;
    background: #CEECF5;
}

/*
To change Login button properties edit below class:
*/
.oneui .loginBtn .idxSpecialButton .dijitButtonNode {
    background-color: greenyellow;
    background-image: -moz-linear-gradient(top, green, red);
    background-image: -webkit-gradient(linear,0% 0%,0% 100%,from(#008abf),to(#085884));
    background-image: -webkit-linear-gradient(top, red 0%, #085884 100%);
    background-image: -o-linear-gradient(top, #008abf 0%, #085884 100%);
    background-image: -ms-linear-gradient(top, #008abf 0%, #085884 100%);
    background-image: linear-gradient(top, #008abf 0%, #085884 100%);
    border: 1px solid #0a5f8e;
    color: #ffffff;
    text-shadow: 0px 1px 0px rgba(0, 0, 0, 0.35);
}

.dijitRtl .page{
 text-align: right;
}

.mainLogin{
 margin-top: 0px;
 white-space: nowrap;
}

.loginColumn{
 display: inline-block;
 padding-right: 8px;
 min-width: 0px;
}

.dijitRtl .loginColumn{
 display: inline-block;
 padding-left: 8px;
 padding-right: 0px;
}

/*
To change Login page Footer properties edit below class:
*/
.loginForm .footer .footerText {
  display: inline-block;
  width: 100%;
  word-wrap: break-word;
}
```

HomePage.css

This sample file contains style data for the home page.

```
/*
To change the Background color of Home Page edit below class
*/
.SVCENTER_HOMEPAGE .itimui{
    height: 100%;
    background: #E0F8F7; /* Old browsers */
    background: -moz-radial-gradient(center, ellipse cover, #F2F2F2 1%, #E0F8F7 100%); /* FF3.6+ */
    background: -webkit-gradient(radial, center center, 0px, center center, 100%, color-stop(1%,#F2F2F2), color-stop(100%,#E0F8F7)); /* Chrome,Safari4+ */
    background: -webkit-radial-gradient(center, ellipse cover,#F2F2F2 1%,#E0F8F7 100%); /* Chrome10+,Safari5.1+ */
    background: -o-radial-gradient(center, ellipse cover,#F2F2F2 1%,#E0F8F7 100%); /* Opera 12+ */
    background: -ms-radial-gradient(center, ellipse cover, #F2F2F2 1%,#E0F8F7 100%); /* IE10+ */
    background: radial-gradient(ellipse at center,#F2F2F2 1%,#E0F8F7 100%); /* W3C */
}

/*
To change the Navigation card properties edit below class
*/
.navcard {
    display: inline-block;
    width: 312px;
    margin: 0 0 0 20px;
}

/*
To change the Navigation card Border edit below class
*/
.navCardBorder {
    box-shadow: 0 2px  12px rgba(0,0,0,2);
}

/*
To change the property of navigation card new window icon edit below class
*/
.navcard .navCardBorder .textArea .newWindowIcon {
    position: absolute;
 right: 7px;
 top: 7px;
}

.dijitRtl .navcard .navCardBorder .textArea .newWindowIcon {
    right: auto;
    left: 7px;
}

/*
To change the Navigation card properties edit below class
*/
.navcard .navCardBorder .textArea {
    border: 1px solid #fff;
    outline: 2px solid rgba(0, 0, 0, 0.2);
    cursor: pointer;
    font-size: 1.5em;
    height: 143px;
    overflow: hidden;
    padding: 22px;
    position: relative;
    vertical-align: top;
    width:266px;
    margin-bottom: 22px;
    background: #81BEF7; /* Old browsers */
    background: -moz-linear-gradient(top, #81BEF7 0%, #A9D0F5 100%); /* FF3.6+ */
    background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#81BEF7), color-stop(100%,#A9D0F5)); /* Chrome,Safari4+ */
    background: -webkit-linear-gradient(top, #81BEF7 0%,#A9D0F5 100%); /* Chrome10+,Safari5.1+ */
    background: -o-linear-gradient(top, #81BEF7 0%,#A9D0F5 100%); /* Opera 11.10+ */
    background: -ms-linear-gradient(top, #81BEF7 0%,#A9D0F5 100%); /* IE10+ */
    background: linear-gradient(to bottom, #81BEF7 0%,#A9D0F5 100%); /* W3C */
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#81BEF7', endColorstr='#A9D0F5',GradientType=0 ); /* IE6-9 */
}
```

```
/*
To change the selected Navigation card properties edit below class
*/
.navcard .navCardBorder .textArea:hover {
    border: 1px solid #fff;
    outline: 2px solid rgba(0, 0, 0, 0.2);
    background: #81BEF7 ; /* Old browsers */
    background: -moz-linear-gradient(top, #81BEF7  0%, #58D3F7 20%); /* FF3.6+ */
    background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#81BEF7 ), color-stop(20%,#58D3F7)); /* Chrome,Safari4+ */
    background: -webkit-linear-gradient(top, #81BEF7  0%,#58D3F7 20%); /* Chrome10+,Safari5.1+ */
    background: -o-linear-gradient(top, #81BEF7  0%,#58D3F7 20%); /* Opera 11.10+ */
    background: -ms-linear-gradient(top, #81BEF7  0%,#58D3F7 20%); /* IE10+ */
    background: linear-gradient(to bottom, #81BEF7  0%,#58D3F7 20%); /* W3C */
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#81BEF7 ', endColorstr='#58D3F7',GradientType=0 ); /* IE6-9 */
}

.navcard .navCardBorder .textArea:focus {
 outline: 2px solid #8CCAE3;
 border: 1px solid #0297D1;
}

.dijitRtl .navCardBorder .textArea .name {
    text-align: right;
}

.navcard .navCardBorder .textArea .desc {
    color: #222;
    font-size: .7em;
    word-wrap: break-word;
    margin-bottom: 10px;
    text-align: left;
}

.navcard .navCardBorder .textArea .name {
    font-size: 1.35em;
    font-weight: bold;
    word-wrap: break-word;
    text-align: left;
    color: #045FB4;
    margin-bottom: 10px;
}

.dijitRtl .navCardBorder .textArea .desc {
    text-align: right;
}

.navcard .navCardBorder .textArea .iconStrip {
    position: absolute;
    bottom: 15px;
    left: 22px;
}

.dijitRtl .navcard .navCardBorder .textArea .iconStrip {
    right: 22px;
}

.navcard .navCardBorder .textArea .navCardCount {
 display: inline-block;
    position: relative;
  width: 100px;
  left: 100px;
   top: 25px;
}

:lang(ja) .navcard .navCardBorder .textArea .navCardCount {
 top: 0px;
}

.dijitRtl .navcard .navCardBorder .textArea .navCardCount {
 right: 70px;
}

.navcard .navCardBorder .textArea .count {
  color: #ee5334;
   font-size: 48px;
   font-weight: bold;
   text-align: center;
   left: 55px;
}

.navcard .navCardBorder .textArea .navCardCount .countText {
 font-family: 'Arial';
 font-size: 13px;
 font-weight: bold;
 word-wrap: normal;
 color: #00649b;
 position: relative;
 bottom: 15px;
 padding: 10px 0px 0px 0px;
 text-align: center;

.oneui .dijitMenuBar .dijitMenuItem {
 padding: 6px 10px 7px 10px;
 margin: 0 3px 0 2px;
 color: #A9BCF5;
 margin-left: -1px;
}

}
.oneui .dijitMenuTable .oneuiMenuColumn tr:first-child .dijitMenuArrowCell,
.oneui .dijitMenuTable .oneuiMenuColumn tr:first-child .dijitMenuItemIconCell,
.oneui .dijitMenuTable .oneuiMenuColumn tr:first-child .dijitMenuItemLabel,
.oneui .dijitMenuTable .oneuiMenuColumn tr:first-child .oneuiMenuHeadingLabel{
  border-top: 1px solid #A9BCF5;
}
```

Header.css

This sample file contains style data for the header.

```
/*
To change the Primary Header color edit below class
*/
.oneui .idxHeaderContainer .idxHeaderPrimary {
  background-color: #0B4C5F;
  background-image: none;
}

/*
To change the Header logo and Size edit below class
*/
.oneui .idxHeaderContainer .idxHeaderPrimary .idxHeaderLogo {
    background-image: url('/itim/ui/custom/ui/images/headerLogo.png');
```

```
    background-color: rba(0,0,0,0);
    background-repeat: no-repeat;
    background-attachment: scroll;
    background-size: 100px 25px;
}

/*
To change the Header logo and Size edit below class
*/
.dijitRtl .oneui .idxHeaderContainer .idxHeaderPrimary .idxHeaderLogo {
    float: center;
    background-image: url('/itim/ui/custom/ui/images/headerLogo.png');
    background-color: rba(0,0,0,0);
    background-repeat: no-repeat;
    background-attachment: scroll;
    background-size: 100px 25px;
}

/*
To change the secondary Header properties edit below class
*/
.itimSSNavHeader {
 background: #A9D0F5; /* Old browsers */
 background: -moz-linear-gradient(top, #A9D0F5 0%, #819FF7 80%, #A9D0F5 100%); /* FF3.6+ */
 background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#A9D0F5), color-stop(80%,#819FF7), color-stop(100%,#A9D0F5)); /* Chrome,Safari4+ */
 background: -webkit-linear-gradient(top, #A9D0F5 0%,#819FF7 80%,#A9D0F5 100%); /* Chrome10+,Safari5.1+ */
 background: -o-linear-gradient(top, #A9D0F5 0%,#819FF7 80%,#A9D0F5 100%); /* Opera 11.10+ */
 background: -ms-linear-gradient(top, #A9D0F5 0%,#819FF7 80%,#A9D0F5 100%); /* IE10+ */
 background: linear-gradient(to bottom, #A9D0F5 0%,#819FF7 80%,#A9D0F5 100%); /* W3C */
 filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#A9D0F5', endColorstr='#819FF7',GradientType=0 ); /* IE6-9 */
    width: 100%;
    height: 34px;
    padding-top: 7px;
    padding-left: 0;
    min-width: 980px;
}

.itimSSNavHeader .MenuBar{
    background:transparent;
    border:none;
    display: inline-block;
}
.oneui .itimui .idxHeaderContainer .idxHeaderNavigationHome {
    width: 15px;
}

.oneui .itimui .idxHeaderContainer .dijitMenuItemHover .idxHeaderNavigationHome {
    width: 15px;
}

.dijitRtl .oneui .itimui .idxHeaderContainer .idxHeaderNavigationHomeButtonOnly.cssForHebrew {
 background-position: -151px -5px;
}

.dijitRtl .oneui .itimui .idxHeaderContainer .dijitMenuItemHover .idxHeaderNavigationHomeButtonOnly.cssForHebrew{
 background-position: -217px -5px;
}

.oneui .itimui .idxHeaderContainer .idxHeaderPrimary .idxHeaderDropDown .idxHeaderDropDownArrow{
 display:none;
}

.oneui .itimui .idxHeaderContainer .idxHeaderPrimary li.idxHeaderPrimaryAction .idxHeaderUserName{
 cursor:default;
}
```

Customizing the Select Me option

You can customize the Select Me option in Identity Service Center so that the
current logged in user gets selected automatically.

About this task

This procedure describes how to customize the Select Me option to be enabled
automatically in the Request Access flow and the Edit and Delete Access flow
scenarios. With this customization, you are not required to select a logged in user
on the Home page of the Identity Service Center. The current logged in user is
automatically selected.

Procedure

1. If you did not already do so, make a custom copy of the .../itim/custom/ui/
   original/config/UIconfig.properties file. Copy the UIconfig.properties file
   to the .../itim/custom/ui/config/ folder. Use the same relative path as the file
   that IBM Security Identity Manager provides in the .../itim/custom/ui/
   original/ folder.

   For example, .../itim/custom/ui/config/UIconfig.properties.

2. In a text editor, open the UIconfig.properties file and locate the
   ui.userPicker.defaultSelfSelect property.

3. Set the value of the ui.userPicker.defaultSelfSelect property to true.

4. Save and close the UIconfig.properties file.

What to do next

Log in to the Identity Service Center and verify that the current user is selected automatically in the Request Access flow and Edit and Delete Access flow.

Customizing the context root

You can customize the context root of Identity Service Center.

About this task

This procedure describes how you can customize the context root of Identity Service Center. By default the context root is /itim/ui. You can customize this context root to be, for example, /custom/ui.

Important: After you migrate from a previous fix pack version to the next fix pack version, the Identity Service Center URL context root customization reverts back to the original values for the Identity Service Center user interface and help. However, the Identity Service Center subform URL customization is not affected. Hence, you must specify the URL context root customization values again for the Identity Service Center user interface and help after migration.

Procedure

1. Log on to the WebSphere Application Server console with administrator credentials.
2. Go to **Applications** > **Application Types** > **Business-level applications**.
3. Click **IdentityServiceCenterApplication**.
4. Click **com.ibm.icim_CU.eba**.
5. Under **Additional Properties**, click **Context root for web modules**.
6. Update the context root mapping for com.ibm.isim.ui with your custom context root.

   For example, if you update com.ibm.isim.ui from /itim/ui to /custom/ui, the Identity Service Center URL becomes https://*hostname:port*/custom/ui.
7. Customize the mapping for com.ibm.isim.ui.help with your custom context root. For example: /custom/uihelp.

   Note: If you update com.ibm.isim.ui.help, you must also update the helpBaseUrl property in the UIHelp.properties file. The custom settings for this property must match the settings that you specify for com.ibm.isim.ui.help in the WebSphere Application Server console.
   a. Copy the .../itim/custom/ui/original/config/UIHelp.properties file to the .../itim/custom/ui/config folder.
   b. Open the .../itim/custom/ui/config/UIHelp.properties file in a text editor and update the helpBaseUrl property with your custom settings.
   c. Save and close the UIHelp.properties file.

   For more information, see "Redirecting help content" on page 93.
8. In the WebSphere Application Server console, click **OK** to save changes.
9. Go to **Applications** > **Application Types** > **Business-level applications**.
10. Select **IdentityServiceCenterApplication** and restart the application.
11. Go to **Applications** > **Application Types** > **WebSphere enterprise applications** > **ITIM** > **Context Root For Web Modules**.

12. Update the context root mapping for `isim_isc_subform` with your custom context root.

    For example: `custom/ui/subforms`.

    Note: The following requirement must be met when you customize the context root for subforms:

    You must use the same context root that you set for `com.ibm.isim.ui` and then append /subforms. For example, if you customized the context root for `com.ibm.isim.ui` to `/custom/ui`, then for `isim_isc_subform` the context root should be `/custom/ui/subforms`.

13. In the WebSphere Application Server console, click **OK** to save changes.
14. Go to **Applications** > **Application Types** > **WebSphere enterprise applications**.
15. Select **ITIM** and restart the application.

Request Access wizard

You can customize the user interface characteristics of the Request Access wizard to suit your needs.

The following items in the Request Access wizard can be changed or customized:
- The appearance and content of the user cards
- The appearance and content of the access cards
- The text and styling of badges on access cards
- The access card selection limit
- The search control properties

Customizing a user card in the Request Access wizard

The first step in the Request Access wizard is used to select the user for whom access is being requested. The set of users to choose from is displayed as a collection of user cards that are arranged in a grid. You can customize the information that is displayed in the user cards, and also how the user cards in the grid can be sorted.

Before you begin

You must have read or write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details about where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

A *user card* is like a business card for people in your organization. The information that is displayed on a user card is arranged into several areas. You can customize which user attributes are displayed in each of the areas to meet your needs.

The `primary` area of the user card displays the most important user attribute, such as the user name. The information in this area displays at the top of the card and in the largest font. Only one user attribute can be assigned to the `primary` area, but you can choose a different attribute for each of the user profiles that are defined in your environment.

The `secondary` area of the user card displays the next most important user attribute, such as the user email address. The information in this area is displayed just under the `primary` area and in a smaller font than the `primary` area. Only one user attribute can be assigned to the `secondary` area, but you can choose a different attribute for each of the user profiles that are defined in your environment.

The `tertiary` area of the user card displays extra information about the user, such as the user title, department name, or sponsor name. The information in this area is displayed just under the `secondary` area and in a smaller font than the `secondary` area. Multiple user attributes can be assigned to the `tertiary` area. You can choose different sets of attributes for each of the user profiles that are defined in your environment. Each assigned attribute is given a label, such as **Title** or **Sponsor** that is displayed on the user card with the attribute value. The label is to help the user understand the information that is displayed on the card.

The *icon* area of the user card displays an image that is associated with the user, such as the user picture from your organization directory.

Procedure

1. Optional: Customize the user attributes that are displayed in the different areas of user cards and whether sorting on the information in those areas is supported. Make a custom copy of the `config/Person.json` file and open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is a way of representing structured data.

   The `primary` section of this file contains `attribute` and `sort` subsections. For example:

   ```
   "primary": {
       "attribute": {
           "default": "name",
           "Person" : "CN",
           "BPPerson" : "CN"
       },
       "sort": {
           "enabled": true,
           "labelKey" : "name"
       }
    },
   ```

   In the `attribute` section, define the user attribute to display in the `primary` area of the user card. You can choose a different user attribute for each user profile that is defined in your environment. You must always set a default user attribute to use for any user profiles that are not explicitly defined. In the earlier example, the default user attribute is `"name"`, but the attribute for users in the `"Person"` and `"BPPerson"` user profiles is `"CN"`.

   Note: Ensure that the `primary` section is defined with a valid LDAP attribute for the specified profiles or for a default attribute that is common across all profiles.

   In the `sort` section, you can enable or disable sorting of the user card that is based on the information in the `primary` area of the card. If you enable sorting, `"enabled": true`, the uppercase value of the **labelKey** field is used to look up the display string for this sort option in the customizable `nls/Picker.properties` file. In this example, the **labelKey** value `NAME` is looked up as a property in the `nls/Picker.properties` file to find the sort option string to display.

   The `secondary` section of this file is identical to the `primary` section. For example:

```
"secondary": {
    "attribute": {
        "default": "mail",
        "Person": "manager.name"
    },
    "sort": {
        "enabled": true,
        "labelKey" : "contactInfo"
    }
},
```

In the `attribute` section, define the user attribute to display in the `secondary` area of the user card. You can choose a different user attribute for each user profile that is defined in your environment. You must always set a default user attribute to use for any user profiles not explicitly defined. In the earlier example, the default user attribute is `"name"`, but the attribute for users in the `"Person"` user profile is `"manager.name"`.

In the `sort` section, you can enable or disable sorting of the user card that is based on the information in the `secondary` area of the card. If you enable sorting, `"enabled": true`, the uppercase value of the **labelKey** field is used to look up the display string for this sort option in the customizable `nls/Picker.properties` file. In this example, the **labelKey** value `CONTACTINFO` is looked up as a property in the `nls/Picker.properties` file to find the sort option string to display.

The `tertiary` section of this file contains an `attributes` section. The `attributes` section is used to define the list of user attributes to be displayed in the `tertiary` area of the user card. For example:

```
"tertiary": {
    "attributes": {
        "default": [ "title", "department" ],
        "BPPerson":[ "ersponsor.name" ]
    }
},
```

The attributes to be displayed are separated by commas and enclosed in square brackets. You can choose a different set of user attributes for each user profile that is defined in your environment. You must always set a default list of user attributes to use for any user profiles that are not explicitly defined. In the earlier example, the default list of user attributes is `[ "title", "department" ]`, but the attribute list for users in the `"BPPerson"` user profile is `[ "ersponsor.name" ]`.

Sometimes the attribute that you want to display is not an attribute of the user, but it might be an attribute of an object that is related to the user. For example, a user might have attributes that are called `"manager"` or `"ersponsor"` that are actually references to related users, namely the manager or sponsor of this user. To display an attribute like `"name"` from the related user in this user card, you can use the dotted notation that is shown in the earlier examples:

```
"manager.name"
"ersponsor.name"
```

2. Optional: Customize the labels that are displayed with the user attributes in the `tertiary` area of the user card. Make a custom copy of the `nls/Picker.properties` file and open the file with a text editor. The properties in this file define the text that displays in various parts of the user selection step of the Request Access wizard.

User attributes assigned to the `tertiary` area of the user card are displayed with a label to help the user understand what information they see. For example, if the `config/Person.json` file contains this definition for the `tertiary` section:

```
"tertiary": {
    "attributes": {
        "default": [ "title", "department" ],
        "BPPerson":[ "ersponsor.name" ]
    }
},
```

Then, for the users in the `BPPerson` user profile, the `tertiary` field of the user card might be displayed as follows:

`Sponsor: John Doe`

To customize the label for a user attribute in the `tertiary` area of the user card, look for a property in the `nls/Picker.properties` file. The property must match the uppercase form of the user attribute name that is specified in the `tertiary` section of the `config/Person.json` file. For example, ERSPONSOR.NAME. If this property does not exist in the file, add a property with this name. Customize this property value to specify the string that you want to display as the user attribute label in the `tertiary` area of the card.

3. Customize the text that is displayed in the sort option list for the `primary` or `secondary` areas of the user card. Make a custom copy of the `nls/Picker.properties` file and open the file with a text editor. The properties in this file define the text that is displayed in various parts of the user selection step of the Request Access wizard.

   You can enable sorting of the user cards that are based on information in the `primary` and `secondary` areas of the user card. For example, if the `config/Person.json` file contains the following definition for the `primary` and `secondary` sections.

```
"primary": {
    "attribute": {
        "default": "name",
        "Person" : "CN",
        "BPPerson" : "CN"
    },
    "sort": {
        "enabled": true,
        "labelKey" : "name"
    }
},
"secondary": {
    "attribute": {
        "default": "mail",
        "Person": "manager.name"
    },
    "sort": {
        "enabled": true,
        "labelKey" : "contactInfo"
    }
},
```

   Then, sorting of user cards is enabled for both the `primary` and `secondary` areas of the user card. The set of user cards has a sort control at the top that displays as follows:

   `Sort By: Name, Contact Information`

   Note: Sorting is not supported for attributes from objects that are related to the user, such as "manager.name". If any attributes that are specified in the `primary` section are from related objects, then the sort control does not include an option to sort on the `primary` area of the user card. Similarly, if any attributes that are specified in the `secondary` section are from related objects, then the sort control does not include an option to sort on the `secondary` area of the user card.

You can customize the text to display in the list of sort options for the `primary` or `secondary` areas of the user card. To customize the text, look for a property in the `nls/Picker.properties` file. The property must match the uppercase value of the **labelKey** of the corresponding section of the `config/Person.json` file. For example, `NAME` or `CONTACTINFO` in the earlier example. If this property does not exist in the file, add a property with this name. Customize the value of this property to specify the string that you want to display in the list of sort options.

4. Optional: Customize the icon area of the user card to display an image for the associated user. Make a custom copy of the `config/Person.json` file and open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is a way of representing structured data.

   The `icon` section of this file contains an attribute subsection. For example:

   ```
   "icon": {
       "attribute": {
           "default": "erimageuri",
           "BPPerson": null
       }
    }
   ```

   In the `attribute` section, define the user attribute that contains the location of the icon or image to display on the user card. You can choose a different user attribute for each user profile that is defined in your environment. You must always set a default user attribute to use for any user profiles that are not explicitly defined. If some user profiles do not have an image attribute, you can specify `null` to indicate no image to be displayed for users in that profile. In the earlier example, the default user attribute is `erimageuri`, but no image is displayed for users in the `BPPerson` user profile.

   See "Customizing the server to generate user image URIs" on page 75 for information about how to configure a plug-in for the IBM Security Identity Manager Server that can dynamically generate the location of an image for the `erimageuri` attribute by using the values of attributes that are associated with users.

5. Customize the display value for user attributes with values that are not intuitive.

   You might want to display some user attributes on a user card, but the value of these user attributes is not intuitive to the user. For example, there might be a user attribute name such as `employeeType` whose value is encoded as `"a"` for active employees, `"r"` for retired employees or `"p t"` for part-time employees. Displaying the actual value of this attribute on the user card might not be intuitive to the user.

   To customize the displayed values for some user attributes, make a custom copy of the `nls/CardCustomValue.properties` file and open the file with a text editor. The properties in this file define the custom text that is displayed in place of the actual values for various user attribute and value combinations. For example, to define the display text for the values of the `employeeType` user attribute, you can add or modify the properties in this file as follows:

   ```
   employeeType.a=Active
   employeeType.r=Retired
   employeeType.p__DELIMITER__t=Part-time
   ```

   With these assigned values, the user card displays `Active` when the *employeeType* value is `"a"`, `Retired` when the *employeeType* value is `"r"`, and `"Part-time"` when the *employeeType* value is `"p t"`.

   Note: The property names in this file cannot contain spaces. If any of the possible user attribute values contain a space, you must replace it with the

special character sequence `__DELIMITER__`. See the earlier example for reference. The `employeeType` value of "p t" is represented by a property name of `employeeType.p__DELIMITER__t`.

Results

The appearance of the user cards on the user selection step of the Request Access wizard is changed to reflect the customization that you made.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center. Log in to the Identity Service Center. Start the Request Access wizard and verify that the appearance of the user cards reflects the customization that you made.

Customizing the server to generate user image URIs

The IBM Security Identity Manager Server can be customized to dynamically generate the location (URI) for the user images that are shown on user cards.

Before you begin

Depending on your system customization, you might not have access to this task. Contact your system administrator to obtain access to this task, or to have someone complete it for you.

About this task

The user cards that are shown in the request access wizard can be configured to show an image that is associated with each user. The IBM Security Identity Manager Server provides a plug-in that can be used to dynamically generate the location (URI) of a user image. It is based on one or more attributes of the user.

Procedure

1. Open the `enroleExtensionAttributes.properties` file in the *ISIM_HOME*/data directory.
2. Uncomment the following line in the file to enable the default plug-in.

`person.extension.classname = com.ibm.itim.dataservices.extensions.plugins.PersonExtensionPlugin`

3. Set a value for the URI of the picture of each user. For example, if all your people images are stored on the web server, `images.myserver.com` under the `/uid` directory, then the configuration is as shown in the following example.

   `plugin.person.erImageURI=http://images.myserver.com/uid/${uid}.jpg`

   Note: Variables that refer to other attributes can be included in the URI, for example, **${uid}**. The variables are replaced with the real attribute values of the user at run time.

4. Optional: Set a value for the default URI in case you cannot do any substitution. For example, if the **uid** attribute is not set for the user, then you cannot substitute any other values. As a result, the default URI is returned.

   `plugin.person.erImageURI.default=http://images.myserver.com/default.jpg`

5. When you finish your customization, save and close the property file.

6. Log in to the Identity Service Center user interface.
7. From the Home page, click **Request Access** to open the Select user page.

Results

Your customization updates are shown in the user card in the Select user page with the associated picture.

What to do next

You can also write a custom plug-in that generates the picture URI for each person. See the Readme.html in the examples directory at *ISIM_HOME*/extensions/6.0/ examples/dataservices for instructions about compiling the supplied example plug-in.

Customizing an access card in the Request Access wizard

The second step in the Request Access wizard is used to select the accesses that are requested for a user. The set of access items to choose from is displayed as a collection of access cards that are arranged in a grid. You can customize the information that is displayed in these access cards.

Before you begin

You must have read or write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details about where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

An *access card* is like a brochure for the access items in your organization. The information that is displayed on an access card is arranged into several areas. You can customize which access attributes to display in each of these areas to meet your needs.

The following access attributes are available to be displayed on the access card:
- accessName
- description
- additionalInformation
- tags

Note: The tags attribute refers to the Search terms that are defined for the access item.

The primary area of the access card displays the most important access attribute, such as the access name. The information in this area is displayed at the top of the card and in the largest font. Only one access attribute can be assigned to the primary area.

The secondary area of the access card displays the next most important access attribute, such as the access description. The information in this area is displayed just under the primary area and in a smaller font than the primary area. Only one access attribute can be assigned to the secondary area.

The `tertiary` area of the access card displays extra information about the access item, such as the additional information. The information in this area is displayed just under the `secondary` area and in a smaller font than the `secondary` area. The `tertiary` area of the access cards displays the additional information about the attributes.

The `image` area of the access card displays an icon that is associated with the access item.

Procedure

1. Optional: Customize the access attributes that are displayed in the different areas of access cards. Make a custom copy of the `config/Access.json` file and open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is way of representing structured data.

   The `primary` field specifies the name of the access attribute to display in the `primary` area of the access card. For example:

   ```
   "primary" :"accessName",
   ```

   You can specify a different attribute to display in the `primary` area to meet your needs.

   The `secondary` field specifies the name of the access attribute to display in the `secondary` area of the access card. For example:

   ```
   "secondary": "description",
   ```

   You can specify a different attribute to display in the `secondary` area to meet your needs.

   The `tertiary` field specifies the access attribute to display in the `tertiary` area of the access card. For example:

   ```
   "tertiary": [ "additionalInformation" ],
   ```

   You can choose a different set of access attributes to display in the tertiary area to meet your needs. The attributes to be displayed are separated by commas and enclosed in square brackets.

2. Optional: Customize the labels that are displayed with the access attributes in the `tertiary` area of the user card. Make a custom copy of the `nls/Picker.properties` file and open the file with a text editor. The properties in this file define the text that is displayed in various parts of the access selection step of the Request Access wizard.

   Access attributes assigned to the `tertiary` area of the access card are displayed with a label to help the user understand what information is displayed. For example, the `config/Access.json` file contains the following definition for the `tertiary` section:

   ```
   "tertiary": [ "additionalInformation" ],
   ```

   To customize the label for an access attribute in the `tertiary` area of the access card, search for a property in the `nls/Picker.properties` file. The property must match the uppercase form of the access attribute name that is specified in the `tertiary` section of the `config/Access.json` file. For example, `additionalInformation`. If this property does not exist in the file, then add a property with this name. Customize this property value to specify the string that you want to display as the label of the access attribute in the `tertiary` area of the card.

3. Customize the text that is displayed in the sort option list for the `primary`, `secondary`, or `tertiary` areas of the access card. Make a custom copy of the `nls/Picker.properties` file and open the file with a text editor. The properties in this file define the text that is displayed in various parts of the access selection step of the Request Access wizard.

   Sorting of access cards is only supported for the `accessName`, `description`, and `additionalInformation` attributes. If the attribute in the `primary` area is supported for sorting, then it is displayed as the first choice in the sort option list. If the attribute in the `secondary` area is supported for sorting, then it is displayed as the next choice in the sort option list. If any of the attributes in the `tertiary` area are supported for sorting, then they are displayed next in the sort option list. The support is only up to a maximum of three sort options. For example, the `config/Access.json` file contains the following definition:

   ```
   "primary": "accessName",
   "secondary": "description",
   "tertiary": ["additionalInformation"],
   ```

   Then, sorting on the access name, description, and additional information attributes is supported. The set of access cards has sort control at the top that is represented as follows:

   ```
   Sort By: Name, Description, Additional Information
   ```

   To customize the text that is displayed in the list of sort options, search for a property in the `nls/Picker.properties` file. This property must match the uppercase form of the corresponding attribute name. For example, `accessName`, `description`, or `additionalInformation`. If this property does not exist in the file, add a property with this name. Customize this property value to specify the string that you want to display in the list of sort options.

4. Customize the image area of the access card to display an icon for the associated access item. Make a custom copy of the `config/Access.json` file and open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is a way of representing structured data.

   The `image` field specifies whether an image is to be displayed with each access item. For example:

   ```
   "image": "icon"
   ```

   This condition specifies that access icons must be displayed on access cards, when the appropriate image file can be found. If you do not want to display images on access cards, remove the `"image": "icon"` from the `config/Access.json` file.

   Images for access items can be defined for each individual access item, or for access categories. When you configure an access category image, then it displays for any access items in the category that do not have their own image explicitly defined. IBM Security Identity Manager includes default images for the predefined access categories, but you can provide custom images for these access categories, and custom images for individual access items. By convention, images for access items are maintained in the `images/access` folder of your customizable files. For example, the image for the `Application` access category is `images/access/iconApplicationAccess.gif`.

   - To define a custom image for one of the predefined access category images, create the image in GIF format by using the naming convention `icon<access-Category>Access.gif`. *<access-Category>* is the access category to

which the image is applicable. For example, `iconApplicationAccess.gif`. Place the custom image in the `images/access` folder of your customizable files.

- To define a custom image for a customer-defined access category, create the image in GIF format by using the naming convention `icon<access~category~hierarchy>Access.gif`. Place the custom image in the `images/access` folder of your customizable files. If your site administrator defined access categories in a hierarchy, then the GIF name must reflect that hierarchy by using "~" characters. For example, if a `Finance` category is defined as a child of the `Application` category, then the image file must be called `iconApplication~FinanceAccess.gif`.

- To define a custom image for a specific access item, create the new image file in any image format. Use any file name that you choose for the image file. Place the image file in the `images/access` folder of your customizable files. Your site administrator can then specify this image file location in the Access Information page of the service that is associated with the access item. For example, if you create an image that is called `iconMyApplicationAccess.jpg`, the image location is specified in the **Access icon** > **Icon URL** field as:

Results

The appearance of the access cards on the access selection step of the Request Access wizard is changed to reflect the customization that you made.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center. Log in to the Identity Service Center. Start the Request Access wizard, and verify that the appearance of the access cards reflects the customization that you made.

Customizing badges on access cards in the Request Access wizard

The second step in the Request Access wizard is used to select the accesses being requested for a user. The set of access items to choose from is displayed as a collection of access cards that are arranged in a grid. Access cards can be annotated with highlighted text called *badges*. Badges are used to alert the user to special considerations that are associated with the access, such as risk, data sensitivity, or regulatory compliance requirements. You can customize the text that is displayed for badges on access cards, and the style of the badges.

Before you begin

Depending on your system customization, you might not have access to this task. Contact your system administrator to obtain access to this task, or to have someone complete it for you.

About this task

A site administrator or a service owner uses the IBM Security Identity Manager Console to create badges and associate those badges with access items.

The badge text is defined as either a fixed string such as `High Risk` or the name of a property in the `CustomLabels.properties` file such as `$highrisk`. If a fixed string was defined as the badge text, then it cannot be customized. But if a property name was defined, you can customize the text that is displayed on the badge by modifying the value of the property.

The badge class is selected from one of the *Cascading Style Sheets* (CSS) style classes that are defined in the `Badge.css` file.

Procedure

1. Optional: Customize the text that is displayed for a badge on an access card. Consult your site administrator or service owner to determine the property name that is defined as the text for the badge that you want to customize. For this example, assume that the badge text was specified as `$highrisk`. So the property name is `highrisk`.

   Open the `CustomLabels.properties` file in the *ISIM_HOME*/data directory, where *ISIM_HOME* is the IBM Security Identity Manager installation directory.

   Open the `CustomLabels.properties` file.

   Locate the property name that is associated with the badge you want to customize, such as `highrisk`. If the property does not exist in the `CustomLabels.properties` file, then create a new property for the property name.

   Change the value of the property to the text that you want to display for the badge.

2. Optional: Customize the style for a badge on an access card. Changing the style for badges is an advanced topic that requires a working knowledge of *Hypertext Markup Language* (HTML) and *Cascading Style Sheets* (CSS). IBM Security Identity Manager contains several predefined CSS classes for badges. These classes might be suitable for your organization, but you can change these predefined classes or add new classes to meet your needs.

   To change the badge style for an existing CSS class, open the `Badge.css` file in the *ISIM_HOME*/data directory, where *ISIM_HOME* is the IBM Security Identity Manager installation directory. Locate the CSS class definition for the badge that you want to change. Modify the style attributes that are associated with the CSS class to suit your needs. For example, to change the style of a badge that is associated with the `green` badge class, you can do the following actions:

   - Find the `.badge.green` CSS selector in the `Badge.css` file.
   - Modify the style attributes that are associated with it.

   To create a new badge class that can be assigned to access entities, open the `Badge.css` file. Create a CSS selector or copy an existing CSS selector for the new badge class. CSS selectors for badges must always be in the form, `.badge.`*customName*, where *customName* is the name of the new badge class. The IBM Security Identity Manager Console displays this *customName* in the drop-down list of badge classes when the site administrator or service owner assigns badges to access entities. Modify the style attributes associated with the new CSS class to suit your needs.

   If you want to define more complex styles for badges, you can also create custom CSS selectors that include dynamic pseudo-classes. For example, `.badge.customName:after`.

Results

The badges that are displayed on access cards are changed to reflect the customization that you made in the CustomLabels.properties and Badge.css files.

What to do next

Select one or more accesses for a user that is based on your request access requirements.

Customizing the access card selection limit in the Request Access wizard

The second step in the Request Access wizard is used to select the accesses that are being requested for a user. There is a limit to the number of accesses that can be requested at one time. You can change this limit to meet your needs.

Before you begin

You must have read or write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details about where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

The Select accesses page of the Request Access wizard is used to select the accesses that are being requested for a user. By default, the number of accesses that can be selected on the Select accesses page is limited to 25. But you can change this limit to meet your needs by modifying the config/UIconfig.properties file.

Procedure

1. Make a custom copy of the config/UIconfig.properties file.
2. In a text editor, open the UIconfig.properties file and locate the access.selection.maximum.number property.
3. From the top level of the **Appliance Dashboard**, click **Configure** > **Advanced Configuration** > **Custom File Management**.
4. Select UIconfig.properties and then click **Download**.
5. Change the value to the expected access limit for your organization. For example, 20.
6. Click **Save Configuration**.

Results

The Select accesses page of the Request Access wizard restricts the number of accesses that the user can select to the specified limit in the config/UIconfig.properties file.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center. Log

in to the Identity Service Center. Start the Request Access wizard and confirm that the number of accesses selected is restricted to the specified limit.

Customizing the search controls in the Request Access wizard

The third step in the Request Access wizard is used to provide required information for accesses that are being requested. The forms for this required information might contain fields that are defined as `Search Control` or `Search Match Control`. You can customize the appearance of the search controls to meet your organizational requirements.

Before you begin

You must have read and write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details about where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

For some access requests, the user provides required information by completing fields of the form that is associated with the access. Some fields of the form might have a search control that enables the user to search the IBM Security Identity Manager Server for the appropriate value.

Each search field on a form is configured to search for a specific category of object, such as a `Person`, `Account`, or `Organizational Unit`. Search controls on forms can be used in two modes:

- The basic search mode allows the user to type some search text into the form field. The objects that match the search text are displayed as a drop-down list of cards under the form field.
- The Advanced Search mode is displayed as a dialog box. You can do the following actions:
  – Specify the search text.
  – Select the specific attribute to compare.
  – Select a comparison operator such as **Equals** or **Contains**.

  The objects that match the search criteria are displayed in rows of a table.

The object attributes in the drop-down list of cards and in the columns of the Advanced Search table can be customized to suit your needs. The information is arranged into several areas, and you can choose which object attribute is displayed in each area.

The `primary` area of the card displays the most important attribute, such as the object name. The information in this area is displayed at the top of the card and in the largest font. Only one attribute can be assigned to the `primary` area. But you can choose a different attribute for each of the object profiles that are defined in your environment. The attribute that is assigned to this area of the card is displayed as the first column in the Advanced Search table.

The `secondary` area of the card displays the next most important attribute, such as the object description. The information in this area is displayed just under the `primary` area and in a smaller font than the `primary` area. Only one attribute can be assigned to the `secondary` area. But you can choose a different attribute for each of

the object profiles that are defined in your environment. The attribute that is assigned to this area of the card is displayed as the second column in the Advanced Search table.

The `tertiary` area of the card displays extra information about the object, such as the user title. The information in this area is displayed just under the `secondary` area and in a smaller font than the `secondary` area. Only one attribute can be assigned to the `tertiary` area. But you can choose a different attribute for each of the object profiles that are defined in your environment. The attribute that is assigned to this area of the card is displayed as the third column in the Advanced Search table.

The `icon` area of the card displays an image that is associated with the object. The icon is displayed at the side of the card, next to the `primary`, `secondary`, and `tertiary` areas. The attribute that is assigned to this area must provide the location (URI) of the image to display. You can choose a different attribute for each of the object profiles that are defined in your environment. The attribute that is assigned to this area of the card is not displayed in the Advanced Search table.

Sometimes the attribute that you want to display is not an attribute of the object, but it might be the attribute of a related object. For example, a user might have attributes that are called "manager" or "ersponsor" that are actually references to related users, namely the manager or sponsor of this user. To display an attribute like "name" from the related user in the card or the Advanced Search table, you can specify the attribute by using dotted notation. For example, `"manager.name"` or `"ersponsor.name"`.

Note: Some types of attributes, such as mapped attributes and attributes from related objects, can be selected and displayed in search results. But they cannot be used as the search criteria.

Procedure

1. Optional: Customize the attributes that are displayed in the different areas of search cards and the Advanced Search table. Make a custom copy of the `config/Search.json` file and open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is a way of representing structured data.

   The `config/Search.json` file has sections for each object category, such as `Person` or `ACCOUNT`. You can use this file to select different display attributes for each type of object. There are sections within each object category that define the attributes to display for that object type.

   The `primary` section contains an `attribute` subsection. For example:

   ```
   "primary": {
        "attribute": {
            "default": "name",
            "Person" : "CN"
        },
        . . .
    },
   ```

   In the `attribute` section, define the object attribute to display in the `primary` area of the search card and in the first column of the Advanced Search table. You can choose a different attribute for each profile that is defined in your environment. You must always set a default attribute to use for any profiles not explicitly defined. In the preceding example, the default attribute is `"name"`, but the attribute for objects in the `"Person"` profile is `"CN"`.

   The `secondary` section is identical to the `primary` section. For example:

```
"secondary": {
    "attribute": {
        "default": "mail"
    },
    . . .
},
```

In the `attribute` section, define the object attribute to display in the `secondary` area of the search card and in the second column of the Advanced Search table. You can choose a different attribute for each profile that is defined in your environment. You must always set a default attribute to use for any profiles that are not explicitly defined. In the previous example, the default attribute is "mail", and no other attributes are defined for specific profiles.

The `tertiary` section of this file is identical to the `primary` and `secondary` sections. For example:

```
"tertiary": {
    "attribute": {
        "default": "title"
    },
    . . .
},
```

In the `attribute` section, define the object attribute to display in the `tertiary` area of the search card and in the third column of the Advanced Search table. You can choose a different attribute for each profile that is defined in your environment. You must always set a default attribute to use for any profiles not explicitly defined. In the previous example, the default attribute is "title", and no other attributes are defined for specific profiles.

2. Optional: Customize the labels that are displayed for the column headings in the Advanced Search table. Make a custom copy of the `nls/SearchCustomAttributes.properties` file and open the file with a text editor. The properties in this file define the text that is displayed in the column headings of the Advanced Search table.

   The `primary`, `secondary`, and `tertiary` sections of each object category in the `config/Search.json` file contain a **labelKey** field. For example:

```
"primary": {
    . . .
    "labelKey": "name"
},
"secondary": {
    . . .
    "labelKey": "contactInfo"
},
"tertiary": {
    . . .
    "labelKey": "title"
},
```

   The uppercase value of these **labelKey** fields is used to look up the display strings for the column headings of the Advanced Search table in the `nls/SearchCustomAttributes.properties` file. In this example, the **labelKey** value NAME is looked up as a property to find the column heading to display for the `primary` attribute. It is the first column in the Advanced Search table. If any properties are not found, then the value of the **labelKey** field is used as the column heading.

3. Customize the `icon` area of the search card to display an image for the associated object. Make a custom copy of the `config/Search.json` file and open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is a way of representing structured data.

The `icon` section of each object category of this file contains an attribute subsection. For example:

```
"icon": {
    "attribute": {
        "default": "erimageuri"
    }
}
```

In the `attribute` section, define the attribute that contains the location of the icon or image to display on the search card. You can choose a different attribute for each profile that is defined in your environment. You must always set a default attribute to use for any profiles that are not explicitly defined. If some profiles do not have an image attribute, you can specify `null` to indicate that no image must be displayed for objects in that profile. In the earlier example, the default attribute is `"erimageuri"`, and no other attributes are defined for specific profiles.

Results

The appearance of the search controls on form fields of the Request Access wizard is changed to reflect the customization that you made.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center.

Log in to the Identity Service Center. Start the Request Access wizard, and verify that the appearance of the search control reflects the customization that you made.

Customizing the hint and help text for form fields in the Request Access wizard

The third step of the Request Access wizard is used to provide required information for the accesses that are being requested. For some access requests, the user provides the required information by completing fields of the form that is associated with the access. These forms can be customized by the site administrator. For more information, see the "Form customization" section in the IBM Security Identity Manager product documentation. The hint text and help text that are associated with the fields on a form can be customized to suit your needs.

Before you begin

Depending on your system customization, you might not have access to this task. Contact your system administrator to obtain access to this task, or to have someone complete it for you.

About this task

The site administrator can set properties for form fields to define hint text and help text. A user can view the hint text and help text in the Identity Service Center to understand what values are needed or are appropriate for the fields.

- If the administrator sets the `Hint text` property for a form field, the text is displayed inside the input field as a hint to the user. The hint text is replaced by the field data that is entered by the user.

- If the administrator sets the `Help text` property for a form field, a help icon is displayed next to the label for the form field. If the user selects or hovers over the help icon, the help text is displayed to provide more information about the form field.

The hint text and help text are specified as either a fixed string such as `Select a primary group` or the name of a property in the `CustomLabels.properties` file such as `$PrimaryGroupHintText`. If a fixed string was defined as the hint text or help text, then it cannot be customized. But if a property name was defined, you can customize the hint text or help text by modifying the value of the property.

Procedure

1. Consult with the site administrator to determine the property names that were defined as the values of the `Hint text` and `Help text` for the form fields that you want to customize.
2. Open the `CustomLabels.properties` file in the `ISIM_HOME\data` directory, where `ISIM_HOME` is the IBM Security Identity Manager installation directory.
3. Optional: Find the property names that are defined for the `Hint text` in the `CustomLabels.properties` file. For each of the form fields that you want to customize, modify the values to the hint text that you want to display.
4. Optional: Find the property names that are defined for the `Help text` in the `CustomLabels.properties` file. For each of the form fields that you want to customize, modify the values to the help text that you want to display.
5. Save and close the `CustomLabels.properties` file.

Results

The fields on the Provide account information form of the Request Access wizard that have the `Hint text` property defined now display the customized hint text in the input area. The fields that have the `Help text` property defined now display a help icon next to the field label, and selecting or hovering over the help icon displays the customized help text.

What to do next

Log in to the Identity Service Center. Start the Request Access wizard, and verify that the Provide account information form fields reflect the customization that you made.

View Access wizard

You can customize the user interface of the View Access wizard to suit your needs.

You can change or customize the following items:
- The display of user ID to make the user ID sortable
- The display of the account status and compliance information on the access cards

Customizing an account attribute in the View Access wizard

You can customize an account attribute to display or sort on the access cards.

Before you begin

- You must have read or write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47

on page 47 for specific details about where these files are located. Contact your system administrator if you do not have the necessary permissions.

- By default, the user ID is displayed on the access cards and sorting is enabled for the user ID.

About this task

The file `config/Access.json` contains the access attributes that are customizable. The access attributes can be in the following categories:

- Standard attributes that are in the primary, secondary, or tertiary areas of the access cards. See "Customizing an access card in the Request Access wizard" on page 76.
- Account attributes that are customized for user-specific views. For example, `account.eruid`.

Procedure

1. Optional: Customize an account attribute that is displayed on the access cards.
   a. Make a custom copy of the `config/Access.json` file.
   b. Open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is way of representing structured data.

   The `userAccessView` section contains the `userId` subsection. For example:

   ```
   "userAccessView": {
      "userId": {
      "attribute": "account.eruid" "display": true,
       .....
       }
      }
   ```
   c. If you do not want to display an account attribute on the access cards, set `"display": false`.

2. Optional: Customize an account attribute for the sort functionality.
   a. Make a custom copy of the `config/Access.json` file.
   b. Open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is way of representing structured data.

   The `uerId` section contains the `sort` subsection. For example:

   ```
   "userAccessView": {
      "userId": {
      "attribute": "account.eruid" "display": true,
       "sort": {
        "enabled": true,
        "labelKey": "userId"
       }
      }.....
   }
   ```
   c. To disable the sort on an account attribute, set `"enabled": false`.

   Note: If you enable the sort functionality, an account attribute is the third sort attribute in the grid sortable attributes.

Results

An account attribute that is displayed on access cards reflects the customization that you made in the `config/Access.json` file.

Customizing account status and compliance information in the View Access wizard

You can customize the account status and compliance information that is displayed on the access cards in the View Access wizard.

Before you begin

- Depending on your system customization, you might not have access to this task. Contact your system administrator to obtain access to this task, or to have someone complete it for you. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details about where these files are located.

- By default, the account status and compliance information is displayed on the access cards. Account status is displayed for the suspended and disallowed accounts. The compliance information is displayed for the accounts that are non-compliant. You might see the following compliance messages on the access cards:

Compliance evaluation is pending
> Indicates that the access was returned from a reconciliation, which means it was not checked against the existing policies.

Access is not compliant with the policy
> Indicates that the access can exist for the user, but that one or more of the underlying account attributes do not comply with the existing provisioning policies.

Access is not allowed by the policy
> Indicates either that the access is not supposed to exist because the user is not allowed to have access to the specified resource, or that a provisioning policy is not defined for the resource.

No message
> Indicates that the access is compliant.

About this task

The file `config/Access.json` contains the access attributes that are customizable. The access attributes can be in the following categories:

- Standard attributes that are in the primary, secondary, tertiary, or image areas of the access cards. For more information about customizing the standard attributes, see "Customizing an access card in the Request Access wizard" on page 76.
- Attributes that are customized for the user-specific views. For example, account status and compliance information.

Procedure

1. Optional: Customize the account status that is displayed on the access cards.

   a. Make a custom copy of the `config/Access.json` file.

   b. Open the file with a text editor. The contents of this file are maintained in *JavaScript Object Notation* (JSON) format, which is way of representing structured data.

   For example:
   ```
   "userAccessView": {
     "userId": {
       "attribute": "account.eruid",
       "display": true,
   ```

```
        "sort": {
         "enabled": true,
         "labelKey": "userId"
        }
       },
       "showCompliance": true,
                "showAccountStatus": true
      }
```

   c. If you do not want to display the account status on the access cards, set the
value for showAccountStatus to false. For example, "showAccountStatus":
false.

2. Optional: Customize the compliance information that is displayed on the access
cards.

   a. Make a custom copy of the config/Access.json file.

   b. Open the file with a text editor. The contents of this file are maintained in
JavaScript Object Notation (JSON) format, which is way of representing
structured data.

For example:

```
"userAccessView": {
   "userId": {
    "attribute": "account.eruid",
    "display": true,
    "sort": {
     "enabled": true,
     "labelKey": "userId"
    }
   },
   "showCompliance": true,
            "showAccountStatus": true
  }
```

   c. If you do not want to display the compliance information on the access
cards, set the value for showCompliance to false. For example,
"showCompliance": false.

Results

The account status and compliance information that are displayed on access cards
are changed to reflect the customization that you made in the config/Access.json
file.

View pending accesses on access cards

Access cards display the information about the pending accesses that are requested
from the IBM Security Identity Manager.

You can view the pending accesses that are requested from the following IBM®
Security Identity Manager user interfaces:

- Administrative console
- Self-service user interface
- Identity Service Center

You can also view the accesses requests that are originated from public APIs or
web services.

Viewing service-based accesses

If you cannot view the service-based accesses on the View Access page in the
Identity Service Center, change the access control item (ACI) to grant the required
permissions.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. Log on to administrative console
2. From the navigation tree, select **Set System Security** > **Manage Access Control Items**.
3. On the Manage Access Control Items page, type `Default ACI for Account: Grant Search, Add, Change Password, and All groupMember operations to Self` in **Search information**.
4. Click **Search**.
5. In the **Access Control Items** table, select `Default ACI for Account: Grant Search, Add, Change Password, and All groupMember operations to Self`.
6. Click **Permissions** tab.
7. For the **Service** attribute, grant the **Read** permission.
8. Click **Apply**.

Results

You can now view service-based accesses on the View Access page in the Identity Service Center.

Manage Activities wizard

You can view and act on the activities that are assigned to you. You can also review activities that you completed.

View approval details

Before you act on the requests, you can view the details about the requests.

You can view the operation that triggers an approval request. You can configure the following entities and operations for approval activities:

Access
> Includes the roles and groups in the system. You can configure these operations for approval activities: `Add`, `Modify`, and `Delete`.
>
> **Note:** On the Approval Details page, the **View Details** link is not visible in the Identity Service Center user interface for access entity.

Account
> Includes the accounts on a service. You can configure these operations for approval activities: `Add`, `Modify`, `Delete`, `Restore`, and `Suspend`.

Person
> Includes any user in the system. You can configure these operations for approval activities: `Add`, `Modify`, `Delete`, `Restore`, `SelfRegister`, `Suspend`, and `Transfer`.

Dependent access overview:

Dependent accesses are provisioned to a user upon approval of the access request. The dependent accesses are evaluated based on the provisioning policy that is configured with the service access.

The following cases explain what can be considered as dependent accesses.

Provisioning policy membership
> The provisioning policy membership is one of the criteria that governs the calculation of the dependent accesses. For a provisioning policy with a membership type as role and the provisioning option as automatic, the entitlements that are added are considered as dependent accesses. The following target type options are available for a provisioning policy:
> - Specific service
> - Service type
> - Service selection policy

Group-based access requests
> When a group membership is requested and if there is no account for a user on the service, a new account is created. This new account is considered as dependent access.

Customizing the due date notification period

You can customize the default due date notification period (24 hours) to suit your business needs.

Before you begin

You must have read or write access to the customizable files and their directories. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47. Contact your system administrator if you do not have the necessary permissions.

About this task

Procedure

1. Edit the `config/UIconfig.properties` file with a text editor.
2. Search for the `activity.duedate.threshold` property.
3. Set the value for the property `activity.duedate.threshold` in hours. For example, `activity.duedate.threshold=48`.

Results

The due date notification period changes.

Customizing the labels

You can rename the labels to suit your business needs.

Before you begin

You must have read or write access to the customizable files and their directories. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47. Contact your system administrator if you do not have the necessary permissions.

About this task

AA and AR are the default approval activity result codes that are used by workflows in IBM Security Identity Manager. The user interface displays labels from the `CustomLabels.properties` file that are based on the property names. These property names are derived from the approval activity result codes. If you customize workflows to use customized approval activity result codes, you must add the appropriate properties to the `CustomLabels.properties` file. You customize the labels for the interface when you define these properties. For example, if you use the customized result code `XYZ` to represent approval, you must add properties with the names `XYZ`, `XYZ_inProgress`, and `XYZ_complete` to `CustomLabels.properties` file.

Note: If you do not add the values to the `CustomLabels.properties` file, the actual key names are displayed in the user interface. For example, `XYZ_inProgress` is displayed instead of "Processing".

Procedure

1. Optional: Rename any of the following labels in the `CustomLabels.properties` file by changing the property value to the text that you want to display:

Label	Property value
Approve	AA
Reject	AR
Approving	AA_inProgress
Rejecting	AR_inProgress

   a. Open the `CustomLabels.properties` file in the *ISIM_HOME*/data directory, where *ISIM_HOME* is the IBM Security Identity Manager installation directory.
   b. Change the value of the property to the text that you want to display for the label.
2. Optional: Rename the either `Approved` or `Rejected` labels or both.
   a. Make a custom copy of the `nls/ActivityListCard_en.properties`.
   b. In a text editor, open the custom copy of `ActivityListCard_en.properties` file.
   c. Change the value of the property `AA_Complete` or `AR_Complete` to the text that you want to display for the label.

Results

The labels change to reflect the customization that you made in the `CustomLabel.properties` and `ActivityListCard_en.properties` files.

Customize an account activity card to display application attributes

You can customize activity cards to view application attributes on it.

The `config/Access.json` file contains details about the primary, secondary, and tertiary attributes that are displayed on an activity card. You can add attributes as a tertiary attribute that can be displayed on an account activity card.

Attributes assigned to the `tertiary` area of the activity card are displayed with a label to help the user understand what information is displayed. For example, to display the service attribute in the activity card, add the service attribute to the `tertiary` area.

```
"tertiary": ["additionalInformation", "serviceName" ],
```

Redirecting help content

You can redirect help requests to your own website to deliver custom help content.

Before you begin

You must have read and write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details of where these files are located. Contact your system administrator if you do not have the necessary permissions.

About this task

Editing the help content that is shipped with the Identity Service Center user interface is not supported. But you can redirect help requests to your own website to deliver custom help content in line with your corporate appearance.

The `config/UIHelp.properties` file specifies properties that control the redirection of help file requests. The following table shows the property and property description for Identity Service Center help.

Table 22. Identity Service Center help properties and description

Property	Description
helpBaseUrl	Specifies the base URL to which to send help requests. A blank or null value indicates that help goes to the default URL for the help files in the Identity Service Center, `/itim/uihelp`.
helpLocales	Restricts the locales for which help is supported. For example, **helpLocales=en,fr** restricts help support to English and French regardless of the number of available locales. If the attribute is not specified or null, the supported help locales are the same as the supported locales for the Identity Service Center user interface. These locales are specified by the **isim.ui.supportedLocales** property in the `config/UIconfig.properties` file.
Help Id mappings: helpId = relativeHelppageURL	The help mappings section maps IDs from specific pages to a relative URL sent to the help server.

The Help URL is the combination of the helpBaseUrl + locale + relativeHelppageURL

For example, if your custom `config/UIHelp.properties` file contains:

```
helpBaseUrl=http://myserver:80/helpfiles
login_help_url=ui/ref_ui_login.html
```

Then for a user who selects the English (en) locale, the request for the Login help page is redirected to `http://myserver:80/helpfiles/en/ui/ref_ui_login.html`.

Procedure

1. Create a custom copy of the `config/UIHelp.properties` file.
2. Change the **helpBaseUrl** property in the file.
3. Update **helpId** mappings to use the relative URLs for your server.
4. Optional: If you want to restrict the list of supported help locales, uncomment the **helpLocales** property. Modify it to specify the list of locales for which help is supported in your environment.
5. Add pages to your server for the appropriate locales.

Results

The help requests are now redirected to your customized help files.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center. Start the Identity Service Center in a browser to verify that the customized file is being used.

Supporting more languages

You can extend the Identity Service Center to support more languages by providing your own translations for all of the IBM-provided globalization files.

Before you begin

You must have read and write access to the customizable files and the directories where they are maintained. See "Location of Identity Service Center customizable files" on page 43 and "Customizing Identity Service Center files" on page 47 for specific details of where these files are located. Contact your system administrator if you do not have the necessary permissions.

Determine the value of the locale identifier for the language (xx) or language and country (xx_YY) associated with the translation you want to provide. For example: da (Danish) or ro_MO (Romanian, Moldova).

About this task

Procedure

1. Create a custom version of the `common.properties` file for the new language.

   For example, create the `nls/common_xx_YY.properties` file.

   You must create the properties file in the new locale. The server determines the list of available locales by searching for all variants of the `common.properties` file. You must also translate all of the properties in the `nls/common_xx_YY.properties` file to the new language.
2. If you previously restricted the locales that you support, you must modify your customized copy of the `config/UIconfig.properties` file to include the new

language. Update the **isim.ui.supportedLocales** property in your custom version of this file to include the new locale, xx_YY. If all locales are supported, no change is required to the `config/UIconfig.properties` file because the default is to support all available locales.

3. The new language might be a language that is read from right-to-left. In this case, you must modify your customized copy of the `config/UIconfig.properties` file to include the new language in the list of right-to-left locales. Update the **isim.ui.rtlLocales** property in your custom version of this file to include the new locale, xx_YY.

4. Create custom versions of all supported language variants of the `custom/ui/nls/UILanguages*.properties` files.

   Add a line to each file that specifies the display name for the new locale. This file is used to build the language selection control of the Login page. For example: xx_YY=*New language name*

5. Create custom versions of all of the other files under `custom/ui/nls` for the new locale.

6. Translate the text in all of the `nls/*_xx_YY.properties` files into the new language.

7. You can add a language for page help files only if you already provided custom help files, as described in "Redirecting help content" on page 93. Take these steps:

   a. Create a directory for the xx_YY locale at the same level as the directory that contains the existing `en` (English) and other locales.

   b. Copy the custom help files for an existing language into the new xx_YY directory with the same directory structure.

   c. Translate the help files in the xx_YY directory to the new language.

   d. Update the <html> element in each of the help files to specify the locale of the new language, such as <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="**xx-yy**" lang="**xx-yy**">. If the new language is read from right-to-left, you must also modify the <html> element in each of the help files to specify the direction as "rtl". For example: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="xx-yy" lang="xx-yy" dir="**rtl**">

8. If you previously restricted the help locales that you support, you must modify your customized copy of the `config/UIHelp.properties` file to include the new language. Update the **helpLocales** property in your custom version of this file to include the new locale, xx_YY. If all help locales are supported, no change is required to the `config/UIHelp.properties` file because the default is to support all available locales.

Results

The Identity Service Center user interface is available to users in the new language translation. If you translated the help files, the Identity Service Center page help is also available to users in the new language translation.

What to do next

You can verify the change immediately in a single-server WebSphere Application Server environment. In a managed-cluster environment, you must wait until after the configuration is synchronized to the application servers in the cluster. For information about synchronizing the WebSphere managed-cluster environments, see the WebSphere Application Server section of the IBM Knowledge Center.

To view the new language translation, select the new language from the dropdown list of languages on the Identity Service Center Login page. Alternatively, configure your browser preferences to use the new language.

Chapter 2. Service type management

A *service type* is a category of related services that share the same schemas. It defines the schema attributes that are common across a set of similar managed resources.

Overview

Service types are profiles, or templates, that are used to create services for specific instances of managed resources. For example, if you have several Lotus® Domino® servers that users need access to, you might create one service for each Lotus Domino server using the Lotus Domino service type. In previous versions of IBM Security Identity Manager, a service type is referred to as a *service profile*.

Some service types are installed by default when IBM Security Identity Manager is installed. Other service types can be installed when you import the service definition files for adapters for managed resources. A service type definition is provided by the Security Identity Manager Adapter for a managed resource. There is a service type for each type of managed resource that Security Identity Manager supports, such as UNIX, Linux, Windows, IBM Security Access Manager, and so on.

A service type is defined in the service definition file of an adapter, which is a Java Archive (JAR) file that contains the profile. The service type for an adapter is created when the adapter profile (JAR file) is imported. For example, a service type is defined in the `WinLocalProfile`JAR file. You can also define a service type using the interface for Security Identity Manager.

Security Identity Manager supports the following types of service providers:
- DAML for Windows Local Adapter, Lotus Notes® Adapter, and so on
- IDI (IBM Security Directory Integrator for UNIX and Linux adapters)
- Custom Java class for defining your own implementation of a service provider
- Manual for managing user-defined "manual" activities

Default service types

The following default service types are provided with Security Identity Manager:

Identity feed service types:

> **DSML**
> > A Directory Services Markup Language (DSML) Identity Feed service imports user data, with no account data, from a human resources database or file and feeds the information into the Security Identity Manager directory. The service uses a placement rule to determine where in the organization a user will be placed. The service can receive the information in one of two ways: a reconciliation or an event notification. This service is based on the DSML Identity Feed Service Profile.

Note: DSMLv2 is deprecated in Security Identity Manager Version 5.0 in favor of the remote method invocation (RMI)-based IDI adapter framework. The use of DSMLv2 continues to be supported in this release.

AD The AD Identity Feed Service imports user data from Windows Active Directory. The organizationalPerson objects are fed into Security Identity Manager and add or update users to Security Identity Manager. The user profiles selected from this service must have an objectclass that is derived from the organizationalPerson class.

CSV The CSV Identity Feed Service imports user data from a comma-separated value (CSV) file and adds or updates users to Security Identity Manager. The CSV file contains a set of records separated by a carriage return/line feed (CR/LF) pair (\r\n). Each record contains a set of fields separated by a comma. If the field contains either a comma or a CR/LF, the comma must be escaped with double quotes as the delimiter. The first record in the CSV source file defines the attributes provided in each of the following records. Attributes must be valid based on the class schema for the selected person profile for this service.

IDI Data Feed
The IDI Data Feed service type uses the Security Directory Integrator to import user data, with no account data, into Security Identity Manager and to manage accounts in the Security Identity Manager data store on external resources. This service is based on the IDI Data Feed Service Profile.

INetOrgPerson
The INetOrgPerson Identity Feed imports user data from the LDAP directory. The inetOrgPerson objects are loaded and add or update users in Security Identity Manager.

Account service types:

Security Directory Integrator-based
This service type can be optionally installed during the installation of Security Identity Manager. All these are Security Directory Integrator-based adapters; each is a specific service type. Security Directory Integrator is one type of service provider. There can be multiple service types defined for the same type of service provider.

ITIM Service
The ITIM service type is used to create accounts in the Security Identity Manager system and represents the IBM Security Identity Manager Server itself. This is a standard service with no configuration parameters. All users that need access to the Security Identity Manager system must be provisioned with a Security Identity Manager account.

Hosted Service
The Hosted Service type is used to create a service that is a proxy to the hosting service that is residing in the service provider organization.

The hosted service connects to the managed resource target through the hosting service indirectly. The configuration details of

the hosting service is invisible and protected from administrators in the secondary organization where the Hosted Service is defined. Administrators can define policies for the hosted service, specifically, without affecting the hosting service.

The primary usage of a Hosted Service is to allow users in business partner organizations to have accounts and access to internal IT resources of an organization and to allow administrators in the secondary organization to define specific service policies for the user accounts.

Custom Java class
The custom Java class service type allow you to define your own profile by defining and implementing a Java class.

Manual services and service types

The manual service type manages user accounts on a target resource manually. Account requests are routed to a specific user rather than a service provider so that it can be handled manually or by using other tools outside of IBM Security Identity Manager Server.

These are resources for which at least one of the following statements apply:
- There is no adapter currently available to do the provisioning, and it is not possible or practical to develop a custom adapter.
- Some or all of the provisioning activity requires a person to do the necessary setup process.
- You choose to do the task manually.

Examples of resources for manual service types and manual services include:
- Voice mail setup
- Telephone setup
- Personal computer setup
- Physical mail setup
- Employee badge request

Creating manual services

Create a manual service instance when IBM Security Identity Manager does not provide an adapter for the managed resource.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Before you can create a manual service in Security Identity Manager, you must create a service type. Add new schema classes and attributes for the manual service to your LDAP directory.

Tip: You can also specify labels for new schema attributes in the `CustomLabels.properties` file.

About this task

A manual service is a type of service that requires manual intervention to complete the request. For example, a manual service might be defined for setting up voice mail for a user. A manual service generates a work order activity that defines the manual intervention that is required.

If you choose to create a provisioning policy as part of this task, the service is automatically added to the provisioning policy as an entitlement. In addition, a membership of "All" is defined for the provisioning policy. Also, an ownership type of "Individual" is defined for the provisioning policy. You can later edit the provisioning policy and change the membership and ownership types after the service is created.

The service name and description that you provide for each service are displayed on the console. Therefore, it is important to provide values that make sense to your users and administrators.

To create a manual service instance, complete these steps:

Procedure

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, click **Create**. The Create a Service wizard is displayed.
3. On the Select the Type of Service page, click **Search** to locate a business unit. The Business Unit page is displayed.
4. On the Business Unit page, complete these steps:
   a. Type information about the business unit in the **Search information** field.
   b. Select a business type from the **Search by** list, and then click **Search**. A list of business units that match the search criteria is displayed.

   If the table contains multiple pages, you can:
   - Click the arrow to go to the next page.
   - Type the number of the page that you want to view and click **Go**.
   c. In the **Business Units** table, select business unit in which you want to create the service, and then click **OK**. The Select the Type of Service page is displayed, and the business unit that you specified is displayed in the **Business unit** field.
5. On the Select the Type of Service page, select a manual service type, and then click **Next**.

   If the table contains multiple pages, you can:
   - Click the arrow to go to the next page.
   - Type the number of the page that you want to view and click **Go**.
6. On the General Information page, specify the appropriate values for the manual service instance, and then click **Next**. The content of the General Information page depends on the type of service that you are creating. The creation of some services might require additional steps.
7. On the Participants page, specify the users who are involved in completing the activities for the manual service. Specify the amount of time before the service is escalated. Click **Next**.
8. Optional: On the Messages page, complete these steps, and then click **Reconciliation**:

a. Select the default email message that you want to change, and then click **Change**. The Change Message page is displayed.

b. Modify the **Subject** and **Body** fields, and then click **OK**.

9. Optional: On the Access Information page, select the **Define an Access** check box to activate the access definition fields. Select the type of access you want to enable. Specify the expected access information and any other optional information such as description, search terms, more information, or badges.

10. On the Configure Policy page, select a provisioning policy option, and then click **Next** or **Finish**. The provisioning policy determines the ownership types available for accounts. The default provisioning policy enables only Individual ownership type accounts. Additional ownership types can be added by creating entitlements on the provisioning policy.

11. Optional: On the Reconciliation page, click **Browse** to locate the reconciliation file, and then click **Upload File** to load the new reconciliation file. You can also choose whether to reconcile supporting data only.

   Note: The file type that is supported for the reconciliation file is CSV. For more information, see the topic "Example comma-separated value (CSV) file" in the *IBM Security Identity Manager Administration Guide*.

12. Click **Finish**.

Results

A message is displayed, indicating that you successfully created the manual service instance for a specific service type.

What to do next

Select another services task, or click **Close**. When the Select a Service page is displayed, click **Refresh** to refresh the **Services** table and display the new service instance.

Changing a manual service

Change information for a manual service instance.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Before you can change a service in IBM Security Identity Manager, you must create a service instance.

Procedure

To change a manual service instance, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.

2. On the Select a Service page, complete these steps:

   a. Type information about the service in the **Search information** field.

   b. In the **Search by** field, specify whether the search must be done against services or business units.

c. Select a service type from the **Search type** list.

    d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

       If the table contains multiple pages, you can:

       - Click the arrow to go to the next page.

       - Type the number of the page that you want to view and click **Go**.

3. In the **Services** table, select the check box next to the manual service that you want to change, and then click **Change**.

4. On the General Information page, change the appropriate values for the service instance, and then click **Participants**.

5. On the Participants page, change the participants type, escalation time in days, or escalation participant type.

6. Optional: On the Messages page, complete these steps, and then click **Reconciliation**:

   a. Select the email message that you want to change, and then click **Change**. The Change Message page is displayed.

   b. Modify the **Subject** and **Body** fields as wanted, and then click **OK**.

7. Optional: On the Reconciliation page, click **Browse** to locate the reconciliation file, and then click **Upload File** to load the new reconciliation file. You can also choose whether to reconcile supporting data only.

   Note: The file type supported for the reconciliation file is CSV. For more information, see the topic "Example comma-separated value (CSV) file" in the *IBM Security Identity Manager Planning Guide*.

8. Click **OK** to save the changes and to close the page.

Results

A message is displayed, indicating that you successfully changed the service instance.

What to do next

Select another services task, or click **Close**. When the Select a Service page is displayed, click **Refresh** to refresh the **Services** table.

Configuring a manual service type to support groups

To support group assignment, but not group management for manual services, the group profile needs to be set up in the manual service type configuration.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To set up a manual service type to support group assignment, but not group management (which includes create, read, update, delete) for manual services, complete these steps:

Procedure

1. Define the group schema as an LDAP objectclass in the IBM Security Identity Manager LDAP server.

2. Define a manual service (complete with service and account objectclasses). The account objectclass should contain an optional multi-valued attribute that will be used to store the group membership information. This service type should reference the group schema created in the previous step.

   The Manage Service Types page allows the administrator to select an existing LDAP objectclass for use as the group schema class. If you want to create a new objectclass, you must create it manually and load it directly into the LDAP server.

   The mapped **Group ID**, **Group name**, and **Group description** attributes can all reference the same group schema attribute, if desired. You cannot define multiple groups that use the same group ID. The ID must be unique per group.

   More than one group schema can be defined for a given service type. The definition of the second and subsequent schemas is performed in the same manner as the first.

3. Modify service and account forms for the service type using the form designer. This step is required to properly display needed information when creating the service instance as well as creating accounts.

4. Create a manual service instance using the manual service type that you created earlier in this process.

Reconciling accounts for manual services

Initiate a reconciliation activity on a manual service.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

You must have completed the steps for configuring a manual service type to support groups. You must also have created a manual service instance before you begin this task.

About this task

The service instance creation steps allow you to perform a reconciliation of a manual service using a comma-separated value (CSV) file that you provide. The reconciliation populates IBM Security Identity Manager with accounts and groups that exist on the manual service. The CSV file contains group and account information.

You can provide the reconciliation file at service creation time or at any time the service is modified. There is also a *supporting data only* option for reconciliation that is used when you want to pull group information from the CSV file, but you do not want to touch accounts in Security Identity Manager.

To perform a reconciliation on a manual service, complete these steps:

Procedure

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether the search should be performed against services or business units.
   c. Select a service type from the **Search type** list, and then click **Search**. A list of services matching the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▸) adjacent to the service to show the tasks that can be performed on the service, and then click **Change**. The tasks that you can perform are dependent on the type of service. The Select Query page is displayed.
4. On the Reconciliation page, click **Browse** to locate the reconciliation file, and then click **Upload File** to load the new reconciliation file. You can also choose whether or not to reconcile only supporting data.
5. Click **OK** to save the changes and to close the page.

Results

A message is displayed, indicating that you successfully submitted a reconciliation request.

What to do next

To view the results of the reconciliation, click **View the status of the reconciliation request**. You can also select another services task, or click **Close**. When the Select a Service page is displayed, click **Refresh** to refresh the **Services** table.

Service definition file or adapter profile

A *service definition file*, which is also known as an *adapter profile*, defines the type of managed resource that IBM Security Identity Manager can manage.

The service definition file creates the service types on the IBM Security Identity Manager Server.

The service definition file is a Java archive (JAR) file that contains the following information:
- Service information, including definitions of the account provisioning operations that can be performed for the service, such as add, delete, suspend, or restore.
- Service provider information, which defines the underlying implementation of how the IBM Security Identity Manager Server communicates with the managed resource.
- Schema information, including the LDAP classes and attributes.
- Account forms and service forms, along with the label for the attributes, which are displayed in the user interface for creating services and requesting accounts on those services.

Creating service types

As an administrator, you can create a service type. For example, you might create a service type for a manual service that you want to create.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Defining a new service type allows you to define new LDAP attributes and objectclasses. You can also change the existing LDAP attributes and objectclasses. You must understand the impact of changing the LDAP schema through this task. Do not change the syntax or schema of existing attributes and objectclasses. If a new service type is needed, define one.

See your directory documentation for restrictions and best practices to use for schema extension. For IBM Security Directory Server Version 6.3.1.5, see http://www.ibm.com/support/knowledgecenter/SSVJJU_6.3.1.5/com.ibm.IBMDS.doc_6.3.1.5/c_ig_schema_management.html.

About this task

You can create a service type for a manual service or for a custom service.

To create a service type, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Manage Service Types**. The Manage Service Types page is displayed.
2. On the Manage Service Types page, click **Create**. The Manage Service Types notebook is displayed.
3. On the General page of the Manage Service Types notebook, complete these steps:
   a. In the **Service Type Name** field, provide a unique name for your service type.
   b. From the **Service Provider** list, select the protocol that IBM Security Identity Manager uses to provision accounts for the service type.
   c. Click the **Service** tab.
4. On the Service page, specify an LDAP class and attributes to associate with the service type, and then click the **Account** tab. The LDAP class and attributes vary, depending on the accounts that the managed resource provides.
5. On the Account page, specify an LDAP class and attributes to associate with the account schema, and then click either the **Group** tab or **OK**.
6. Optional: On the Group page, complete these steps:
   a. To add a group to the service type, click **Add**. The Add Group page is displayed.
   b. On the Add Group page, specify an LDAP class and schema information. A group schema must be supported by the adapter for this service type.
   c. Click either the **Miscellaneous** tab, or click **OK**.
7. Optional: On the Miscellaneous page, complete these steps:

a. Select the check box if you want the service type to participate in reports for dormant accounts.

b. From the **Last access date** list, select an attribute of the account schema that is associated with the service type, and then click **OK**.

Results

A message indicates that you successfully created a service type.

What to do next

Verify the generated service and account forms for the new service type with the form designer, set up account defaults for the service type, or click **Close**.

Tip: You can also specify values for **Service Type Name** and **Description** fields in the `CustomLabels.properties` file.

Changing service types

You can change a service type to select a different service provider. You can also change a service type to change the LDAP class or attributes for the service type or the accounts.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

A service type must exist, but no instance of the service type can exist.

Defining a new service type allows you to define new LDAP attributes and objectclasses. You can also change the existing LDAP attributes and objectclasses. You must understand the impact of changing the LDAP schema through this task. Do not change the syntax or schema of existing attributes and objectclasses. If a new service type is needed, define one. See your directory documentation for restrictions and best practices to use for schema extension. For IBM Security Directory Server Version 6.3.1, see *Managing the IBM Directory schema* http://www.ibm.com/support/knowledgecenter/SSVJJU_6.3.1/ com.ibm.IBMDS.doc_6.3.1/admin_gd56.htm%23wq73.

About this task

You cannot change a service type if there is a service instance of the service type. Users might actively be working in accounts on that service instance.

To change a service type, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Manage Service Types**. The Manage Service Types page is displayed.

2. Manage Service Types page, select the check box next to the service type that you want to change, and then click **Change**. The Manage Service Types notebook is displayed.

3. On the Manage Service Types notebook, make the wanted changes, and then click **OK**. The name of the service type cannot be changed.

Results

A message indicates that you successfully modified the service type.

What to do next

If necessary, use the form designer to update the service and account forms to match any service type attribute changes, or click **Close**.

Importing service types

As an administrator, you can import a service definition file, which creates a service type. Service definition files are also called adapter profile files, which are provided with the various IBM Security Identity Manager adapters.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

The file to be imported must be a Java archive (JAR) file.

About this task

You can create a service type for an adapter that provides a JAR file.

To import a service type, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Manage Service Types**. The Manage Service Types page is displayed.
2. On the Manage Service Types page, click **Import**. The Import Service Type page is displayed.
3. On the Import Service Type page, complete these steps:
   a. In the **Service Definition File** field, type the directory location of the file, or click **Browse** to locate the file. For example, if you are installing the Security Identity Manager adapter for a Windows server that runs Active Directory, locate and import the `ADProfileJAR` file.
   b. Click **OK** to import the file.

Results

A message indicates that you successfully submitted a request to import a service type.

What to do next

The import occurs asynchronously, which means it might take some time to complete. On the Manage Service Types page, click **Refresh** to see the status of the new service type. If the service type status is `Failed`, check the log files to determine why the import failed.

Deleting service types

You can delete a service type that has no service instances. For example, if your enterprise replaces an application, you might migrate user records to the new application. Then, delete the obsolete service type.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Before you delete a service type, you must remove all of its service instances.

About this task

When you delete a service type, changes made to the LDAP class persist even after the service type is deleted.

To delete a service type, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Manage Service Types**. The Manage Service Types page is displayed.
2. Manage Service Types page, select the check box next to the service type that you want to change, and then click **Delete**. Selecting the check box at the top of this column selects all service types. The Manage Service Types notebook is displayed.
3. On the Confirm page, click **Delete** to delete the service type, or click **Cancel**.

Results

A message indicates that you successfully deleted the service type.

What to do next

Do other service type management tasks, or click **Close**.

Management of account defaults on a service type

You can define default values for account attributes either on a service or on a service type.

Types of account defaults

Service type account defaults
> When account defaults are defined at the *service- type level*, they apply to all services of that type. However, a service type default can be overridden by defining an account default at the *service level*.
>
> You can define global account default values in one place, a service type. You do not need to define the same account default values for a service in multiple places. This single definition reduces the amount of customization and the chance of omissions or errors.

Service account defaults
> These defaults are initially inherited from the service type account defaults,

but they become local to the service as soon as it is being changed. They become local account defaults and can be changed or removed. Changes (including removals) do not affect the service type account defaults.

Options for defining default values for account attributes

Basic Allows you to hard code default values. You can also build a rule to extract information from an attribute on any IBM Security Identity Manager person class object. You can use it to set the value for an account attribute.

Advanced
Allows you to code JavaScript to retrieve LDAP data from IBM Security Identity Manager objects and set the value for an account attribute. As a starting point, you can create a basic account default and then use the advanced option to edit the generated JavaScript.

Adding account defaults to a service type

Add account defaults to a service type.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

The wanted the service type must exist. If it does not exist, you must import the profile for the service type.

About this task

You can add default values for attributes. When you create a service instance from this service type, the account defaults for the service type are copied to the service.

To add account defaults to a service type, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Manage Service Types**. The Manage Service Types page is displayed.
2. In the **Service Types** table, click the icon (▶) next to the service type, and then click **Account Defaults**. The Select an Account Attribute page is displayed.
3. On the Select an Account Attribute page, click **Add** to add an attribute. The Select an Attribute to Default page is displayed.
4. On the Select an Attribute to Default page, select an account attribute. Click one of these choices:
   - **Add**, which adds a default value for the selected attribute. Complete the appropriate fields, which vary depending on the type of service, and then click **OK**. The attribute default is added to the list on the Select an Attribute to Default page.
   - **Add (Advanced)**, which adds a script that specifies a default value for the selected attribute. Type the wanted JavaScript code in the **Script** field, and then click **OK**. The attribute default is added to the list on the Select an Attribute to Default page.
5. On the Select an Account Attribute page, finish adding attribute defaults to the service type. Then, click **OK** to save the changes and to close the page.

Results

A message indicates that you successfully saved the account defaults on the service type.

Changing account defaults for a service type

Change the account defaults for a service type.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To change account defaults for a service type, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Manage Service Types**. The Manage Service Types page is displayed.
2. In the **Service Types** table, click the icon (▸) next to the service type, and then click **Account Defaults**. The Select an Account Attribute page is displayed.
3. On the Select an Account Attribute page, select the check box next to the attribute that you want to modify, and then click one of these choices:
   - **Change**, which changes the default value for the selected attribute. Complete the appropriate fields, which vary depending on the service type, and then click **OK**. The template value for the attribute is updated in the list on the Select an Attribute to Default page.

      Note: If you select this option when an attribute currently has a scripted default value, the existing script is overwritten with the template value that you specify.
   - **Change (Advanced)**, which adds or changes the script that specifies a default value for the selected attribute. Type the wanted JavaScript code in the **Script** field, and then click **OK**. The template value for the attribute is updated in the list on the Select an Attribute to Default page.
4. On the Select an Account Attribute page, finish changing attribute defaults for the service type. Then, click **OK** to save the changes and to close the page.

Results

A message indicates that you successfully saved the account defaults on the service type.

Removing account defaults from a service type

Remove account defaults from a service type.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To remove account defaults from a service type, complete these steps:

Procedure
1. From the navigation tree, select **Configure System** > **Manage Service Types**. The Manage Service Types page is displayed.
2. In the **Service Types** table, click the icon (▶) next to the service type, and then click **Account Defaults**. The Select an Account Attribute page is displayed.
3. On the Select an Account Attribute page, select the check box next to the attribute that you want to remove, and then click **Remove**. Selecting the check box at the top of this column selects all attributes. The attribute default is removed from the list on the Select an Attribute to Default page.
4. On the Select an Account Attribute page, finish removing attributes from the service type. Then, click **OK** to save the changes and to close the page.

Results

A message indicates that you successfully removed the account defaults from the service type.

Chapter 3. Access type management

Access types are a way to classify the kinds of access that users see. Use the **Manage Access Types** task to classify the types of accesses in your organization.

The following access types are included with: IBM Security Identity Manager
- Application, which is access to an application
- E-mail group, which is membership in an email group
- Role, which is a role for IT resource access
- Shared folder, which is access to a shared folder

As an administrator, you can create more access types, such as for intranet web applications or Active Directory (AD) application shared folders.

Over time, several accesses might be defined. Classify them into commonly available accesses, or use categories for smarter searches for infrequent accesses.

Creating access types

As an administrator, you can create additional access types, such as for intranet web applications or Active Directory (AD) application shared folders.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Over time, several accesses might be defined. Classify them into commonly available accesses or use categories for searches for infrequent accesses.

Note: The name of an access type cannot contain a back slash (\).

To create an access type in the tree structure, complete these steps:

Procedure
1. From the navigation tree, click **Configure System** > **Manage Access Types** to display the Manage Access Types page. The Manage Access Types page lists the default access types.
2. On the Manage Access Types page, click the icon next to the **Access Types** node.
3. Click **Create Type** to display the Create Access Type page.
4. On the Create Access Type page, complete the following steps:
   a. In the **Access Type Key** field, provide a key name. For example, `Payroll`.
   b. In the **Description** field, provide a description about the access type.
5. Click **OK** to save the access type. A message indicates that you successfully created an access type. The Manage Access Types page displays the new access type in the tree structure.

6. Create additional access types, or click **Close**.

What to do next

- You can add property key and value pairs in the `CustomLabels.properties` resource bundle to provide the display label for this access type.

  See the CustomLabels.properties topic in the *IBM Security Identity Manager Reference Guide*.
- Users can request access to the new access type.

Changing access types

As an administrator, you can change access types, such as for intranet web applications or Active Directory (AD) application shared folders.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you create at least one access type in the tree structure. See "Creating access types" on page 113.

About this task

Nodes that you can select depends on the position or hyperlink of the node that you select within the tree structure.

To change an access type in the tree structure, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Access Types** to display the Manage Access Types page. The Manage Access Types page lists the default access types.
2. On the Manage Access Types page, click the icon next to the **Access Types** node and click **Change**. Alternatively, click an access type. The Change Access Type page is displayed.
3. On the Change Access Type page, modify the description in the **Description** field. You can provide a description associated to the access type key.

   Note: The **Access Type Key** field value is read-only.
4. Click **OK** to save the access type.

Results

A message indicates that you successfully changed an access type. The Manage Access Types page displays the modified access type in the tree structure.

What to do next

Users can request access to the new access type.

Change additional access types, or click **Close**.

Deleting access types

As an administrator, you can delete access types that are no longer needed in your organization.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

You must remove all access definitions for an access type before you can delete an access type.

About this task

You cannot delete an access type if any access definitions for that access type exist.

To delete an access type in the tree structure, complete these steps:

Procedure
1. From the navigation tree, click **Configure System** > **Manage Access Types** to display the Manage Access Types page. The Manage Access Types page lists the default access types.
2. On the Manage Access Types page, click the icon next to the **Access Types** node that you want to delete. Then, click **Delete** to display the Confirm page. You cannot delete an access type node that has child items or a group or role association. You must first delete the child items or the group or role association before deleting the access type.
3. On the Confirm page, click **Delete** to delete the access type, or click **Cancel**.

Results

A message indicates that you successfully deleted an access type. The Manage Access Types page no longer displays the deleted access type in the tree structure.

What to do next

Create or change access types, delete additional access types, or click **Close**.

Chapter 4. Global adoption policies

An *adoption policy* is used during reconciliation to determine the owner of an account. A *global adoption policy* is defined for a service type or all service types, for the entire system. Global adoption policies are applicable to all service instances if no adoption policy is defined for the specific service.

The default global adoption policy assigns an account to a user if the account user ID attribute matches the IBM Security Identity Manager user UID attribute. A service-specific adoption policy takes precedence over the global adoption policy.

Creating a global adoption policy

You can add a customized rule for generating passwords with the IBM Security Identity Manager Server.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To create a global adoption policy, complete these steps:

Procedure
1. From the navigation tree, select **Configure System** > **Global Adoption Policies**.
2. On the Global Adoption Policies page, in the **Adoption Policies** table, click **Create**.
3. On the Global Adoption Policies page, on the General page, type a name for your adoption policy. You can add a description also.
4. Click the Service Type tab, and select a specific service type to associate with the policy. You must specify at least one service type for the global adoption policy. You cannot associate more than one global adoption policy with a service type.
5. Click the Rule tab, and specify a custom rule to govern the attributes that the adoption policy uses to match accounts to users. If you choose to define matches, click **Add a match field** to select the account and user attributes that must match during reconciliation. The user attribute drop-down list provides a few commonly used attribute combinations that can be used when defining the match. For example, a combination is the first letter of the given name plus the family name or the given name plus the first letter of the family name. If your adoption rule is more complex, you can choose the more advanced path by selecting **Providing a Script**. If you defined matches, the associated scripts are populated for you in the script definition field.

   Important: If you choose to provide a script, the Security Identity Manager Server does not verify that the JavaScript is correct. Verify the JavaScript before using it to define the policy.
6. Click **OK** to save the changes.

7. On the Success page, click **Close**. The new global adoption policy is displayed on the Global Adoption Policies page. This Global Adoption policy can be changed and deleted.

Changing a global adoption policy

An administrator can change a global adoption policy that is defined for a service type.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

The effect of changes to an adoption policy can be seen when the next reconciliation is run. Changing an existing adoption policy does not affect existing accounts of the specific service or service type. Changes do not affect accounts that are already adopted. Only new and existing orphan accounts are adopted based on the new policy.

To change a global adoption policy, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Global Adoption Policies**.
2. In the **Global Adoption Policies** table, locate and select an adoption policy that you want to change, and then click **Change**.
3. On the Global Adoption Policies page, modify the information on the General, Service, or Rule pages
4. Click **OK** to save the changes.
5. On the Success page, click **Close**.

Deleting a global adoption policy

An administrator can delete a global adoption policy that is defined for a service type.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Deleting an existing adoption policy does not affect existing accounts of the specific service type.

To delete a global adoption policy, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Global Adoption Policies**.

2. In the **Global Adoption Policies**table, locate and select an adoption policy that you want to delete, and click **Delete**.
3. On the Confirmation page, review the adoption policy to be deleted, and click **Delete**.
4. On the Success page, click **Close**.

Chapter 5. Post office configuration

The post office provides a mechanism for reducing the number of email notifications a user receives regarding similar tasks in IBM Security Identity Manager.

Overview

You can configure the post office to collect similar notifications over an interval of time. The configuration combines those multiple emails into one notification that is then sent to a user. In the workflow designer, you use the **Group E-mail Topic** field in each manual activity definition to determine similar tasks to group email notifications.

Assume that the post office is enabled. If the manual activities that generate notifications have the **Use Group E-mail Topic** option enabled, the post office intercepts email notifications that the system generates for those manual activities. The post office holds the notifications for a specified interval. When that interval expires, the post office uses the aggregate email template to aggregate all notifications that have the same **Group E-mail Topic** value into one email message for each email recipient. The preferred locate of the recipient, which is specified in the Person object, is honored. This process reduces the volume of individual email messages for notifications of the same **Group E-mail Topic** value that a user receives.

The post office uses the **Group E-mail Topic** value on the **Notification** tab of the manual activity configuration page, to determine which messages to aggregate together. All notifications that are generated with the same **Group E-mail Topic** value are aggregated together for the collection interval specified. This field can be any string, but the default is the Activity ID. This field accepts JavaScript and dynamic content tags, if it results in the execution of a string.

Assume that the collection interval expires and notifications are aggregated. If there is only one notification for a specified **Group E-mail Topic** value and email address, that message is sent in its original form. The post office email template is not applied. Although the notification is sent in its original form, the notification is delayed until the post office collection interval expires.

There might be errors while attempting to aggregate the individual emails. The messages are sent in their original form and an error message is written to the log. The process means that notifications might be delayed in getting sent, but not result in the loss of any notifications. The **Test** button on the Post Office page is useful for troubleshooting template errors.

Example email notification

The default template generates an email notification similar to this message:

```
Subject: You have 3 work items requiring your attention.

Body:
You have 3 work items requiring your attention.

Here are the email subjects:
This is subject 1
```

```
This is subject 2
This is subject 3

Here are the email message bodies:
This is the text body 1
This is the text body 2
This is the text body 3
```

The template can consist of any valid dynamic content tag and JavaScript code. In addition, the post office has a set of custom dynamic content tags and JavaScript extensions.

Customizing the post office email template

You can enable or disable the post office and set the time interval that the post office uses to collect messages to aggregate. You can also customize the email template that is used to generate the aggregate message that is sent to the recipients.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

When you enable the post office, all email notifications are stored until the time interval that you specify. At that time, the notifications are aggregated into one email message that is sent to the recipients.

The post office email template can use dynamic content. Dynamic content includes dynamic content message tags and JavaScript code. Dynamic content also includes tags that replace variables with other values, or reference a property that allows translation with the use of a `CustomLabels.properties` file.

The template is applied to the collection of notification messages that the system holds for a specified **Group E-mail Topic** value and message recipient. This template can be as simple or as complex as required. The **Group E-mail Topic** value is set in the workflow designer.

To enable the post office and configure a post office aggregation email template, complete these steps:

Procedure
1. From the navigation tree, click **Configure System** > **Post Office**. The Post Office page is displayed.
2. On the Post Office page, select the **Enable store forwarding** check box.
3. In the **Collection interval** field, type the number of minutes that you want to pass before the post office aggregates the stored email messages and sends them to the recipients. The value of the collection interval must be an integer 5 - 10080.
4. In the **Subject** field, type the text to specify the subject of the email notification that is sent as the aggregate message instead of being sent as an individual email message. The subject can consist of plain text and dynamic content tags.

5. In the **Plaintext body** field, type the text to be displayed in the body of the aggregate message. The content can consist of plain text, dynamic content tags, and JavaScript code. These contents are shown to email recipients that do not see HTML email notifications.

6. In the **XHTML body** field, type the text to be displayed in the body of the email notification as HTML. The content can consist of plain text, dynamic content tags, and JavaScript code. These contents are shown to email recipients that see HTML email notifications. For the correct aggregation of XHTML bodies of individual email templates with the post office email aggregation template, use an optional attribute 'escapeentities'. This attribute is in the <JS> tag of Post Office XHTML body template. Set the value to `false`. See Sample post office email aggregation template for more details.

7. Click **OK** to save the changes and then click **Close**.

Results

After the next interval expires, the combined notifications are aggregated and sent as one email notification.

What to do next

Test the post office email aggregation template that you created before with it to aggregate email notifications that are sent to activity participants.

Post office dynamic content custom tags

The post office defines a set of custom tags to simplify the creation of the aggregate message template. The aggregate message template is a user interface template for defining how multiple email notifications are displayed in a single email notification for a user.

The following post office dynamic content custom tags can be used to get data:

<POGetAllBodies/>
Returns a string that contains the text body of each of the original notifications that are separated by a newline. For example:

```
You have the following ToDo items in Identity Manager.
Here are the notification bodies <POGetAllBodies/>
```

<POGetAllSubjects/>
Returns all subjects from the notifications that are associated with the aggregate email notification as a string that is separated by a newline. For example:

```
You have the following ToDo items in Identity Manager.
Here are the notification subjects. <POGetAllSubjects/>
```

<POGetEmailAddress/>
Returns the email address that is the destination for the aggregate email notification as a string with no newline. For example:

```
This collection of notifications was sent to <POGetEmailAddress/>.
```

<POGetNumOfEmails/>
Returns the number of emails that are associated with the aggregate email notifications as a string with no newline. For example:

```
You have <POGetNumOfEmails/> ToDo items in Identity Manager.
```

Post office label and messages properties

Both post office labels and notification messages can be customized by editing their properties.

Custom labels for interface elements

The labels for post office configuration GUI elements can be customized by editing the following properties that are contained in the `Labels.properties` file:

- POST_OFFICE_CONFIG=Post Office Configuration
- POST_OFFICE_PROPERTIES_CUE=Modify Post Office Properties
- POST_OFFICE_PATH=Post Office
- GENERAL_TAB=General
- AGGREGATE_MESSAGE_TAB=Aggregate Message
- ENABLE_STORE_FORWARDING_LABEL=Enable Store Forwarding
- COLLECTION_INTERVAL_LABEL=Collection Interval
- SUBJECT=Subject
- TEXT_BODY=Text Body
- HTML_BODY=XHTML Body
- POST_OFFICE_DONE_ALT=Save post office properties
- POST_OFFICE_CANCEL_ALT=Cancel changes

Custom properties for notification messages

The following properties can be customized for post office notification messages. These properties are the message keys for the dynamic content tags (<RE>) that are included in the default post office template configuration.

- postoffice_subject=You have {0} work items that require your attention.
- postoffice_subject_list=Here are the email subjects:
- postoffice_body_list=Here are the email message bodies:

Post office template extensions

Review usage examples of dynamic content and JavaScript code that can be entered on the Post Office page.

Subject

```
Identity Manager: You have <POGetNumOfEmails/> work items requiring your attention.
```

Plaintext body

```
You have <POGetNumOfEmails/> work items requiring your attention.
The emails are all addressed to: <POGetEmailAddress/>
Here are the email Subjects:
<POGetAllSubjects/>
Here are the email bodies:
<POGetAllBodies/>
Here is the topic fetched using the JavaScript extension:
<JS>
    return PostOffice.getTopic();
</JS>
Here is the recipient's email address fetched using the JavaScript extension:
<JS>
    return PostOffice.getEmailAddress();
</JS>
Here are the email text bodies fetched using the JavaScript extension:
<JS>
        var msgListIterator = PostOffice.getAllEmailMessages().iterator();
```

```
                var returnString = "\n";
                while (msgListIterator.hasNext()) {
                            returnString = returnString + msgListIterator.next().getMessage() + "\n"; }
                return returnString;
</JS>
Here is the recipient's surname taken from the Person fetched using the JavaScript extension:
<JS>
        var person = PostOffice.getPersonByEmailAddress(PostOffice.getEmailAddress());
        return "Last: " + person.getProperty("sn")[0] + "\n";
</JS>
```

XHTML body

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>You have <POGetNumOfEmails/> work items requiring your attention.</title>
</head>
<body>
<POGetNumOfEmails/> notifications have been collected by Identity Manager Post Office
and aggregated below. These indicate you can have up to <POGetNumOfEmails/> work items
requiring your attention.<br />
The notifications were all addressed to: <POGetEmailAddress/><br />
<hr />
Here are the notification Subjects:<br />
<POGetAllSubjects/><br />
<hr />
Here are the notification bodies: <br />
<POGetAllBodies/><br />
<hr />
    Here is the topic fetched using the JavaScript extension:
    <JS>
return PostOffice.getTopic();
    </JS>
    <br />
    Here is the email address fetched using the JavaScript extension:
    <JS>
return PostOffice.getEmailAddress();
    </JS>
    <br />
    Here are the email text bodies fetched using the JavaScript extension:
    <JS>
var msgListIterator = PostOffice.getAllEmailMessages().iterator();
var returnString = "<br />";
while (msgListIterator.hasNext()) {
     returnString = returnString + msgListIterator.next().getMessage() + "<br />"; }
return returnString;
    </JS>
    <br />
    Here is the recipient's surname taken from the Person fetched using the JavaScript extension:
    <JS>
var person = PostOffice.getPersonByEmailAddress(PostOffice.getEmailAddress());
return "<br />Last: " + person.getProperty("sn")[0] + "<br />";
    </JS>
<hr />
Please take care of these right away.  Have a nice day !<br />
    IT Dept
  </body>
</html>
```

Post office JavaScript extensions

Use the Mail Application Programming Interface (API) to customize mail content, format, and notification recipients.

With this API, you can make notification requests and extend construction of notification messages. The Mail API contains the Mail Client API, which makes notification requests, and the Mail Provider API, which implements notification requests.

The Mail API also contains a post office function that prevents workflow participants from receiving multiple email notifications that have similar content. Similar email messages are stored, combined into a single email notification, and forwarded to a user.

Testing and troubleshooting the post office email template

Test and validate the post office email aggregation template that you created before sending it to an activity participant.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

A post office email aggregation template must already be configured.

About this task

To test the email aggregation template, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Post Office**.
2. Click **Test**. The Test email page is displayed.
3. On the Test email page, specify an email address to receive the test message, and then click **Test**. The email aggregation template is validated, and if successful, a sample email notification is sent to the email address you specified. The email message contains simulated system information, which is supplied by default in the properties file. The message is presented in the post office email template that you created.
4. Click **OK** to save the changes, and then click **Close**.

What to do next

If an error message is displayed, correct the content of the field that is indicated in the error, and then click **Test** again.

The error message describes the problem and includes an approximate line and column number where the error occurred in the message. The value returned is meant to serve as a general pointer to where the problem exists, but it is not an exact location. You cannot include the XHTML body content of the original notifications directly in your aggregation template XHTML body. By default, the post office has no XHTML body aggregation template.

View the sample email notification that you sent to the email address you specified. If necessary, you can make additional changes to the template and test it again.

Modifying the sample email content

You can modify the content of the sample email notifications that are used for testing.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

A post office email aggregation template must already be configured.

About this task

Modify the content of the sample email notification.

Procedure

1. Edit the `enRole.properties` file.
2. Specify the new `enrole.postoffice` values, and then save the `enRole.properties` file. `enRole.properties` is the name of the properties file, and `enrole.postoffice` is the name of the key for which you specify a value. This key-value pair resides in the properties file.
3. Restart your application server for the new values to take effect.

Results

The results of this task can be seen only after you test the aggregation template that you created or modified. The new sample email notifications are aggregated and sent to the test email address.

Example

The `enRole.properties` file contains the following default values:

```
###########################################################
## Post Office Template Test Configuration
###########################################################
# These are the contents of the emails that will be used
# when the "test" button is used on the Post Office
# configuration page. These 3 emails will be used as the
# content to which the template will be applied.
enrole.postoffice.test.subject1=This is subject 1
enrole.postoffice.test.textbody1=This is the text body 1
enrole.postoffice.test.xhtmlbody1=This is the xhtml body 1

enrole.postoffice.test.subject2=This is subject 2
enrole.postoffice.test.textbody2=This is the text body 2
enrole.postoffice.test.xhtmlbody2=This is the xhtml body 2

enrole.postoffice.test.subject3=This is subject 3
enrole.postoffice.test.textbody3=This is the text body 3
enrole.postoffice.test.xhtmlbody3=This is the xhtml body 3

# The topic to use for the test emails above
enrole.postoffice.test.topic=topic1

# The locale to use for the test emails above
enrole.postoffice.test.locale=en_US
```

What to do next

Test the new aggregate template by sending it to a test email address.

Enabling the post office for workflow activities

Use the workflow designer to enable the post office notifications for workflow activities.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

A workflow activity must exist.

About this task

All email notifications that have the same Group Email Topic are aggregated together with the template and sent to each recipient.

To enable the post office for a workflow activity, complete these steps:

Procedure

1. From the workflow designer, double-click an existing activity to access its Properties page.
2. From the Properties page, click the **Notification** tab.
3. Select the **Use Group Email Topic** check box.
4. In the **Group Email Topic** field, type a value to use to aggregate similar messages.
5. Click **OK** to save the workflow activity, then click **OK** to save and exit the workflow designer.

Results

The workflow activity is saved. The next time this workflow is triggered, this change is in effect.

Chapter 6. Form customization

You can create and modify forms for the attributes on the IBM Security Identity Manager interface.

Only individuals who are part of the administrator group can access this feature.

IBM Security Identity Manager provides default forms to create, view, and modify system entities. The form designer allows system administrators to manage all entity forms from one location.

System administrators can customize forms for the following system entities with the form designer:

- Account
- Admin Domain
- Business Partner Organization
- Business Partner Persons
- Credential Lease
- Identity Manager User
- Location
- Organization
- Organizational Unit
- Person
- Role
- Service

Each form category folder has object profiles that represent system entities. Each object profile is associated with a form template.

Default form templates are generated from the configuration of an entity. Form templates have at least one tab and one form element. A tab is a container for grouping form elements. A form element is a system entity attribute. Each tab consists of a label that describes the group and at least one form element. Each form element consists of a label that describes its data and the data input format. Form elements are listed in the order the elements are presented on the form.

Customizing form templates

You can use the form designer applet to open form templates, which display required form elements, form element organization, and form element control type.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To open a form template, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.

Results

The form template associated with the object profile is displayed in the middle pane.

What to do next

You can select a form element and right-click to do various actions. Mouse over the icons on the top of the form to get hints about the function of the icon.

Adding tabs to form templates

Use these instructions to add tabs to form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To add a tab to a form template, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.

4. Click **Tab** > **Add Tab**. A new tab is displayed in the form template.
5. To name the new tab, click **Tab** > **Rename Tab**.
6. Type a name for the new tab in the entry field, and then click **OK**. The name of the new tab is displayed in the form template.
7. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

Renaming tabs on form templates

Use these instructions to rename tabs on form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To rename a tab on a form template, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.
4. Click **Tab** > **Rename Tab**.
5. Type a new name for the tab in the entry field, and then click **OK**. The new name of the tab is displayed in the form template.
6. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

Arranging tabs on form templates

Use these instructions to arrange tabs on form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To move a tab to a different position on a form template, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.
4. In the middle pane, select the tab that you want to move.
5. Select one of the following options:
   - Click **Tab** > **Shift Tab Left** to move the tab one position to the left.
   - Click **Tab** > **Shift Tab Right** to move the tab one position to the right.
6. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

Deleting tabs from form templates

Use these instructions to delete tabs from form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

If a tab contains required attributes, you cannot delete the tab.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To delete a tab from a form template, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.
4. In the middle pane, select the tab that you want to delete.
5. Click **Tab** > **Delete Tab**. The tab is removed from the form template.

6. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

Adding attributes to form templates

Use these instructions to add attributes to form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To add an attribute to a form template, complete these steps:

Procedure
1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Then double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.
4. Select the tab to which you want to add the attribute.
5. In the Attribute List pane, double-click the attribute name that you want to add to the form. The attribute is added to the form.
6. Click **Form** > **Save Form Template**, and then click **OK** when a message indicates that the form template is saved successfully.

What to do next

Continue adding attributes as needed.

Modifying attribute properties

The form element properties section consists of two tabs, **Format** and **Constraint**. The **Format** tab lists all the formatting properties that might or might not be applicable, depending on the input control type defined for the element. Similarly, the **Constraint** tab lists all available constraints that might or might not be applicable to the input control type defined. If a property or constraint is not applicable, you cannot select or set a value.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

It is possible to combine custom constraints on a single field in a way that makes input impossible. The form designer applet checks for constraint conflicts so that invalid combinations do not occur on a single field.

As a rule, use only one syntax constraint per field, and use only one data type constraint per field.

For example, if the Minimum Value exceeds the Maximum Value, and if both constraints are placed on the same field, then a conflict exists. If a conflict exists, you must change the values or remove one of the constraints.

To modify the properties of an attribute, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.
4. In the middle pane, select the attribute for which you want to modify properties. The properties of the attribute are displayed in the **Properties** pane.
5. In the **Format** tab, change the property to the wanted value. The new property value is displayed, and the changes are reflected in the attribute.
6. In the **Constraint** tab, select the check box next to the constraint that you want to modify.
7. Enter parameters for any value constraint types.
8. Type a sample value in the field at the bottom of the list of constraint types.
9. Click the **Validate and Update Constraints** button.

   The form designer applet notifies you if a conflict between constraints exists. Alternatively, a **Pass** message is displayed if the value entered is valid according to the constraints used, and if none of the constraints conflict with each other.
10. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

Changing attribute control types

Control types define the interface for users to input data for that form element. Currently supported control types are CheckBox, Date, DropDown Box, Editable

Text List, ListBox, LoginHours, Password, Password Popup, Search Control, Search Match, SubForm, TextField, TextArea, and Umask.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To change the control type of an attribute, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.
4. In the middle pane, select the attribute for which you want to change the control type.
5. Click **Attribute** > **Change To**. A list of control types is displayed.
6. Select the wanted control type. For some control types, an editor is displayed.
7. If a control type editor is displayed, enter the wanted parameters, and click **OK**.
8. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

Arranging attributes on form templates

Use these instructions to arrange attributes on form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To move an attribute to a different position on a form template, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.
4. In the middle pane, select the attribute that you want to move.
5. Select one of the following options:
   - Click **Attribute** > **Move Up Attribute** to move the attribute up one position.
   - Click **Attribute** > **Move Down Attribute** to move the attribute down one position.
6. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

Deleting attributes from form templates

Use these instructions to delete attributes from form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To delete an attribute from a form template, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the left pane, double-click the wanted category folder to display the object profiles for the entity type. Double-click the wanted object profile to open the template for that profile. The form template associated with the object profile is displayed in the middle pane.
4. In the middle pane, select the attribute that you want to delete.
5. Click **Attribute** > **Delete Attribute**. The attribute is removed from the form template.
6. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

Customizing account form templates for a service instance

You can open a customized account form directly from the service instance.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

You can customize the account form for each service instance. When the form designer applet is launched for customizing the account form at the service instance level, the navigation tree panel is not shown. This session is only for customizing the account form for the specific service instance. For instructions on how to customize the form, see Customizing form templates sections.

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Note: The custom form for the actual ITIM service is not supported because there is only one ITIM service instance. This account form can be configured at the system level. However, the custom account form is supported for the hosted ITIM service instance because there can be one or more hosted ITIM service instances.

Procedure

To open a form template, complete these steps:
1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the service to show the tasks that can be done on the service, and then click **Customize Account Form**. The form designer applet is started.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started.

Results

The customized account form associated with the service instance is displayed. If there is no customized account form for the service instance then the form template is displayed.

What to do next

You can select a form element and right-click to do various actions. Mouse over the icons on the top of the form to get hints about the function.

Adding tabs to form templates for a service instance

Use these instructions to add tabs to form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Procedure

To add a tab to a form template, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▸) next to the service to show the tasks that can be done on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. Click **Tab** > **Add Tab**. A new tab is displayed in the form template.
6. To name the new tab, click **Tab** > **Rename Tab**.
7. Type a name for the new tab in the entry field, and then click **OK**. The name of the new tab is displayed in the form template.

8. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue adding tabs as needed.

Renaming tabs on form templates for a service instance

Use these instructions to rename tabs on form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Procedure

To rename a tab to a form template, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

   If the table contains multiple pages, you can:
   - Click the arrow to go to the next page.
   - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the service to show the tasks that can be done on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. Click **Tab** > **Rename Tab**.
6. Type a new name for the tab in the entry field, and then click **OK**. The new name of the tab is displayed in the form template.
7. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue renaming tabs as needed.

Arranging tabs on form templates for a service instance

Use these instructions to arrange tabs on form templates for a service instance.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Procedure

To arrange a tab to a form template, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.
      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the service to show the tasks that can be done on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. Select the tab that you want to move.
6. Select one of the following options:
   - Click **Tab** > **Shift Tab Left** to move the tab one position to the left.
   - Click **Tab** > **Shift Tab Right** to move the tab one position to the right.
7. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue arranging tabs as needed.

Deleting tabs from form templates for a service instance

Use these instructions to delete tabs from form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Procedure

To delete a tab to a form template, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▸) next to the service to show the tasks that can be performed on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. Select the tab that you want to delete.
6. Click **Tab** > **Delete Tab**. The tab is removed from the form template.
7. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue deleting tabs as needed.

Adding attributes to form templates for a service instance

Use these instructions to add attributes to form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Procedure

To add an attribute to a form template, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the service to show the tasks that can be done on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. Select the tab to which you want to add the attribute.
6. In the Attribute List pane, double-click the attribute name that you want to add to the form. The attribute is added to the form.
7. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue adding attributes as needed.

Modifying attribute properties

The form element properties section consists of two tabs, **Format** and **Constraint**. The **Format** tab lists all the formatting properties that might or might not be applicable, depending on the input control type defined for the element. Similarly, the **Constraint** tab lists all available constraints that might or might not be applicable to the input control type defined. If a property or constraint is not applicable, you cannot select or set a value.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Procedure

To modify the properties of an attribute, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the service to show the tasks that can be done on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. Select the attribute for which you want to modify properties. The properties of the attribute are displayed in the **Properties** pane.
6. In the **Format** tab, change the property to the wanted value. The new property value is displayed, and the changes are reflected in the attribute.
7. In the **Constraint** tab, select the check box next to the constraint that you want to modify.
8. Enter parameters for any value constraint types.
9. Type a sample value in the field at the bottom of the list of constraint types.
10. Click the **Validate and Update Constraints** button.

    The form designer applet notifies you if a conflict between constraints exists. Alternatively, a **Pass** message is displayed if the value entered is valid according to the constraints used, and if none of the constraints conflict with each other.
11. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue modifying attributes as needed.

Changing attribute control types

Control types define the interface for users to enter data for that form element. Currently supported control types are CheckBox, Date, DropDown Box, Editable Text List, ListBox, LoginHours, Password, Password Popup, Search Control, Search Match, SubForm, TextArea, TextField, and Umask.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template. Close the browser and reopen it before beginning a new procedure if you encounter browser or system performance issues.

Procedure

To change the control type of an attribute, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▸) next to the service to show the tasks that can be performed on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. Select the attribute for which you want to change the control type.
6. Click **Attribute** > **Change To**. A list of control types is displayed.
7. Select the control type. For some control types, an editor is displayed.
8. If a control type editor is displayed, enter the parameters, and click **OK**.

9. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue changing the control types of attributes as needed.

Arranging attributes on form templates for a service instance

Use these instructions to arrange attributes on form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Procedure

To move an attribute to a different position on a form template, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.
      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the service to show the tasks that can be done on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. Select the attribute that you want to move.
6. Select one of the following options:
   - Click **Attribute** > **Move Up Attribute** to move the attribute up one position.
   - Click **Attribute** > **Move Down Attribute** to move the attribute down one position.

7. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue to arrange attributes as needed.

Deleting attributes from form templates for a service instance

Use these instructions to delete attributes from form templates.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

Procedure

To delete an attribute to a form template, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the service to show the tasks that can be done on the service. Click **Customize Account Form**.
4. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is started. The form template associated with the service instance is displayed.
5. In the middle pane, select the attribute that you want to delete.
6. Click **Attribute** > **Delete Attribute**. The attribute is removed from the form template.
7. Click **Form** > **Save Form Template**, and then click **OK** when a message is displayed, indicating that the form template is saved successfully.

What to do next

Continue deleting attributes as needed.

Removing a customized form template from a service instance

You can remove a customized account form template from a service instance and restore the system account form.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

Procedure

To remove a customized account form template, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:

      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the service to show the tasks that can be done on the service. Click **Delete Custom Account Form**. A confirmation page is displayed.
4.
   - Click **Delete** to remove the customized form from the service instance.
   - Click **Cancel** to return to the Select Service page without removing the customized form.

   A message is displayed to indicate whether the account form was successfully deleted.
5. Click **Close** to return to the Select Service page.

What to do next

Perform additional service actions.

Customizing an account request

You can change an account request to include a group description in the administrative console user interface.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To view a group description during an account request, use the form designer to manually change the account form. The **Description** column is added only if the service type supports the description. Supported service types are Active Directory, Windows Local, and LDAP.

To change the account form, complete these steps:

Procedure

1. Click **Configure System** > **Design Forms**.
2. Under **Account**, select an account type, such as **Windows Local Account** to customize the input form.

Table 23. Service types and group attributes

Service Type	Group Attribute
Active Directory	eradprimarygroup
Windows Local Account	erntlocalname
LDAP	erladapgroupname

3. Double-click on the group attribute to display the **SearchFilter Editor** dialog. On the **SearchFilter Editor** dialog:
   a. Specify a value for the **Object Class** field.

Table 24. Service types and object classes

Service Type	Object Class
Active Directory	eradgroup
Windows Local Account	erwinlocallocalgroup
LDAP	erLdapGroupAccount

   b. Ensure that the **Description Attribute** field contains a value that is appropriate for the service type, as listed in the following table. If there is no entry in this field, a subsequent group search page contains no **Description** information in the search results table.

Table 25. Service types and description attributes

Service Type	Description Attribute
Active Directory	eradgroupdescription
Windows Local Account	erntgroupcomment
LDAP	erLdapGroupDescription

   c. Optional: Click **Show Query UI** and click **OK**.
   d. Click the **Save Form Template** icon to save the change.
4. Verify the change.

a. Go to **Manage Users** > **Select a User** > **Request Accounts** > **Select a Service**.

b. On the **Search Control** properties, select the **Service Name** with the Service Type for which the account form was changed.

c. Click **Continue**. On the account form, click **Search** next to the group attribute and verify that the search result includes a **Description** column.

Resetting form templates

Before saving changes to the form template, you can reset the form template to its original configuration.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Only individuals who are part of the administrator group can access this feature.

About this task

The form designer Java applet does not automatically close and clear from memory after starting. Complete and save any changes to a form template.

To reset the form template to its original configuration, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Design Forms**.
2. Choose either to open a file with the Java Web Start Launcher or save the file. Click **OK**. If you chose to save the file, locate the file and run it in the Java Web Start Launcher. The form designer applet is displayed.
3. In the form designer applet, click **Form** > **Reset Form Template**.
4. Click **Yes** when you are prompted that the changes to the form template will be lost.

Form designer interface

Use the work areas in the form designer applet to design custom forms by doing actions on form templates, tabs, and attributes.

The form designer interface has these work areas:

Menu and toolbar buttons
Use the menu bar and toolbar buttons to do actions on form templates, tabs, and attributes. Place the mouse cursor over a toolbar button to view its function. The following menu items and toolbar buttons are available:

Table 26. Form designer applet menu and toolbar buttons

Menu bar	Menu item	Toolbar button	Action
Click **Form** to open, save, or reset a form template to the last saved design.	**Open Form Template**		Opens the form template that is selected from the form category folders.
	Save Form Template		Saves the form template that is currently open.
	Reset Form Template	None	Resets the form template to the last saved design.
Click **Tab** to add, rename, delete, or shift a tab left or right in the interface. Tabs are displayed in the Template Attributes work area of the form designer applet. The tab names in the form designer correspond to the tab names in the resulting notebook forms in the IBM Security Identity Manager interface.	**Add Tab**		Adds a container for grouping form elements.
	Rename Tab	None	Renames an existing tab container.
	Shift Tab Left		Shifts an existing tab container to the left.
	Shift Tab Right		Shifts an existing tab container to the right.
	Delete Tab		Deletes an existing tab from the form template.
Click **Attribute** to edit, delete, move an attribute up or down in the interface, or change the control type of an attribute. Attributes are displayed in the Template Attributes work area of the form designer applet.	**Edit Attribute**	None	Edit and configure an attribute.
	Delete Attribute		Removes an attribute from a form template.
	Move Up Attribute		Repositions the attribute up 1 space in the attribute list of the form template.
	Move Down Attribute		Repositions the attribute down 1 space in the attribute list of the form template.
	Change To	None	Change the selected attribute control type to a newly selected control type.

Table 26. Form designer applet menu and toolbar buttons (continued)

Menu bar	Menu item	Toolbar button	Action
Click **View** to select various interface viewing options, such as floating work areas or viewing the source of the form template.	**Float Attribute List**		Moves the attribute list from the form designer to a floating pop-up window.
	Float Property		Moves the property list from the form designer to a floating pop-up window.
	View Source		Opens a pop-up window that displays the XML source for the form template.
Click **Menu.theme** to select an interface theme for the form designer applet.	**Default Theme**	None	Applies the default menu theme to the form designer interface.
	High Contrast, Big Font Theme	None	Applies a large font and high contrast colors to the form designer interface.
	High Contrast Theme	None	Applies high contrast colors to the form designer interface.

Categories
> Use the left pane of the form designer to select a category, such as Account, Organization, or Service. Each form category is associated with object profiles that represent system entities. Each object profile is associated with a form template.

> Double-click a category folder to expand the list of available form templates for that category. Loading the list of form templates might take some time. The list of form templates for some categories varies, depending on which service types exist.

> Double-click a form template to open it.

Template Attributes
> Use the middle pane of the form designer to view and change the active attributes for a selected form template. Right-click the attribute to display the available actions for that attribute.

> For example, a Service form template has a $servicename attribute. To change the control type that is associated with an attribute, right-click the attribute and click **Change to** on the list.

Attribute List
> Use this list to view all of the attributes for the selected object that are not currently included on the form. You can sort the list in ascending or descending order, and you can add attributes from this list to the list of active template attributes. For example, an Organization object has additional attributes, such as $postalcode, that you might add to the list of active template attributes.

Properties
> Contains **Format** and **Constraint** tabs, which specify data type and other

parameters for a specific attribute. For example, the data type for a
$servicename attribute is Directory String, and it is a required attribute.

Control types used by the form designer

Use control types in the form designer applet to specify how users enter a value
for an attribute.

CheckBox

Assigns a single check box as the data gathering field. This control type is
typically used for attributes that are Boolean in nature.

Date

Provides a calendar pop-up window that allows users to select the desired
date. This control type has additional attributes that can be used to
configure the date.

When you select this control type in the form designer applet, the Date
Editor page is displayed. You can use the fields in the editor to configure
the control type. The Date Editor contains the following fields:

DateInput Type
Select the type of date input for the calendar pop-up window.

Default
Provides a calendar pop-up window and a **Never** check
box. If the user selects the check box, then the attribute
value never expires.

Alternative Date
Provides a calendar pop-up window without a **Never**
check box. Use this type if the attribute value must expire
at some point in time.

Show Time
Select this check box to include a pop-up window that you can use
to view and specify a time.

DropDown Box

Creates a list for an attribute. You must populate the attributes to be
contained in the list by with one of the following options:

Custom Values
Limits the information that is available in the list on the resulting
form. When you select this option, the Select Editor page is
displayed. You can use the fields in the editor to configure the
control type. The Select Editor contains the following fields and
toolbar buttons:

Number of Rows
Type the number of rows to include in the list and press
Enter. Use this field to specify the number of rows in the

list. If the original list contains more rows than the number that you enter, then the extra rows are removed.

Data Value
> Type a data value.

Display Value
> Type a display value to display in the list.

Use Blank Row
> Select this check box to insert a blank entry into the list.

Add Row
> Click to add a row to display in the list.

Delete Row
> Click to delete a row from the list.

Use Display Value as Data Value
> Click to use the same value that is entered in the **Display Value** column for the **Data Value** column.

Use Index as Data Value
> Click to use the same value that is in the index for the **Data Value** column.

Search Filter
> Provides a broader range from which to gather information when populating the box. Use an LDAP search filter that assigns a value to an attribute through the use of a search control. When you select this option, the SearchFilter Editor page is displayed. You can use the fields in the editor to configure the control type. The SearchFilter Editor contains the following fields:
>
> **Search Base**
>> Select the scope of the search from these options:
>>
>>> **org** searches the organization of the selected container in the organization tree.
>>>
>>> **contextual** searches the selected organizational unit in the organization tree.
>
> **Object Class**
>> Type the name of the LDAP class to search for, such as erNTGlobalGroup. The value for the group field on the resulting form must be erroles.
>
> **Attribute**
>> Type the attribute to search for, such as erNTLocalName.
>
> **Source Attribute**
>> Type the attribute value to return after the search completes, such as erNTGlobalGroupId.
>
> **Description Attribute**
>> Type the attribute value that is appropriate for the service type. If there is no entry in this field, the group search page in the user interface contains no **Description** column in the search results table. For more information, see "Customizing an account request" on page 147.
>
> **Filter** Type any additional filter that needs to be applied to the

search, such as (objectclass=erNTLocalGroup). The value for the group field on the resulting form must be objectclass=erroles.

Delimiter
Type the delimiter to use to separate attribute values in the resulting form.

Source Attribute Delimiter
Type the delimiter to use to separate multiple source attribute values of an entity to store in the directory server. Make sure that the specified delimiter is not part of any source attribute value. If configured properly, you can use the search filter to resolve the attribute value correctly when you must store multiple source values of an entity in a delimiter-separated string format.

Multiple Value
Select this check box to change a dropdown box to a list box in the resulting form. The list box allows users to select more than one value.

Show Query UI
Select this check box to display a search page in the resulting form. When this option is not selected, only search results are displayed in a separate page.

Paginate Results
Select this check box to display the search results across multiple pages.

Editable Text List

Enables the display of multi-value attributes on the user interface. This control type is a list box that displays user-provided information. Users can enter information into the text field and add it to the list box by clicking **Add**, and they can delete information from the list box by selecting the entry and clicking **Delete**.

ListBox

Provides a list box for an attribute. The list box contains user-selected data. Users can add one or more items to a list box, and they can delete one or more items from the list box.

Custom Values
Limits the information that is available in the list on the resulting form. When you select this option, the Select Editor page is displayed. You can use the fields in the editor to configure the control type. The Select Editor contains the following fields and toolbar buttons:

Number of Rows
Type the number of rows to include in the list and press **Enter**. Use this field to specify the number of rows in the list. If the original list contains more rows than the number that you enter, then the extra rows are removed.

Data Value
> Type a data value.

Display Value
> Type a display value to display in the list.

Use Blank Row
> Select this check box to insert a blank entry into the list.

Add Row
> Click to add a row to display in the list.

Delete Row
> Click to delete a row from the list.

Use Display Value as Data Value
> Use the same value that is entered in the **Display Value** column for the **Data Value** column.

Use Index as Data Value
> Use the same value that is in the index for the **Data Value** column.

Search Filter
> Provides a broader range from which to gather information when populating the box. Use an LDAP search filter to assign a value to an attribute through the use of a search control. When you select this option, the SearchFilter Editor page is displayed. You can use the fields in the editor to configure the control type. The SearchFilter Editor contains the following fields:

Search Base
> Select the scope of the search from these options:
>
>> **org** searches the organization of the selected container in the organization tree.
>>
>> **contextual** searches the selected organizational unit in the organization tree.

Object Class
> Type the name of the LDAP class to search for, such as erNTGlobalGroup. The value for the group field on the resulting form must be erroles.

Attribute
> Type the attribute to search for, such as erNTLocalName.

Source Attribute
> Type the attribute value to return after the search completes, such as erNTGlobalGroupId.

Filter Type any additional filter that needs to be applied to the search, such as (objectclass=erNTLocalGroup). The value for the group field on the resulting form must be objectclass=erroles.

Delimiter
> Type the delimiter to use to separate attribute values in the resulting form.

Multiple Value

Select this check box to change a dropdown box to a list box in the resulting form. The list box allows users to select more than one value.

Show Query UI

Select this check box to display a search page in the resulting form. When this option is not selected, only search results are displayed in a separate page.

Paginate Results

Select this check box to display the search results across multiple pages.

LoginHours

Defines the hours that a service is available for users to log in to it. Use this control type only on forms for services that support restricted login times, such as a Windows 2000 service.

When you select this control type in the form designer applet, the LoginHours Editor page is displayed. You can use the fields in the editor to configure the control type to default to a specific type of search. The LoginHours Editor contains the following fields:

Time Interval

Select the time interval to be displayed in the resulting form:

One Hour sets the time interval to one-hour blocks.

Mid Hour sets the time interval to half-hour blocks.

Orientation

Select the orientation for the editor that is used to define login times on the resulting form:

Portrait places the days of the week along the X-axis and the time (in half-hour or one-hour blocks) along the Y-axis.

Landscape places the time (in half-hour or one-hour blocks) along the X-axis and the days of the week along the Y-axis.

Password

Provides a text box for an attribute that does not display the information that a user provides. The information is masked on the screen for security.

Password Popup

Opens a window for the user to enter secure information. The information is masked on the screen and provides two text fields to enter the information. This control type is typically used for the shared secret of an individual.

Search Control

Provides a text field search page for the selected attribute, and includes **Search** and **Clear** buttons. Users populate the text field by selecting the wanted search result. In the resulting form in the user interface, the **Search** button opens a search page with the search type already selected, and the **Clear** button clears the text field.

When you select this control type in the form designer applet, the Search Control Editor page is displayed. You can use the fields in the editor to configure the control type to default to a specific type of search. The Search Control Editor contains the following fields:

Category
Select the category for the search.

Profile
Select the profile to use for the search.

Attribute
Select the attribute to use for the search.

Operator
Select the operator, such as **Contains** or **Equals**, that links the **Attribute** and Value **fields** together.

Value Type the value for the attribute.

Type Select the type of attributes to be returned. A single-value type provides a text field for the user to populate. A multi-value type provides a list box of attributes. In this scenario, users can identify which attributes to search by selecting the attributes that they do not want to include in the search and clicking the **Delete** button. Deletion removes the selected attributes from the list of searchable attributes.

Search entire organization (current container only if not checked)
Select this check box if you want the search to include the entire organization.

A related control type is the Search Match control type. This type is the Search Control control type with an additional feature that allows automatic searching and populating the list box of an attribute.

Search Match

Similar to the Search Control control type, with an additional feature that allows automatic searching and populating of the list box of an attribute. Users can use the automatic searching feature by typing in the first few letters of the wanted value in the text field and clicking **Add**. If one result is found, the result is automatically added to the list box. If more than one result is found, a Search Results page is displayed. A user can then select which items to add to the list box.

Provides a text field search page for the selected attribute. Users populate the text field by selecting the wanted search result. In the resulting form, the **Search** button opens a search page with the search type already

selected. The **Clear** button clears the text field. The **Delete** button is used to remove a selected item from the list box.

When you select this control type in the form designer applet, the Search Control Editor page is displayed. You can use the fields in the editor to configure the control type to default to a specific type of search. The Search Control Editor contains the following fields:

Category
Select the category for the search.

Profile
Select the profile to use for the search.

Attribute
Select the attribute to use for the search.

Operator
Select the operator, such as **Contains** or **Equals**, that links the **Attribute** and Value **fields** together.

Value   Type the value for the attribute.

Type   Select the type of attributes to be returned. A single-value type provides a text field for the user to populate. A multi-value type provides a list box of attributes. In this scenario, users can identify which attributes to search by selecting the attributes that they do not want to include in the search and clicking the **Delete** button. Deletion removes the selected attributes from the list of searchable attributes.

Search entire organization (current container only if not checked)
Select this check box if you want the search to include the entire organization.

A related control type is Search Control.

SubForm

The SubForm control type provides a means to use custom user interfaces for complex multi-valued attributes. Some Security Identity Manager adapters use this control type infrequently.

SubForm is a special control type used to start a Servlet, JSP, or static HTML page from a popup window that opens from a custom Security Identity Manager form. Subforms provide a means to submit an arbitrary number of parameter names and values to a custom Servlet or JSP. Subforms are used to create custom user interfaces for complex multi-valued attributes.

Table 27. SubForm parameters

Parameter	Description	Value
customServletURI	The URI to the Servlet, JSP, or static HTML page to be started from the main form. If a Servlet is implemented and deployed in the default web application for IBM Security Identity Manager, the value for this parameter is the same as the *URL-pattern* value defined in web.xml in the *servlet-mapping* tag, without the slash (/). If a JSP is implemented, the value for this parameter is the JSP file name that includes the jsp file extension. This parameter is required on all subforms.	Servlet name or JSP file name, such as `sample.jsp`
Parameter Name	Arbitrary parameter name and value that is included in the HTTP request that starts the resource at customServletURI.	*Parameter Value*, such as racfconnectgroup servlet

TextArea

Places a text area next to the attribute. A text area is a multiline text field used to gather user input and display data previously gathered.

TextField

Places a text field next to the attribute. A text field is a single-line area used to gather user input or display data previously gathered.

UMask

Allows a user to define UNIX access rights to files and directories.

Properties used by the form designer

Use the Properties page to configure attribute format and constraints.

The Properties page includes the following tabs:

Format

Use this tab to change the format of a form. Available fields in this tab are:

Name Use this field to add or modify the name of an attribute. This value is the identifier that the form uses to process LDAP attributes.

Data Type

Use this field to add or modify the data type of an attribute, such as Directory String, Distinguished Name, binary code, or another data type

Label Use this field to add or modify a user-readable label for the attribute. For example, $homepostaladdress, where the $ (dollar) symbol indicates a key to look up a string in a resource bundle.

Size Use this field to add or modify the visible width in units of pixels

for the following control type: TextField, Password, Search Control, and Search Match. Size represents the number of visible items for the following control type: ListBox and Editable TextList.

Rows Use this field to add or modify the value used by the TextArea control type to represent the number of visible text lines.

Cols Use this field to add or modify the value used by the TextArea control type to represent the visible width in average character widths.

Width Use this field to add or modify the value used by the SubForm control type to represent the width of a pop-up window in units of pixels.

This property is also used by the DropDownBox, EditableTextList, ListBox, SearchControl, and SearchMatch controls to represent the width of their associated combo boxes, in pixels. For EditableTextList and SearchMatch controls, width also determines the width of associated text boxes in pixels.

If width is not specified, it is assumed to be a default of 300 pixels. If the width for these controls is set to 0, the associated combo boxes are not a fixed size and resize dynamically. The size depends on the options added.

Height
Use this field to add or modify the value used by the SubForm control type to represent the height of a pop-up window in units of pixels

Read-Only on Modify
Select this check box to set an attribute to read-only. Only the label is displayed in the form, and users cannot modify the attribute value.

Direction
Select the direction of text:

> **inherit** displays text in the same direction as the form category to which the attribute belongs

> **ltr** displays text from left to right

> **rtl** displays text from right to left

Hide on Modify
Select this check box to hide the attribute field in the form when the form is in modify state. For example, if you select this check box for the Owner field within a service form, the Owner field is displayed when users create a service. The field is not displayed when users change a service.

Constraints
Use this tab to enter values for constraint fields to guarantee the type of data and the syntax of the data users are allowed to enter in form fields. Custom constraints are field-level data restrictions of various types. When you select a control type of **Search Control**, **Search Match**, **ListBox**, or **DropDownBox**, all of the constraint fields are disabled, except for the **Required** constraint.

Required

Select this check box to prevent the form from being submitted unless some value is typed into the field where this constraint is placed.

Validate and Update Constraints

In the field next to the **Validate and Update Constraints** button, which is at the bottom of the constraint type list, type a sample value for the attribute you selected from the form template layout area and click the **Validate and Update Constraints** button. This tests the value entered against the constraints activated for the attribute. If the test value you enter complies with all constraints, a message indicates success after you click the **Validate and Update Constraints** button.

Constraints fall into one of these general categories:

Syntactic constraints

Allow only values that conform to rules that define sequences of characters and structured parts.

E-mail address

Select this check box to guarantee that the syntax of the value supplied on the field where this constraint is placed complies with the following rules:

- Has one @ sign
- Invalid characters, such as < > ( ) . ; " \ [ ] do not occur before the @ sign
- The @ sign must be followed by a valid domain name or IP Address

IP address (IPV4)

Select this check box to guarantee that the value entered in the field where this constraint is placed is a valid IPV4 address of the form 127.0.0.1. The four octets are separated by a dot and none of the octets exceeds 255.

IP address (IPV6)

Select this check box to guarantee that the value entered in the field conforms with the text representation of IP addresses defined in RFC 2373. For example, 0:0:0:0:0:0:0:1 is the loopback IPV6 address. See RFC 2373 for more details.

Domain name

Select this check box to ensure that the value entered in the field where this constraint is placed is compliant with the Windows NT Server Domain Name syntax. The name must have two leading back-slashes (\\) and can contain up to 15 characters, except for these characters: " / \ [ ] : ; | = , + * ? < >

The name cannot consist solely of periods or spaces.

Invalid characters

Type characters in this field to define characters that is not valid when entered for the field.

DN Select this check box to guarantee that the value entered in this field conforms with the distinguished name structure. For example, cn=*common name*, ou=*organizational name*, o=*organization*.

Data type constraints

Allow values that occur within a range of characters or numbers.

ASCII-Only

Select this check box to constrain the characters allowed in the field to ASCII.

ASCII7

Select this check box to constrain the characters allowed in the field to ASCII-7.

ASCII8

Select this check box to constrain the characters allowed in the field to ASCII-8.

Integer only

Select this check box to allow only integers in the field.

Numeric

Select this check box to allow only numbers in the field.

Date range

Type a date range to force an ending date to be after a beginning date.

Value constraints

Require a parameter, such as Max Length = 10, where 10 is the parameter to constrain the value by.

Invalid characters

Type characters that are disallowed.

Maximum length

Type a numeric value that constrains the length of the value entered for the field to the number of characters specified.

Minimum length

Type a numeric value that prevents the form from being submitted unless the value entered has at least as many characters specified by this constraint.

Maximum value

Type a numeric value to set a high end point on the value entered (is at most *n*).

Minimum value

Type a numeric value to set a low end point on the value entered (is at least *n*).

Maximum lines

Type a numeric value to guarantee that the value entered on the form does not exceed the maximum number of lines specified (in a multi-line field).

No white space

Select this check box to disallow any white space from being entered on the form.

Properties that change the form designer user interface

IBM Security Identity Manager has properties that determine the interface appearance of the form designer.

In the `ui.properties` file, these properties change the appearance of the form designer user interface:

express.java.formDesignHeightIE
> Height in pixels of the form designer applet for Internet Explorer

express.java.formDesignWidthIE
> Width in pixels of the form designer applet for Internet Explorer

express.java.formDesignHeightMZ
> Height in pixels of the form designer applet for Mozilla

express.java.formDesignWidthMZ
> Width in pixels of the form designer applet for Mozilla

Chapter 7. Managing manual notification templates

Use this task to modify the default email messages displayed for manual services.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

You can modify the default messages that are displayed for manual services. By modifying the templates you can apply the changes to any manual service that you create. You do not need to modify the messages each time you create a manual service, unless the service requires a specific message change.

Note: Changes to the notification templates do not affect the messages for existing manual services.

Procedure
1. From the navigation tree, click **Configure System** > **Configure Manual Notification Templates**. The Templates page is displayed.
2. Select the operation and click **Change**. The Template Modify page is displayed.
3. In the **Subject** field, modify the text to specify the subject of the email notification that is sent. The subject can consist of plain text and dynamic content tags.
4. In the **Plaintext body** field, modify the text to be displayed in the body of the message. The content can consist of plain text, dynamic content tags, and JavaScript code. These contents are shown to email recipients that do not see HTML email notifications.
5. In the **XHTML body** field, type the text to be displayed in the body of the email notification as HTML. The content can consist of plain text, dynamic content tags, and JavaScript code. These contents are shown to email recipients that see HTML email notifications.
6. Click **OK** to save the changes. You are returned to the Templates page.

What to do next

Change the notification template for another operation or click **Close** to exit.

Chapter 8. Entities management

An *entity* is a person or object for which information is stored.

While there are many types of system entities, such as policy and workflow, only the following entity types are provided for customization:

- Account
- BPPerson (Business Partner Person)
- BusinessPartnerOrganization
- Organization
- Person
- Service

System administrators can customize existing system entities by selectively mapping entity attributes to custom LDAP class attributes. System administrators can also create new Person and BPPerson (Business Partner Person) custom entities. The administrator associates unique entity names with the standard IBM Security Identity Manager entity types.

Adding system entities

Create new Person and BPPerson entities to associate with a new custom LDAP class.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

When you add a Person or BPPerson type entity, the actual LDAP class that stores the entity must be created before you use this task to add entities.

Custom LDAP classes and their attributes must be created directly within your data store with tools compatible with your LDAP data repository software. Create the classes before associating them with a custom IBM Security Identity Manager entity. After it is created, the class can be associated with a custom Security Identity Manager entity. Map its attributes to Security Identity Manager attributes.

About this task

All LDAP classes, auxiliary, and structural, that begin with er are considered Security Identity Manager-managed classes. They are excluded from the list of LDAP classes within the Manage Entities task.

When adding a custom entity, you need to examine the default control type of each attribute. Change it to an appropriate control type from the form customization page. Refer to a standard Security Identity Manager entity of the same entity type as the custom entity to view the control types assigned to the attributes of a standard entity.

To add a custom system entity, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Entities**. The Manage Entities page is displayed.
2. On the Manage Entities page, click **Add**. The Create Entity wizard is displayed.
3. On the Select Type page, select the entity type that you want to create, and then click **Next**.
4. On the Entity Detail Information page, complete the following steps:
   a. In the **Entity name** field, type a unique name for the entity.
   b. Click **Search** to find and specify an LDAP class that stores the entity.
   c. On the Select LDAP Class page, click **Search** to display a list of LDAP classes.
   d. Select the object class name, and then click **OK**. The **LDAP class** field is populated with the object class name that you specified.
   e. Click **Browse name attributes** to find and specify Valid entries for the **Name attributes** field depend on which LDAP class is selected. The Select Attribute page is displayed, which lists the name attributes of the LDAP class that you selected.
   f. On the Select Attribute page, select the name attribute that you want to associate with the new entity, and then click **OK**. The **Name attribute** field is populated with the name attribute that you selected.
   g. In the **Default search attributes** list, select the search attributes that you want to add to the entity, and then click **Add**. Select attributes that are searchable, such as string or numeric type.
   h. When you are finished specifying entity information, click **Next**.
5. On the Attribute Mapping page, map an attribute by completing these steps:
   a. Select an attribute in the **Identity Manager attribute** list.
   b. Select an attribute in the **Custom LDAP attribute** list.
   c. Click **Map**.
   d. Optional: To obtain the default mapping, select an attribute pair in the table, and click **Reset**.
   e. When the mapping is complete, click **Finish**.

Results

A message is displayed, indicating that you successfully created an entity.

What to do next

Perform additional entity management tasks, or click **Close**.

Changing system entities

View and change characteristics of entities.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

You cannot change the entity type due to the associated schema definition. Instead, you must delete the entity and create an entity with the wanted type.

To change an existing entity, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Entities**. The Manage Entities page is displayed.
2. On the Manage Entities page, select the check box next to the entity that you want to modify, and then click **Change**. The Change Entity notebook is displayed.
3. Click the tab for what you want to edit.
   - **Entity Detail Information**: View or assign LDAP class and name attribute.
   - **Attribute Mapping**: Map attributes to customer LDAP attributes.
   - **Attribute Auditing**: Exclude attributes that are too long (> 4000 bytes) from auditing.

     Note: The **Attribute Auditing** tab appears only for entities that belong to an **Entity type** of **Account**, **Business Partner Person**, or **Person**.
4. Change the entity and then click **OK**.

What to do next

A message indicates that you successfully updated the entity.

Perform additional entity management tasks, or click **Close**.

Attribute auditing

Exclude attributes from auditing for entities of entity type Account, Business Partner Person, or Person.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

By default all attributes are included in the audit process. If an attribute value might exceed 4000 bytes, remove it from the **Audited attributes** list.

Note: The audit process fails if it encounters an attribute that exceeds 4000 bytes.

To exclude attributes from auditing, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Entities**. The Manage Entities page is displayed.

2. On the Manage Entities page, select the check box next to the entity that you want to modify, and then click **Change**. The Change Entity notebook is displayed.

3. Click the **Attribute Auditing** tab.

   Note: The **Attribute Auditing** tab appears only for entities that belong to an **Entity type** of **Account**, **Business Partner Person**, or **Person**.

4. Remove the desired attribute from auditing.
   - To remove an attribute from auditing, select the desired attribute in the **Audited attributes** list, then click **< Remove**. It is moved to the **Available attributes** list.
   - To restore a removed attribute, select the desired attribute in the **Available attributes** list, then click **Add >**. It is moved to the **Audited attributes** list.

5. Click **OK** to save your changes.

What to do next

A message indicates that you successfully updated the entity.
- Click **Manage entities** to perform additional tasks.
- Click **Close** to return to the **Home** panel.

Effect on viewing attributes in the Identity Service Center

The Identity Service Center (ISC) user interface shows a substitute value for attributes removed from auditing: `Non-audited attribute`.

If a value was entered for the attribute, the value is retained but not shown . If the attribute is returned to the **Audited attributes** list, then the value is shown in the ISC.

Deleting system entities

Delete system entities from the IBM Security Identity Manager system.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

You cannot delete a system entity if there are dependent units that exist in that entity.

To delete a system entity, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Entities**. The Manage Entities page is displayed.

2. On the Manage Entities page, select the check box next to the entity that you want to delete, and then click **Delete**. Selecting the check box at the top of this column selects all system entities.

3. On the Confirm page, click **Delete** to delete the entity, or click **Cancel**.

Results

A message indicates that you successfully deleted the entity.

What to do next

Perform additional entity management tasks, or click **Close**.

Customizing role schema

Administrators customize a role schema by adding optional attributes to the IBM Security Identity Manager LDAP and then to the role definition schema (*erRole* objectclass).

About this task

Procedure

1. Access the Security Identity Manager LDAP.
2. Add new optional type attributes. For example, add the attribute `designation`. For more information, see *LDAP Installation and Configuration Guide*.
3. Update the *erRole* objectclass in theSecurity Identity Manager LDAP to associate the new attributes. For example, update the *erRole* objectclass in IBM Security Directory Server by using the Security Directory Server web administrative console and associate the `designation` attribute with the *erRole* objectclass. For more information about Security Directory Server, see the IBM Knowledge Center.
4. Ensure that the role schema is customized correctly.
5. Ensure that Security Identity Manager and Security Identity Manager LDAP are running.
6. Launch the Security Identity Manager administrative console.
7. Select **Configure System > Design Forms**.
8. Update the role form template to display the new attribute.

    Note: The LDAP schema is cached, and you might need to restart Security Identity Manager to see the new attribute that was added to the *erRole* objectclass.

Results

You can view the new attributes on the Security Identity Manager administrative console when viewing the role definitions.

What to do next

You can define, set, modify, save, and restore custom attributes when creating or modifying a role.

Chapter 9. Account category management

You can configure account categories to further categorize individual accounts that you want to exclude from password synchronization when **Password synchronization** is turned on.

Accounts with the category that is defined in `enrole.properties` are excluded.

If you configure one or more account categories, IBM Security Identity Manager prompts you to select the account category when you request a new account. Users can also change the account category when they manage their accounts.

Chapter 10. Ownership type management

Ownership types classify the accounts. Use the **Manage Ownership Types** task to classify ownership types in your organization. If you configure multiple account ownership types, IBM Security Identity Manager prompts users to select the ownership type when requesting a new account or assigning accounts to users.

Security Identity Manager includes the following :
- Device
- Individual
- System
- Vendor

As an administrator, you can create additional ownership types.

An account can have only one type of ownership. The ownership type depends on the intended use of the account. The type of ownership affects the password management process. For example, password synchronization provides change of password for accounts that have the ownership type, "Individual".

Creating ownership types

As an administrator, you can create additional ownership types.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To get access to this task, or to have someone complete this task for you, contact your system administrator.

About this task

To create an ownership type, complete these steps:

Procedure

1. From the navigation tree, click **Manage Ownership Types**. The **Manage Ownership Types** page displays the default ownership types.

   The default ownership types are:
   - Device
   - Individual
   - System
   - Vendor
2. Click **Create**. The **Create Ownership Type** page is displayed.
3. Complete the following steps:
   a. At **Ownership Type Key**, type a custom name for the ownership type.
   b. (Optional) At **Description**, type a description for the ownership type.
4. Click **OK** to save the new ownership type.

Results

A message indicates that you successfully created an ownership type. The new ownership type is displayed on the **Manage Ownership Types** page.

What to do next

Create or modify additional ownership types, or click **Close**.

Deleting ownership types

When ownership types are no longer valid, administrators can delete all ownership types, except Individual.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To get access to this task, or to have someone complete this task for you, contact your system administrator. An ownership type can be deleted only if it is not associated with any account.

About this task

You cannot delete an ownership type if it is associated with an account.

Procedure

1. From the navigation tree, click **Manage Ownership Types** to display the page that lists the currently defined ownership types.
2. Select the ownership type you want to delete:
   a. To select a specific one, select the check box next to it.
   b. To select all, click the box at the top of the column.
3. Click **Delete**. A confirmation page is displayed.
4. On the Confirm page, take one of the following actions:
   a. Click **Delete** to delete the ownership type.
   b. Click **Cancel** to stop the deletion process.

Results

A message indicates that you successfully deleted the ownership type. The **Manage Ownership Types** page no longer displays the deleted ownership type.

What to do next

You can create or delete an ownership type.

Chapter 11. Operations management

You can configure operational workflows for IBM Security Identity Manager system entities and entity types. You can customize the out-of-box entity type operations to implement the security requirements of your organization.

An *operation* is an action that can be done on an entity. Operations that are defined for a specific entity type are used by all entities of the specified type. However, if an operation is defined for a specific entity, the operation takes precedence over the entity type operation.

System administrators can create new or modify existing operations for entities and entity types.

You can customize operations for Account, Person, and Business Partner Person entity types. The standard operations apply to these entities unless you define a customized operation at the entity level.

Beginning with IBM Security Identity Manager 6.0.0.3, you can define access control for custom operations when you create access control items.

Add operation

The add operation is initiated any time an add request is submitted for a specified type of entity. For example, an add operation for the person entity type is initiated when a new user is added to the system.

The default workflow for the add operation depends on the type of entity that is added.

The default workflow for Person and Business Partner Person entity add operations use the `createPerson` and `enforcePolicyForPerson` workflow extensions.

Figure 10. Person and Business Partner Person add operation workflow

The default workflow for the account entity add operation uses the `createAccount` workflow extension.

Figure 11. Account add operation workflow

Figure 12. Identity Manager User add operation workflow

177

changePassword operation

The changePassword operation is initiated any time a password change request is submitted for an account entity.

The default workflow for the changePassword operation uses the `changePassword` workflow extension.

Figure 13. changePassword operation workflow

Delete operation

The delete operation is initiated any time a delete request is submitted for a specified type of entity.

The default workflow for the delete operation uses the `deletePerson` or `deleteAccount` workflow extension.

Figure 14. Account delete operation workflow

Figure 15. Person and Business Partner Person delete operation workflow

Modify operation

The modify operation is initiated any time a request to modify an entity is submitted.

The default workflow for the modify operation depends on the type of entity that is modified.

The default workflow for the account entity uses the `modifyAccount` workflow extension.

Figure 16. Account modify operation workflow

The default workflow for Person and Business Partner Person use the `modifyPerson` and `enforcePolicyForPerson` workflow extensions.

Figure 17. Person and Business Partner Person modify operation workflow

Restore operation

The restore operation is initiated any time a restore request is submitted for a specified type of entity.

The default workflow for the restore operation uses the `restorePerson` or `restoreAccount` workflow extension.

Figure 18. Account restore operation workflow

Figure 19. Person and Business Partner Person restore operation workflow

selfRegister operation

The selfRegister operation is used when individuals attempt to add themselves in IBM Security Identity Manager. This operation is available only for a User or Business Partner Person entity.

The default selfRegister operation has these steps:
1. Creating a person entity
2. Verifying that the person entity complies with existing policies

Before you use the selfRegister operation, the start element or the transition line between the start element and the createPerson extension element must have JavaScript code that calculates the container to which the person entity is added. The JavaScript code can be a PostScript in the start element or a custom definition for the transition line.

This diagram illustrates the default workflow for the selfRegister operation.

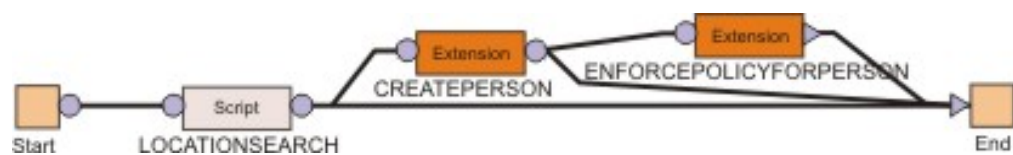*Figure 20. selfRegister operation workflow*

Suspend operation

The suspend operation is initiated any time a suspend request is submitted for a specified type of entity.

The default workflow for the suspend operation uses the `suspendAccount` or `suspendPerson` workflow extension. This diagram illustrates the basic suspend operation workflow.

Figure 21. Account suspend operation workflow

Figure 22. Person and Business Partner Person suspend operation workflow

Transfer operation

The transfer operation is initiated any time a transfer request is submitted for a Person or Business Partner Person entity.

The default workflow for the transfer operation uses the `transferPerson` and `enforcePolicyForPerson` workflow extensions.

Figure 23. Person and Business Partner Person transfer operation workflow

Adding operations for entities

System administrators add entity operations.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Defining a new operation, for example, might add an operation to recertify a person or account entity. You specify an approval workflow that either approves or suspends the entity.

To add an operation for an entity, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Operations**. The Manage Operations page is displayed.
2. On the Manage Operations page, select one of the following operation levels:
   - Select **Global level** to define an operation that is applicable to all entities and entity types. Global operations do not implicitly affect any entities unless they are explicitly started within entity type or entity level operations. Global operations can also be called in a Lifecycle Rule.
   - Select **Entity type** level to define an operation at the entity type level. Select an entity type from the **Entity Type** list.
   - Select **Entity level** to override the operations that are defined at the entity type level. Select an entity type from the **Entity Type** list, and then select an entity from the **Entity** list.
3. Click **Add**. The Add Operation page is displayed.
4. In the **Operation Name** field, Type a name of the workflow operation that you want to define for the corresponding system entity. To override an operation

that is defined at the entity type level, enter the operation name that you want to override, and then click **Continue**. The Define Operation page is displayed, and the workflow designer Java applet is started.

5. In the workflow designer, define the workflow process, and then click **OK**. To define an operation workflow process, drag the design nodes from the node palette onto the operation design space. Then, connect them with transition lines. After you place a design node on the operation design space, double-click the node to configure its properties. Make sure that all the nodes are connected and all the required properties are set for each node. Ensure that the transition condition is set for each link.

Results

A message indicates that you successfully created an operation for the specified level. Click **Close**.

What to do next

When the Manage Operations page is displayed, click **Refresh** to refresh the **Operations** table and display the new operation.

Changing operations for entities

System administrators can change existing entity operations.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To change operation for an entity, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Operations**. The Manage Operations page is displayed.
2. On the Manage Operations page, select **Global level**, **Entity type level**, or **Entity level** to list the operations that you want to modify.
3. Select the check box next to the operation that you want to modify, and then click **Change**. Selecting the check box at the top of this column selects all operations. The Define Operation page is displayed, and the workflow designer Java applet is started.
4. In the workflow designer, modify the operation for the system entity, and then click **OK**. To define an operation workflow process, drag the design nodes from the node palette onto the operation design space. Then, connect them with transition lines. After you place a design node on the operation design space, double-click the node to configure its properties. Make sure that all the nodes are connected and all the required properties are set for each node. Ensure that the transition condition is set for each link.

Results

A message indicates that you successfully updated the operation for the entity. Click **Close**.

What to do next

When the Manage Operations page is displayed, click **Refresh** to refresh the **Operations** table.

Deleting operations for entities

System administrators can delete an existing entity operation.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Only user-defined operations can be deleted.

To delete an operation for an entity, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Operations**. The Manage Operations page is displayed.
2. On the Manage Operations page, select **Global level**, **Entity type level**, or **Entity level** to list the operations that you want to delete.
3. Select the check box next to the operation that you want to delete, and then click **Delete**. Selecting the check box at the top of this column selects all operations. A confirmation page is displayed.
4. On the Confirm page, click **Delete** to delete the operation, or click **Cancel**.

Results

A message indicates that you successfully deleted the operation for the entity. Click **Close**.

What to do next

When the Manage Operations page is displayed, click **Refresh** to refresh the **Operations** table.

Specifying a custom operation and its access control item

You can specify custom operations such as a modify operation that is available on a global basis or entity type basis to help you manage IBM Security Identity Manager business objects. For example, you might create a global operation to help manage business partner organizations when you start or end their use by your company.

About this task

In a business environment that continually changes third-party workers, administrators can create a global operation that creates a business partner organization container. Then, they can create all contractual workers and provision accesses with a single global operation.

When use of the business partner ends, administrators can create a global operation that removes all contractual worker accesses and then removes the organization container.

Procedure

1. Click **Configure System > Manage operations**.

   In the **Operation Level** choices, select **Global Level**. Then, create a new operation as required by your business process. Alternatively, select the Entity or Entity type to create a custom operation to manage the object type as required by your business process.

2. Click **Set System Security > Create Access Control Item > General**.

   Specify the name of the access control item, such as `ACI to Sunset Contractors`. Specify the protection category as **Global operation** or select the entity type of your custom operation.

3. Click **Set System Security > Create Access Control Item > Operations**.

   In the table of available operations, select the operation that you defined earlier, such as `ACI to Sunset Contractors`. In the **Permission** column, specify **Grant**.

4. Click **Set System Security > Create Access Control Item > Membership**.

   Specify the group of IBM Security Identity Manager users that can run the global operation.

Chapter 12. Lifecycle rules management

Lifecycle rules can be used to automate the large number of manual tasks that administrators must make due to common recurring events. Such events can be account inactivity, password expiration, or contract expiration, which are driven by business policies. Lifecycle rules can also eliminate the potential of some policies to go unenforced.

Overview

Establishing lifecycle rules enables administrators to define events that can be triggered based on a time interval or based on time and matching criteria evaluated against an entity. The administrator can then associate lifecycle operations to run as a result of that event. All lifecycle rules consist of two parts:

* The definition of an event that triggers the rule
* The identification of the lifecycle operation that runs the actions specified in the rule

Each rule can be defined in one of these ways:

* Global
* Associated with an entity type
* Associated with an entity

For global rules, an event is defined by a time interval. For example, once a month, or on every Monday at 8:00 a.m. Global lifecycle rules are independent of any particular system entity. The lifecycle operations that can be invoked by a global rule must also be global in nature because there is no context available to call an entity- or entity type-based operation.

Entity and entity type rules also have an event with a time interval. However, the goal of these rules is to affect multiple entities at one time.

Matching criteria for events

A separate event is triggered for each lifecycle object. To prevent events from occurring for possibly thousands of objects that might not be related to the rule, a matching criteria is available for these events.

Without the matching criteria, every object of the specific entity or entity type has the associated lifecycle operation done on it.

With the criteria, only objects that meet the criteria have the operations done. The criteria is defined with an LDAP filter syntax. The filter identifies any objects that meet the criteria and causes the event to be triggered for only those objects. If no object matches the filter, the event is not triggered. For example, the criteria might be for any accounts where (erAccountStatus=1), which means the accounts are suspended.

Lifecycle rule filters and schedules

Because the filter is based on attributes, only the attributes associated with the schema of the entity or entity type are accepted.

There might also be the need to include environment data or external data into the filter. For example, you might need to include the current time or a value obtained from a customer database. The inclusion of this data is achieved by allowing macros to be placed in the filter. For example, a filter checking if a password changed within the last 30 days might read as follows: (erPswdLastChanged>=${system.date - 30}).

Note: Leaving the filter blank returns all entities. Entity relationship macros can be used in lifecycle rule filters.

The interval defined for an event can be constructed from the following options:

Daily Triggers the lifecycle event every day. After you select this option, click the clock icon to specify a time in the **At this time** field.

Weekly
> Triggers the lifecycle event once a week. After you select this option, select a day from the **On this day of the week** list, and then click the clock icon to specify a time in the **At this time** field.

Monthly
> Triggers the lifecycle event once a month. After you select this option, select a date from the **On this day of the month** list, and then click the clock icon to specify a time in the **At this time** field.

Hourly
> Triggers the lifecycle event once an hour. After you select this option, select a time from the **At this minute** list.

Annually
> Triggers the lifecycle event on a specific date and time of the year. After you select this option, select a month from the **Month** list. Then select a date from the **On this day of the month** list, and then click the clock icon to specify a time in the **At this time** field.

During a specific month
> Triggers the lifecycle event on a specific month, day, and time. After you select this option, select a month from the **Month** list. Then select a day from the **On this day of the week** list, and then click the clock icon to specify a time in the **At this time** field.

Quarterly
> Triggers the lifecycle event four times per year on a specific day and time of the quarter. The reconciliation will occur on the specified day past January 1, April 1, July 1, and October 1. After you select this option, select a day from the **On this day** list, and then click the clock icon to specify a time in the **At this time** field.

Semi-Annually
> Triggers the lifecycle event two times per year on a specific day and time of the half-year. The reconciliation will occur on the specified day past January 1 and July 1. After you select this option, select a day from the **On this day** list, and then click the clock icon to specify a time in the **At this time** field.

Note: More than one schedule can be specified.

A lifecycle rule evaluation schedule contains only a reference to a corresponding rule definition. If a lifecycle rule definition changes before the scheduled evaluation

starts, the evaluation uses the updated version of the definition. It does not use the rule definition that was originally scheduled.

In this example, a lifecycle rule is created. It checks once a day for accounts with no password changes in 90 days. An email notification is sent to owners of accounts that meet the lifecycle rule search criteria, informing them that they must change their passwords.

First, a lifecycle operation named `remindToChangePassword` is constructed for the Account entity type. It is defined as an instance-based (not static) operation, and so it has the account object itself as an input parameter. The business logic of the operation is defined with one work order activity that sends the reminder message to the owner of the account. It includes the user ID of the account in the message.

A lifecycle rule is then constructed for the Account Entity Type named `passwordExpiration` that references the `remindToChangePassword` operation. It has an event with an evaluation interval of **daily** at **12:00 A.M.**. It also has the following filter: `(&(erAccountStatus=0)(erPswdLastChanged<=${system.date - 90}))`.

Lifecycle rule processing

Lifecycle rule operations can take an extended period to finish for the entire result set returned from the evaluation of the lifecycle rule filter.

Completion is primarily due to the time it takes to complete manual workflow activities associated with the operation. A lifecycle rule evaluation might be scheduled or manually initiated to run again before operations that result from the first lifecycle rule evaluation are completed for all targets. The second iteration of the lifecycle rule evaluation identifies those targets that remain in a working state from the original evaluation. The second iteration does not initiate the lifecycle operation again for those targets. It will, however, initiate for any targets it identifies during the lifecycle rule evaluation period that are not in a working state.

For example, a lifecycle rule might discover 100 entities that match its criteria. The rule proceeds to initiate the operation associated with the rule for those 100 entities. Assume that 10 entities are added to the system. Addition occurs after the initial lifecycle evaluation and while the lifecycle rule operation is being applied to the original 100 entities. A second iteration of the lifecycle rule might be initiated before the first iteration is complete. The second iteration skips over any entities that have the operation of the lifecycle rule initiated from the first iteration. The second iteration skips entities until it discovers an entity that matches the lifecycle rule filter evaluation but does not currently have this lifecycle rule (matches on rule name) running against it. In this case, the second iteration discovers and initiates for the 10 new entities that were added.

This behavior is important to understand because there might be occasions where the second iteration of a lifecycle rule might complete before the first iteration. Theoretically, the lifecycle rule evaluation you schedule for 10:00 AM might complete before the lifecycle rule evaluation scheduled for 9:00 AM. Do not assume that a lifecycle rule operation is complete for all relevant targets based upon the completion of a subsequent iteration of the same lifecycle rule. To verify which request items are complete and which items are disregarded, check the audit log of the completed request.

Lifecycle rule modification

A modification to the filter or operation of a lifecycle rule will not take effect until the next time the lifecycle rule is evaluated.

The lifecycle rule might be actively evaluated by the system when the modification is made. The currently running evaluation continues to use the previous definition of the lifecycle rule until it completes. The workflow might change for the operation while the lifecycle rule is actively being evaluated by the system. The change affects the currently running evaluation at whatever point the change is made. For example, if the lifecycle rule filter identifies 50 Persons and the operation of the lifecycle rule operation is named `Recertify`. Changing the operation name to `CheckPassword` does not affect the current iteration of the rule. The change takes place the next time the rule is initiated. However, changing the workflow for the `Recertify` operation while it is active might result in 25 Persons being processed under the original workflow. The remaining 25 Persons are processed under the new workflow.

Lifecycle rule implementation has a key dependency on having the database contain the scheduling information. Removing, dumping, or otherwise purging the table that contains lifecycle rule scheduling information deactivates the associated lifecycle rules. If these changes occur, you must reconfigure all lifecycle rules and redefine their schedules.

If the operation associated with a lifecycle rule is deleted or renamed, the operation cannot be implemented within the lifecycle rule until the rule is reconfigured.

Note: When you add or modify a lifecycle rule for an entity, the updates you make take effect after the cache times out (10 minutes, by default).

Lifecycle event schema information

IBM Security Identity Manager supplies specific schema attributes that facilitate the creation of lifecycle events.

These attributes are managed by IBM Security Identity Manager Server and are made available through Data Services and from the lifecycle event interface. The following is the list of additions:

- `erPersonItem`
  - `erCreateDate` – Date the person was added to the system
  - `erLastStatusChangeDate` – Date the state of the person was last changed. The timestamp is updated whenever the person is restored or suspended.
  - `erlastoperation` – Available for custom use
  - `erpswdlastchanged` – Date the synchronized password of the person was last changed
- `erAccountItem`
  - `erCreateDate` – Date that the account was added to the system
  - `erLastStatusChangeDate` – Date that the state of the account was last changed. The timestamp is updated whenever the account is restored or suspended.
  - `erlastoperation` – Available for custom use

Except for the custom use items, these schema items are managed by the system.

Adding lifecycle rules for entities

Use these instructions to define lifecycle rules for entities.

Before you begin

Only system administrators can perform this task.

About this task

Lifecycle rules trigger operations that are defined in the Manage Operations task. Depending on the type of lifecycle rule, the corresponding operations defined at the level are available.

Lifecycle rules are different from operations. The lifecycle rule that is defined at entity type or entity level does not override the lifecycle rule defined at a higher level. Each level has valid lifecycle events that can run independently based on the schedule that is defined.

To add a lifecycle rule for an entity type, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Life Cycle Rules**. The Manage Life Cycle Rules page is displayed.
2. On the Manage Life Cycle Rules page, select one of the following lifecycle rule levels:
   - Select **Global level** to define a lifecycle rule that has no entity context.
   - Select **Entity type** level to define a lifecycle rule that is applicable to the entity type. Select an entity type from the **Entity Type** list.
   - Select **Entity level** to define a lifecycle rule that is applicable to a specific entity instance type. Select an entity type from the **Entity Type** list, and then select an entity from the **Entity** list.
3. Click **Add**. The Manage Life Cycle Rules notebook is displayed.
4. On the **General** page of the Manage Life Cycle Rules notebook, complete these steps:
   a. In the **Name** field, type a unique name for the lifecycle rule that you want to define for the corresponding system entity.
   b. Optional: In the **Description** field, type a description for the lifecycle rule.
   c. From the **Operation** list, select an operation to be invoked when the event occurs. Only operations without input parameters are allowed to be run by the lifecycle rule.
   d. Click the **Event** tab.
5. On the **Event** page of the Manage Life Cycle Rules notebook, complete these steps:
   a. In the **Search filter** field, type an LDAP filter that identifies the objects that are affected by the event. For example, the following filter captures all active employees who did not change their passwords in the past 90 days. The capture is calculated from the date that the lifecycle event occurs:
   `(&(employeeType=active)(erPswdLastChanged<=${system.date - 90}))`

   Note: The Search filter is not applicable to global level lifecycle rules because global level lifecycle rules do not have entity context.

b. Click **Add** to define a schedule for the lifecycle rule. The Define Schedule page is displayed.

6. On the Define Schedule page, define a schedule for the lifecycle rule to run, and then click **OK**. The fields displayed depend on the scheduling option that is selected. The new schedule is displayed on the **Event** page of the Manage Life Cycle Rules notebook.

7. Click **OK** to save the lifecycle rule and close the notebook.

Results

A message is displayed, indicating that you successfully created a lifecycle rule for the entity. Click **Close**.

What to do next

When the Manage Life Cycle Rules page is displayed, click **Refresh** to refresh the **Life Cycle Rules** table and display the new lifecycle rule.

Changing lifecycle rules for entities

Use these instructions to change lifecycle rules.

Before you begin

Only system administrators can perform this task.

About this task

To change a lifecycle rule for an entity type, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Life Cycle Rules**. The Manage Life Cycle Rules page is displayed.

2. On the Manage Life Cycle Rules page, select the check box next to the lifecycle rule that you want to modify, and then click **Change**. The Manage Life Cycle Rules notebook is displayed.

3. Click the **General** tab or the **Event** tab.

4. Make the wanted changes, and then click **OK**.

Results

A message is displayed, indicating that you successfully updated a lifecycle rule for the entity. Click **Close**.

What to do next

When the Manage Life Cycle Rules page is displayed, click **Refresh** to refresh the **Life Cycle Rules** table.

Deleting lifecycle rules for entities

Use these instructions to delete lifecycle rules.

Before you begin

Only system administrators can perform this task.

About this task

To delete a lifecycle rule for an entity type, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage Life Cycle Rules**. The Manage Life Cycle Rules page is displayed.
2. On the Manage Life Cycle Rules page, select the check box next to the lifecycle rule that you want to delete, and then click **Delete**. Selecting the check box at the top of this column selects all lifecycle rules.
3. On the Confirm page, click **Delete** to delete the lifecycle rule, or click **Cancel**.

Results

A message is displayed, indicating that you successfully deleted a lifecycle rule for the entity. Click **Close**.

What to do next

When the Manage Life Cycle Rules page is displayed, click **Refresh** to refresh the **Life Cycle Rules** table.

Running lifecycle rules for entities

Use these instructions to run lifecycle rules.

Before you begin

Only system administrators can perform this task.

About this task

Running a lifecycle rule triggers the event immediately instead of running on a defined schedule.

To run a lifecycle rule for an entity type, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Manage lifecycle Rules**. The Manage lifecycle Rules page is displayed.
2. On the Manage lifecycle Rules page, select the check box next to the lifecycle rule that you want to run, and then click **Run**.
3. On the Confirm page, click **Run** to run the lifecycle rule, or click **Cancel**.

Results

A message is displayed, indicating that you successfully submitted the lifecycle rule to be run. Click **Close**.

What to do next

When the Manage lifecycle Rules page is displayed, click **Refresh** to refresh the **Life Cycle Rules** table.

LDAP filter expressions

IBM Security Identity Manager provides a built-in interpreter for general RFC 2254 LDAP filters and for two custom extensions to the filter syntax defined by the RFC.

The first extension provides a notation for variables in LDAP filters that reference IBM Security Identity Manager relationships. These variables resolve to related or connected objects. The second extension provides a notation for variables in LDAP filters. These filters reference a system object and a date keyword that resolve to the current date and time. The two extensions to the LDAP filter syntax are interpreted and evaluated at run time and are known as *filter expressions*. The filter expressions enable administrators to define filters with dynamic parts that reference useful abstractions in IBM Security Identity Manager. The two types of filter expressions that are supported are *relationship expressions* and *system expressions*.

Relationship expressions

The connection between IBM Security Identity Manager domain objects is given by a relationship.

The owner of an account, for example, is given by the owner relationship. The host service of an account is given by the service relationship. A role of a person is given by the role relationship.

In general,

Target Object	relationship	Related Object

For example:

Person	role	Role

Where a person is related to a role through the role relationship. Relationship expressions in filters provide a way to match up domain objects that are based on their relationship to other domain objects.

The connection between IBM Security Identity Manager domain objects is given by a relationship.

The filter expression syntax consists of an opening dollar sign ($) followed by a left curly brace ({) immediately followed by a relationship name, a dot (**.**) operator, then an attribute name followed by a right curly brace (}) to close the expression. For example:

`(${relationship.attribute}=value)`

`relationship` is the name of a relationship in IBM Security Identity Manager and includes:

- Parent
- Owner

- Organization
- Supervisor
- Sponsor
- Administrator
- Role
- Account
- Service

attribute is any attribute name that is valid for the related object. References to these connections or links between domain objects are often useful in searches. The references are useful in matching during authorization (in ACIs) and in lifecycle management (lifecycle rules) during operation execution.

In ACIs, relationship expressions are used to grant access to domain objects based in part on their relationship to another. For example, an ACI for a person that grants **Modify** with the following relationship expression used as the ACI filter grants permission to all people who have a supervisor, Jen Jenkins:

```
(${supervisor.cn}=Jen Jenkins)
```

Likewise, an ACI for an account that grants search with the following relationship expression used as the ACI filter grants permission to all accounts whose service (host) is named SuSE Server. Access is granted based on the relationship of one object to another.

```
(${service.erservicename}=SuSE Server)
```

In lifecycle management, relationship expressions are also used to match domain objects that are based on their relationship to other domain objects. The rules can start the same operation on all matches. For example, a lifecycle rule for a person where the operation is set to **Suspend** with the relationship expression effectively suspends all people in the Brokers role (dynamic or static) each time the lifecycle rule runs:

```
(${role.errolename}=Brokers)
```

Evaluation of relationship expressions

Relationship expression evaluation can be thought of as answering a yes or no question in four steps.

These steps are:
- What goes in (the expression itself)?
- What is being matched (the target object)?
- What comes out (the connected or related object)?
- Does the related object match the value to the right of the equal sign?

If so, the answer that is given by the evaluation is yes, and the target object is said to match the relationship expression.

The first column in the following table lists relationship expressions in a sample filter. The second column lists the type of objects valid for that expression. The third column shows the type of object to which the relationship points.

Table 28. Sample filter relationship expressions

Relationship Expression	Target Object	Related Object
(${parent.ou}=Sales)	Any (except Account)	Any container

Table 28. Sample filter relationship expressions (continued)

Relationship Expression	Target Object	Related Object
(${owner.cn}=John Smith)	Account	Person
(${organization.o}=Marketing)	Any (except Account)	Organization
(${supervisor.cn}=Jen Jenkins)	Any (except Account)	Person
(${sponsor.cn}=Pete West)	Any (except Account)	Person
(${administrator.cn}=Joe Peterson)	Any (except Account)	Person
(${role.errolename}=Brokers)	Any (except Account)	Role
(${account.uid}=JUser)	Any (except Account)	Account
(${service.erservicename}=SuSE Server)	Account	Service

The evaluation steps are important to keep in mind while composing relationship expressions. Most importantly, the related object type must be known to refer to a valid attribute name after the dot (.) operator to ensure that the expressions are valid and can produce a match. A view of the LDAP schema is a useful reference here. The system resolves relationship expressions to the first entity that fulfills the filter criteria. The system then queries for all objects that have the relationship in the filter for that entity. Be sure to create filters specific enough to return the entity that you intend to target.

Name keyword
One syntax variation for relationship expressions is the inclusion of the special name keyword that appears after the dot (.) operator.

Use of the name keyword after the dot (.) operator refers to the name attribute in a profile. This syntax is a general way to point to an object by name rather than through an explicit attribute name. This generality has the limitation, however, of being useful only in contexts where a profile is known at evaluation time.

For example, assume that you have an ACI for a Lotus Notes account. This ACI grants the ability to modify accounts and uses the following filter:

(${service.name}=SuSE Server)

The name keyword refers to the Lotus Notes service profile name attribute. It is valid to use name in this context. At authorization time (evaluation time), the Lotus Notes service profile is always known, and its name attribute can be resolved. The name keyword is not valid in lifecycle rules because the reference to the name attribute in a specific profile is ambiguous when the lifecycle rule is run. Therefore, the name attribute cannot be resolved.

Known limitations
A relationship expression that matches a lot of related objects might cause a very large LDAP filter to be generated.

In some instances the generated filter might be larger than the maximum length allowed by the directory server.

Be sure to create filters that are specific enough to return the entity that you intend to target.

For example, for the following relationship filter:

(${account.eraccountstatus}=0)

This filter performs an account search .

```
(eraccountstatus=0)
```

From this search result, internally a new filter is generated based on the account owner attribute. The internally generated filter will be in the following format:

```
(|(erglobalid=5423170071729299999)(erglobalid=2448879856996666666)
(erglobalid=3606444774688888888).....)
```

where the `erglobalid` values are taken from the matching account owners. An LDAP search with this filter is then executed.

If approximately more than 128 different account owners are found, then the internally generated filter exceeds the 4K LDAP filter size. This filter size is the limit set by IBM Security Directory Server 6.4 and the expression will not work.

System expressions

System expressions target domain objects that are based on generalized time values relative to the current system date.

The system expression syntax has relatively few elements.

System expressions consist of:

-   an attribute name
-   a relational operator (<= or >=)
-   a dollar sign ($) followed by a curly brace ({)
-    immediately followed by the `system.date` keywords
-   a plus or minus arithmetic operator (+/-) followed by a number in days
-   a right curly brace (}) to close the expression

For example:

```
(gmtattributename[<=|>=]${system.date [ + | - ] days})
```

System expressions resolve to a concrete LDAP filter that is understood by an LDAP directory server or the built-in IBM Security Identity Manager filter interpreter. For example, this filter targets accounts with passwords 90 days or older.

```
(erpswdlastchanged<=${system.date − 90})
```

That example can be used in an ACI for accounts that grants read and write access to the password attribute so that users can update their passwords. The same filter can also be used in a lifecycle rule that suspends accounts if the account password was not changed in the last 90 days. This particular filter expression resolves to the following concrete LDAP filter:

```
(erpswdlastchanged<=200912311200Z)
```

It is also possible and syntactically valid to express a range of dates as the criteria to match against domain objects. Embed more than one system expression in a composite filter as in the following example:

```
(&(erpswdlastchanged>=${system.date − 90})(!(erpswdlastchanged>
=${system.date − 30})))
```

The filter matches accounts with passwords that range from 90 to 30 days old. Other combinations and composite filters are useful, depending on how complex the filter must be and how many objects are targeted for a match.

Chapter 13. Policy join directives configuration

Provisioning policy *join directives* determine the provisioning parameter values that govern when multiple provisioning policies affect the same account. A join directive defines how to process an attribute when a conflict occurs between provisioning policies. Join directives applicable only to the selected attribute are displayed.

The entitlement target type also plays a role in how policy join directives resolve which entitlement is granted when conflicts arise between policies. When two or more policies grant similar entitlements, the more specific entitlement takes precedence. For example, one provisioning policy might include an entitlement defined to grant access to a type of service (that is, AIX® named AIX105). The second policy might include an entitlement defined to grant access to a specific instance of that service (that is, AIX). In this case, the more specific entitlement takes precedence.

IBM Security Identity Manager provides several types of join directives. The following table lists and describes each type.

Note: The Union and Intersection types are defined only on multivalued attributes.

Table 29. Join directives

Join Directive	Description
Union	Combines the attribute values and removes the redundancies. This join directive is the default parameter for multivalued attributes if no other join directive is specified.
Intersection	Only parameter values common to all policies.
Append	Appends the textual attribute value defined in one policy to the attribute value defined in another policy. The APPEND join type was designed for single-valued text attributes such as `comment on winlocal service`. When you join provisioning parameters by using the APPEND join type, all individual values are concatenated into a single string value. Concatenation provides with a user-defined delimiter between values. The delimiter can be defined (changed) in enrolepolicies.properties file, where the current line reads: `provisioning.policy.join.Textual.AppendSeparator=<<<>>>`
And	Specifies the mathematical AND used on a boolean string that represents a boolean value. TRUE & TRUE = TRUE TRUE & FALSE = FALSE FALSE & FALSE = FALSE
Or	Specifies the mathematical OR used on a boolean string that represents a boolean value. TRUE || TRUE = TRUE TRUE || FALSE = TRUE FALSE || FALSE = FALSE
Highest	Uses only the highest numeric attribute value from the conflicting policies.
Lowest	Uses only the lowest numeric attribute value from the conflicting policies.
Average	Averages the numeric attribute values from the conflicting policies and uses the average value.
Bitwise_Or	Specifies the mathematical Bitwise OR used on an attribute value that represents a bitstring.

Table 29. Join directives (continued)

Join Directive	Description
Bitwise_And	Specifies the mathematical Bitwise AND used on an attribute value that represents a bitstring.
Precedence_Sequence	Uses a user-defined ordering precedence to determine which attribute value to use.
Priority	Uses the priority of the policy to determine which attribute value to use. If the conflicting policies have the same priority, then the order in which these conflicting policies are evaluated is random. The evaluation is based on which policy the system retrieves first. For example, two policies have the same priority and define the same attribute with different values. If the attribute uses the 'Priority' join directive type, the attribute value returned by the policy varies based on the system retrieval.

The following table shows each type of service attribute, the corresponding join directive, and the default join directive.

Table 30. Service attributes

Service attribute type	Applicable join directive	Default join directive
Multivalued string or number attribute	UNION, INTERSECTION.PRIORITY, CUSTOM	UNION
Single-valued string	PRECEDENCE_SEQUENCE, PRIORITY, AND, OR, APPEND, BITWISE_AND, BITWISE_OR, HIGHEST, LOWEST, AVERAGE, CUSTOM	PRIORITY
Single-valued boolean string	AND, OR, PRIORITY, CUSTOM	OR
Single-valued integer	HIGHEST, LOWEST, AVERAGE, PRIORITY, PRECEDENCE_SEQUENCE, CUSTOM	HIGHEST
Singled-valued bitstring	BITWISE_AND, BITWISE_OR, PRIORITY, CUSTOM	BITWISE_OR

Note: Custom join directives can be defined by using Java. Administrators can use custom join directives to change the built-in join logic completely.

Customizing policy join behavior

You can customize join directive behavior for your provisioning policies for each attribute based on service type.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

IBM Security Identity Manager provides several types of join directives. You can extend existing join directive functions, or you can create your own.

You can define custom join directives by writing a custom Java class, adding it to the classpath of your application server. Provide the fully qualified Java class name in the policy configuration interface when you set a join directive for an attribute.

If you are extending or replacing one of the existing join directive classes in addition to the tasks above, you must add the custom property key and value to the `enrolepolicies.properties` file. For example, if you developed a new class such as `com.abc.TextualEx` to replace the existing class for textual joins, the registration line is similar to the following example:

```
provisioning.policy.join.Textual= com.abc.TextualEx
```

Procedure

1. From the navigation tree, select **Configure System** > **Configure Policy Join Behaviors**. The Policy Join Behavior table for configuring provisioning policy join directives is displayed as two panes in the window.
2. In the Policy Join Behavior window, click **Service Type** to select from a list of available services, such as `ITIM Service`.
3. Select one of the attributes for the type. The right pane displays the name, description, and applicable join directives of the selected attribute.
4. Click **Join Directive** in the right pane to configure provisioning policy precedence by selecting one of the listed join directives. The following values can apply, depending on the attribute you select:

   Union Specifies the attribute values and removes the redundancies. This join directive is the default if no other join directive is specified.

   Intersection
   Specifies only parameter values that are common to all policies.

   Priority
   Uses the priority of the policy to determine which attribute value to use. If the conflicting policies have the same priority, the first policy found by the system is used.

   OR Specifies the mathematical OR used on a boolean string that represents a boolean value. TRUE || TRUE = TRUE TRUE || FALSE = TRUE FALSE || FALSE = FALSE

   AND Specifies the mathematical AND used on a boolean string that represents a boolean value. TRUE & TRUE = TRUE TRUE & FALSE = FALSE FALSE & FALSE = FALSE

   Append
   Appends the textual attribute value defined in one policy to the attribute value defined in another policy.

   The APPEND join type is used on single-valued text attributes (such as `comment` on WinNT service).

   When joining provisioning parameters with the APPEND join type, all individual values are concatenated into a single string value with a user-defined delimiter between them. The delimiter can be defined (changed) in enrolepolicies.properties file, where the current line reads:

   ```
   provisioning.policy.join.Textual.AppendSeparator=<<<>>>
   ```

   Bitwise OR
   Specifies the mathematical Bitwise OR used on a bitstring.

   Bitwise AND
   Specifies the mathematical Bitwise AND used on a bitstring.

Highest

Uses the highest numeric attribute value from the conflicting policies.

Lowest

Uses the lowest numeric attribute value from the conflicting policies.

Average

Averages the numeric attribute values from the conflicting policies and uses the average value.

Precedence sequence

Uses a user-defined ordering precedence to determine which attribute value to use.

Custom

Defines a custom join directive with Java. Custom join directives provide administrators with the ability to completely change the built-in join logic. Enter the fully qualified Java class name of the custom join directive class you created for the attribute.

5. Click **Compliance Alert Rule** to configure a compliance alert rule that specifies when compliance alerts are sent. To configure a compliance alert rule, select one of the following options:

Numeric Order (higher value generates alert)

Select this option if you want to generate a compliance alert before sending a higher attribute value to the managed resource. Use this option if the attribute value was increased as a result of a provisioning policy evaluation. If the attribute value was decreased as a result of the evaluation, the attribute value is automatically sent to the managed resource. No alert is generated.

Numeric Order (lower value generates alert)

Select this option if you want to generate a compliance alert before sending a lower attribute value to the managed node. Use this option if the attribute value was decreased as a result of a provisioning policy evaluation. If the attribute value was increased as a result of the evaluation, the attribute value is automatically sent to the managed resource and no alert is generated.

Never generate alert

Select this option if you do not want to generate a compliance alert when a provisioning policy evaluation leads to a new value for an attribute. Because no compliance alert is generated, the new attribute value is automatically sent to the managed resource.

Always generate alert

Select this option if you want to generate a compliance alert when a provisioning policy evaluation leads to a new value for an attribute. The participant must accept the new attribute value before it is sent to the managed resource. This value is the default for attributes that have a single value.

Precedence sequence

Select this option if you want higher values in the list to be considered more privileged than lower values. When a provisioning policy evaluation leads to assignment of a higher attribute value, the attribute value is sent to the managed resource. No compliance alert is generated. If the attribute value is decreased as a result of the evaluation, a compliance alert is generated. Then, the attribute value is sent to the managed resource.

Note: When you select this option, you can select **Move Up**, **Move Down**, **Delete**, or **Add** to organize your precedence sequence.

6. Click **Save** to save the changes.

Account validation logic

Account validation logic provides information about a collection of validation rules that affect a joined set of parameter values after the policy join rules are applied.

Allow and deny parameter unions

An *allowing* set of parameter values is a union of the following elements:
- Mandatory constant parameter values (except null)
- Optional constant parameter values (except null)
- Non-negated regular expressions with optional enforcement
- Excluded null value

A *denying* set of parameter values is a union of the following elements:
- Non-negated regular expressions with excluded enforcement
- Excluded constant values (except null)
- Null value with optional, mandatory, or default enforcement

Note: Negated regular expressions, for example: `Match everything except a given word`, can be difficult to create manually. Optional and excluded parameters complement each other; use these types of parameters whenever possible.

Null parameter values

A null mandatory parameter value means that all values on the corresponding attribute of a new or existing account are disallowed except those values that any other valid values permit. When any attribute values on an existing account are denied by a null mandatory parameter, such values are automatically removed.

A null default or optional parameter value means that all values on the corresponding attribute of a new or existing account are disallowed, except those values that any other allowing values permit. Currently set values are not removed.

A null excluded parameter means that all attribute values are allowed on the corresponding attribute of a new or existing account except those values denied by any other denying parameter value.

Effects of governing parameter values on a single-valued attribute

Parameter values for a single-valued attribute can be qualified with mandatory or default enforcement only.

A mandatory parameter value means that the attribute must always have only the indicated value. Any change to the governing mandatory parameter value is automatically reflected on the attribute of the affected account. Removal of a mandatory parameter value from a governing entitlement can cause a value to be automatically changed on a corresponding attribute if no other mandatory parameter governs the same attribute.

A default parameter value is used in provisioning of new accounts. Attribute values governed by a default parameter can be changed at any time to any other value from the allowing parameter set. Removal of a default parameter value from a governing parameter does not cause a value to be removed from

a corresponding attribute unless a parameter join rule is used, through another mandatory parameter now governs the same attribute.

Effects of governing parameter values on a multivalued attribute

Parameter values for a multivalued attribute can be qualified with mandatory, default, optional, and excluded enforcement types.

A mandatory parameter value means that the corresponding attribute must always have this value. The addition of any new mandatory value (except null) causes this value to be added automatically to all existing accounts. The removal of an existing mandatory parameter value (except null) automatically causes removal of this value from the attribute unless another allowing parameter exists for the same value. Any change to a mandatory parameter value is equivalent to one remove and one add operation.

A non-null, default parameter value is effective only in provisioning of new accounts. Corresponding attribute values can be changed later to any other value from the allowing set. The addition of any new default parameter value (except null) has no effect on otherwise compliant attribute. The removal of a default parameter (except null) value does not cause removal of the value from the corresponding attribute unless another allowing (non-default) parameter for the same value exists.

Optional parameter values

Optional parameter values can be defined as a constant or a regular expression.

The addition of any new optional constant parameter value (except null) does not affect an otherwise compliant attribute. The removal of an optional constant parameter value (except null) can cause removal of the value from the corresponding attribute unless another allowing parameter permits the same value. Any change to an optional constant parameter value is equivalent to one remove and one add operation.

The addition of any new optional regular expression has no effect on an otherwise compliant attribute. The removal or change of an optional regular expression can cause the removal of attribute values on an otherwise compliant attribute unless another allowing parameter for the same value exists.

Excluded parameter values

Excluded parameter values can be defined as a constant or a regular expression. Parameter values with excluded enforcement are enforced only in the context of an implicit wildcard entitlement.

The addition of any new excluded constant parameter value can cause removal of the value from the corresponding attribute unless another allowing parameter exists for the same value. The removal of an excluded constant parameter value (except null) has no effect on an otherwise compliant attribute. Any change to an excluded constant parameter value is equivalent to one remove and one add operation.

The addition of any new excluded regular expression can cause the removal of attribute values on an otherwise compliant attribute unless another allowing parameter for the same value exists. Any removal or change of an excluded regular expression has no effect on an otherwise compliant attribute.

Allowed over denied precedence rule

If an attribute value is allowed and denied at the same time by the presence of conflicting parameter values, the allowing parameter value takes precedence over the denying parameter value.

`Implicit wildcard attribute entitlement`

> To help you create default grant-all policies easily, an *implicit wildcard* attribute entitlement is used. An implicit wildcard for an attribute exists if no single allowing parameter value defined on the attribute exists, and therefore all values are allowed minus any excluded (denying) parameter values. Removal of the last parameter for a given attribute reinstates the implicit wildcard.

Join directives examples

This topic provides examples that show how to use provisioning policy join directives.

The following example examines conflict resolution with policy priority, which is a default join directive for single-valued attributes. The erMaxStorage attribute on a Windows server is used to give a user limited storage space on the server.

Policy 1

> **Membership**
>> Managers
>
> **Priority**
>> 1
>
> **erMaxStorage**
>> 1000 (MB), enforcement: mandatory

Policy 2

> **Membership**
>> Employees
>
> **Priority**
>> 2
>
> **erMaxStorage**
>> 200 (MB), enforcement: mandatory

When a person belongs to both the Managers and Employees roles, the priority is used to resolve the conflict between the two erMaxStorage parameter values. A person who belongs to both groups would receive the erMaxStorage value 1000 (MB).

This next example examines conflict resolution with precedence sequence, which is a non-default join directive for a single-valued attribute.

Policy 1

> **Membership**
>> Managers
>
> **Priority**
>> 2
>
> **eraddialincallback**
>> 4, enforcement: mandatory

Policy 2

> **Membership**
>> Employees

Priority
> 1

eraddialincallback
> 2, enforcement: mandatory

custom join directive on eraddialincallback attribute
> Precedence sequence (most important first)
> - **4** User callback
> - **2** Fixed callback
> - **1** No callback

A person might belong to both the Managers and Employees roles. The precedence sequence is used to resolve the conflict between two parameter values, even though the priority on policy for Employees is higher. This person would get the `eraddialincallback` value 4 (user callback).

Join logic examples

This topic provides examples that show how to use provisioning policy join directives.

This section provides more examples of join logic.

Scenario 1

Multiple applicable entitlements might be joined. The parameter value is only allowed to take on the value that is specified by the second policy under these conditions:
- No parameter values are selected for an attribute in one policy, that is, all values are allowed.
- One allowed parameter value is entered for an attribute in another policy, that is, only the specified value is allowed.

Scenario 2

This example illustrates a priority-based provisioning policy join directive for a single-valued attribute. The following table identifies two provisioning policies for this scenario:

Table 31. Two provisioning policies

Policy	Description
Policy 1	Priority = 1 Attribute: `erdivision = divisionA, enforcement = DEFAULT`
Policy 2	Priority = 2 Attribute: `erdivision = divisionB, enforcement = MANDATORY`

Because Policy 1 has a higher priority, only Policy 1's definition for the `erdivision` attribute is used. Policy 2's value for the `erdivision` attribute is ignored. All other values besides divisionA are disallowed.

Scenario 3

This example illustrates a union-based provisioning policy join directive for a multivalued attribute. The following table identifies two provisioning policies for this scenario:

Table 32. Example provisioning policies

Policy	Description
Policy 1	Priority = 1 Attribute: `localgroup` = groupA, enforcement = DEFAULT
Policy 2	Priority = 2 Attribute: `localgroup` = groupB, enforcement = MANDATORY

Because the join directive is defined as UNION, the resulting policy uses the following definitions for the policies:

- During account creation, `localgroup` attribute is defined with both values groupA and groupB.
- During reconciliations, `localgroup` is defined as groupB if the attribute is undefined or incorrectly defined.

Chapter 14. Global policy enforcement

Global policy enforcement is the manner in which the IBM Security Identity Manager system globally allows or disallows accounts that violate provisioning policies.

When a policy enforcement action is global, the policy enforcement for any service is defined by the default configuration setting. You can specify one of the following policy enforcement actions to occur for an account that has a noncompliant attribute.

Mark Sets a mark on an account that has a noncompliant attribute.

Suspend
    Suspends an account that has a noncompliant attribute.

Correct
    Replaces a noncompliant attribute on an account with the correct attribute.

Alert Issues an alert for an account that has a noncompliant attribute.

Note: If a service has a specific policy enforcement setting, that setting is applied to the noncompliant accounts; the global enforcement setting does not apply to them.

Configuring a global enforcement policy

An administrator can create a global enforcement policy to resolve noncompliant accounts on the services.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

You can use the following options when configuring a global enforcement policy:
- **Mark**
- **Suspend**
- **Correct**
- **Alert**

Setting a mark on an account

An administrator can create a global enforcement policy and set a mark on an account that has a noncompliant attribute.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To set a mark on an account that has a noncompliant attribute, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Configure Global Policy Enforcement**.
2. On the Configure Global Policy Enforcement page, select **Mark** and then **Submit** in the Enforcement Action section.

   Note: Changing the global policy enforcement action for the system can cause a re-evaluation of account compliance and a modification of account data.
3. On the Confirmation page, select a time and date to schedule this operation.

   Note: When you select this option, you can select the calendar and clock icons to customize scheduled date and time.
   - Select **Immediate** and then **Submit** if you want to run the request immediately.

     Note: The current date and time are displayed.
   - Select **Effective date** and then **Submit** if you want to run the request at a date and time that you customized.
4. On the Success page, click **Close**.

Suspending an account

An administrator can create a global enforcement policy and suspend an account that has a noncompliant attribute.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

To suspend an account that has a noncompliant attribute, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Configure Global Policy Enforcement**.
2. On the Configure Global Policy Enforcement page, select **Suspend** and then **Submit** in the Enforcement Action section.

   Note: Changing the global policy enforcement action for the system can cause a re-evaluation of account compliance and a modification of account data.
3. On the Confirmation page, select a time and date to schedule this operation.

   Note: When you select this option, you can select the calendar and clock icons to customize scheduled date and time.
   - Select **Immediate** and then **Submit** if you want to run the request immediately.

Note: The current date and time are displayed.

- Select **Effective date** and then **Submit** if you want to run the request at a date and time that you customized.

4. On the Success page, click **Close**.

Replacing a noncompliant attribute with a compliant attribute

An administrator can create a global enforcement policy to resolve disallowed noncompliant accounts on the services. The global enforcement policy can deprovision accounts that are not granted by any applicable provisioning policy entitlement. Disallowed accounts can be exempt from being removed at the remote service if they meet the criteria of exemption accounts. Criteria are defined in the exemption handler.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

For more information about how to define exemption accounts in the exemption handler, see *Policy enforcement actions* in the Policy enforcement topic in the *IBM Security Identity Manager Administration Guide*.

Note: Your administrator can override the exemption handler that you defined or created.

About this task

To replace a noncompliant attribute on an account with a compliant attribute, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Configure Global Policy Enforcement**.
2. On the Configure Global Policy Enforcement page, select **Correct** and then **Submit** in the Enforcement Action section.

   Note: Changing the global policy enforcement action for the system can cause a re-evaluation of account compliance and a modification of account data. Furthermore, selecting **Correct** might cause account *deprovisioning* unless the account is exempt, which is account deletion, if an account is not granted by any provisioning policy entitlement.
3. On the Confirmation page, select a time and date to schedule this operation.

   Note: After you select this option, you can select the calendar and clock icons to customize scheduled date and time.

   - Select **Immediate** and then **Submit** if you want to run the request immediately.

     Note: The current date and time are displayed.

   - Select **Effective date** and then **Submit** if you want to run the request at a date and time that you customized.
4. On the Success page, click **Close**.

Creating an alert on an account

You can create an **Alert** to issue an alarm for an account that has a noncompliant attribute, and set up email notification of this alert.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

When you are working with a particular service with **Manage Services** in the navigation tree, you can set up a global policy enforcement alert for that service. Click the icon next to it in the list. Select **Configure Policy Enforcement**. Clicking **Use Global Enforcement Action: Alert** establishes a global policy alert for that service at the date and time you specify.

To set up an alert for an account that has a noncompliant attribute, complete these steps:

Procedure

1. From the navigation tree, select **Configure System** > **Configure Global Policy Enforcement**.
2. On the Configure Global Policy Enforcement page, select **Alert** in the Enforcement Action section.
3. Click **Continue**.
4. On the Configure Global Policy Enforcement page, select the General page to provide information and settings for the alert. Provide the participant information and time intervals. Specify the process types for which an alert is generated, and click **Submit**.

   Supply the following information:

   Alert name
   > Specify the name that identifies the alert.

   Send compliance alert to
   > Specify the participants who receive the compliance alert.

   Number of days to wait before escalating compliance alert
   > Specify the number of days before an alert is escalated.

   Escalate compliance alert to
   > Specify the participants who receive an escalated compliance alert.

   Number of days after which the system will take corrective action
   > Specify the number of days that the system waits until corrective action is taken.

   Process Types table
   > Specify the processes that generate a compliance alert.
   >
   > **Note:** If no process type is selected, the system automatically corrects a noncompliant account for that process type. The correction can modify or delete the account.

Generate Alert

Specify the process type for which an alert is generated. Select the check box of the process type for which you want to generate alerts.

Process Type

Specify the type of workflow process that generates a compliance alert.

5. On the Configure Global Policy Enforcement page, select the email page to provide text for the alert notification email. Alternatively, choose to use the default template. If you do not use the default template, type the subject line of the email notification. Provide the plain text body or the XHTML dynamic context.

6. Click **Submit**.

7. On the Confirmation page, select a time and date to schedule this operation.

   Note: After you select this option, you can select the calendar and clock icons to customize scheduled date and time.

   • Select **Immediate**, and then **Submit** if you want to run the request immediately.

   Note: The current date and time are displayed.

   • Select **Effective date** and then **Submit** if you want to run the request at a date and time that you customized.

8. On the Success page, click **Close**.

Chapter 15. Data import and export

IBM Security Identity Manager imports and exports data while maintaining data integrity.

Overview

Many enterprise applications, including Security Identity Manager, are often deployed in stages. New policies and business logic can be developed and tested in a test environment and then migrated to a production environment.

The import and export tasks are useful to migrate Security Identity Manager data items and dependent objects from a test environment to a production environment while maintaining data integrity.

You can use the import and export tasks to import previously exported objects from a Java archive (JAR) file. Importing of supported object types is limited to Security Identity Manager exported objects only.

There is a limitation on Java HTTPServletResponse for file downloads in displaying double-byte character file names. When naming your export JAR file, do try to use a conventional ASCII file name.

Data migration

Migrating data across Security Identity Manager servers consists of searching for and exporting configured objects from a source server. The migration imports the objects into a target server.

Data migration automates the extraction of commonly configured object types and their dependencies. Data migration is a mechanism to move working or staged configurations from a test environment to a production environment. The mechanism guarantees that the data is imported without loss of integrity. This information is for administrators who want to take advantage of the data migration feature in Security Identity Manager through the import and export tasks.

Exports

There are two types of exports: partial and full. Both types of exports produce a single downloadable JAR file. The file contains an XML file of serialized objects that is added to a list of completed exports.

Imports

Imports are initialized by an administrator on a target server after extracting objects (after generating an export JAR file) from a source server. Imports consist of these stages:
- JAR file upload
- Difference evaluation
- Conflict resolution
- Data commitment to the system

Policy enforcement

Importing provisioning policies and dynamic organizational roles might result in associating different people with new roles. Imported policies that have changes that require re-evaluation might result in the following policy enforcement tasks:

- Evaluating dynamic role changes and updating role memberships
- Finding provisioning policies associated with host selection policies
- Combining role memberships and provisioning policies with polices that are being imported
- Enforcing policies on all affected users through a new workflow process

Organizational charts

If there are differences in the organizational chart between the test (source) and target (production) systems, then the imported objects are treated as new objects.

To prevent the creation of duplicate objects when those objects exist in the production system, ensure that the organizational charts match in each system.

Object dependencies for data migration

To migrate data, you must ensure that you include all the dependencies of the migrated object.

A *dependency* is generally an individual object referenced by a parent or root object that is required on a target system to successfully import the parent. To protect the integrity of the data throughout the migration process, the import and export tasks automatically detect and include exported object dependencies.

Full exports compared to partial exports

Exporting everything through the use of **Export All** saves all of the data that is supported by **Export All** in the system. If you export individual items with a partial export, you might not actually export all of the dependencies needed for the object to function. A partial export saves only those dependencies that are needed to create the object that is saved. For example, you might export a provisioning policy that includes an automatic account creation function. The identity policy needed to create the user ID is not exported as a dependency of the provisioning policy. The identity policy is not required for the creation of the provisioning policy object. However, it might be required for the purpose that you intend for your provisioning policy. If that is the case, export and import the dependency as a separate object.

Policies

Identity policies and password policies are not exported when a provisioning policy is exported. You must explicitly export these policies as part of the export process.

An identity, password, or provisioning policy role and service objects are not exported by default. If you want to export these items, you must manually add them to the export list.

Services

If a service is exported, the service owner information is also exported. The dn is appropriately set if a person exists with the same name on the target system.

Role relationships

If a role that has a senior or junior role relationship is exported, then the relationship is also exported. The related role itself is not exported as a dependency.

If the dependent role exists on the target system, then the role relationship is created. Otherwise, it is not created. Role relationships are never deleted during import.

Exporting multiple objects

Exporting multiple objects over a period of time might result in saving variations of mutually shared dependencies that change over the course of the daily activity of the system. Keep the possibility of variations in mind as you plan your export strategy.

Dependencies and parent objects

Removal of a parent object is allowed. However, when a parent object is removed, the import and export tasks automatically remove all of its dependencies from the export list.

Table 33. Dependencies and parent objects

Parent object	Dependencies
Identity policy Lifecycle rule Lifecycle operation	Object profile
Identity policy Lifecycle rule Lifecycle operation Password policy Provisioning policy Service Service selection policy Workflow	Service profile
Provisioning policy Workflow	Organizational role
Adoption policy Identity policy Password policy Provisioning policy	Service
Lifecycle rule	Lifecycle operation

Performing a full export

Use this procedure to export all exportable object types and generate a Java archive (JAR) file that contains the export data.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

The JAR file that is generated by this task includes a full extract of all existing exportable objects, including their dependencies and references to containers.

To do a full export, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Export Data**. The Export Data page is displayed.
2. On the Export Data page, click **Export All**. The Export All page is displayed.
3. Optional: In the **Export Name** field, type a name to identify the export.
4. In the **Export to file (.jar)** field, type a file name for the export, and then click **Submit**. The Export Data page is displayed.
5. On the Export Data page, click **Refresh** to update the list of export items in the table.

Results

A full export JAR file is created and is displayed on the Export Data page.

What to do next

Perform additional export management tasks, such as downloading the JAR file, or click **Close**.

Performing a partial export

Use this procedure to selectively export object types and generate a Java archive (JAR) file that contains the export data.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you identified all the dependencies needed for the objects that you want to export.

About this task

The JAR file that is generated by this task includes a full extract of the exportable object types that you specify. The extract includes their dependencies and references to containers.

You can search for and select the following object types and include them in a partial export JAR file:

- Adoption policy
- Group
- Identity policy
- Lifecycle operation
- Lifecycle rule
- Organizational role
- Password policy
- Provisioning policy
- Service
- Service selection policy
- Workflow

To do a partial export, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Export Data**. The Export Data page is displayed.
2. On the Export Data page, click **Create**. The Create a Partial Export page is displayed.
3. To add objects to the export list, click **Add**. The Select Objects page is displayed.
4. To locate an object to export, complete these steps:
   a. In the **Name** field, type information about the object that you want to export.
   b. Select the object type that you want to search by from the **Object type** list, and then click **Search**. The objects that match your search criteria are displayed in the table.
5. Select the check box next to the object that you want to export, and then click **OK**. Selecting the check box at the top of this column selects all objects. The objects that you added are displayed on the Create a Partial Export page.
6. Verify the list of items that you want to export, and then click **Continue**. The Partial Export page is displayed.
7. Optional: In the **Export name** field, type a name to identify the export.
8. In the **Export to file (.jar)** field, type a file name for the export, and then click **Submit**. The Export Data page is displayed.
9. On the Export Data page, click **Refresh** to update the list of export items in the table.

Results

A partial export JAR file is created and is displayed on the Export Data page.

What to do next

Perform additional export management tasks, such as downloading the JAR file, or click **Close**.

Downloading the JAR file

Use this procedure to download a partial or full export Java archive (JAR) file to the local system.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you created an export file, including all dependencies and references to containers.

About this task

Export JAR files vary in size, depending on the type and number of objects exported. Each row in the list of completed exports specifies the type of export (partial or full) and the number of objects that were processed. Each row specifies a time stamp for when the export started and ended, the status of the export, and a link to the JAR file itself. The link to the JAR file allows you to download the file and save it to location on a local system.

To download a JAR file to a local system, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Export Data**. The Export Data page is displayed.
2. On the Export Data page, click the file name of the JAR file that you want to download. The File Download dialog is displayed.
3. On the File Download dialog, click **Save**. The **Save As** dialog is displayed.
4. Navigate to the location for saving the file, and then click **Save**.

Results

The JAR file is downloaded to the local system.

What to do next

Perform additional export management tasks, or click **Close**.

Deleting export records

Use this procedure to delete export records from the table.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

When the export record is deleted, all of its records are deleted from the database, including the Java archive (JAR) file. If you want to keep the JAR file, be sure to download it from the export record onto your local system before deleting the export record.

You cannot delete an export record if the export is still processing.

To delete export records from the table, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Export Data**. The Export Data page is displayed.
2. On the Export Data page, select the export that you want to delete, and then click **Delete**.

Results

The export record is removed from the table on the Export Data page.

What to do next

Perform additional export management tasks, or click **Close**.

Uploading the JAR file

Use this procedure to upload a partial or full export Java archive (JAR) file from the local system.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you saved an exported JAR file on your local system.

About this task

This task initializes the import of the JAR file through standard Java streams as the contents are inserted into a bulk data service database as a blob.

To upload a JAR file from a local system, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Import Data**. The Import Data page is displayed.
2. On the Import Data page, click **Upload File**. The Upload File page is displayed.
3. Optional: In the **Import name** field, type a name to identify the import, and then click **Browse**. The Choose file dialog is displayed.
4. On the Choose file dialog, navigate to the location of the file, select the file, and then click **Open**. The file name is displayed on the Import Data page.
5. Click **Submit** to upload the file. The Import Data page is displayed.

6. On the Import Data page, click **Refresh** to update the list of import items in the table.

Results

The JAR file is uploaded from the local system and is displayed on the Import Data page.

What to do next

Perform additional import management tasks, or click **Close**.

Resolving data conflicts

The import process evaluates differences between the data imported and the data in the target server and helps resolve conflicts between the two.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you imported a Java archive (JAR) file from your local system.

About this task

Difference evaluation generates a list of objects that are found in the import JAR file and in the target system. An administrator can use the list to resolve conflicts on an object-by-object basis. The administrator decides precedence over existing data or by overwriting existing data with the import data. Difference evaluation and conflict resolution are done for both partial and full export types.

Objects that exist in IBM Security Identity Manager at the time of the import that are selected in the conflicts summary to be overwritten are updated.

Objects in the uploaded JAR file that are not in Security Identity Manager at the time of the import are added.

To resolve conflicts between the data in an uploaded JAR file and the data in the server, complete these steps:

Procedure
1. From the navigation tree, click **Configure System** > **Import Data**. The Import Data page is displayed, and the **Status** column of the table indicates whether any conflicts are detected.
2. In the **Status** column on the table, click the **Conflicts Detected** link. The Evaluate Import File page is displayed.
3. On the Evaluate Import File page, select the check box next to the object that you want to import and override the existing object, and then click **Import**. Selecting the check box at the top of this column selects all objects. The Import Data page is displayed.
4. On the Import Data page, click **Refresh** to update status of the import in the table. The **Status** column indicates that the import is successful.

Results

The import process commits the data, re-establishes relationships between parent objects and their dependencies. The process places objects in their correct containers in the Security Identity Manager organizational chart.

If your session with the Security Identity Manager console is idle and times out while conflicts are being evaluated, or if you explicitly log off, then the import process status changes from `Processing` to `Failed - Conflicts not Resolved`. If this status change occurs, repeat this procedure so that the data is committed. Typically, user sessions are configured to be idle for up to 10 minutes before timing out.

What to do next

Do additional import management tasks, or click **Close**.

Deleting imports

Use this procedure to delete import records from the table.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

This procedure deletes the import record and the Java archive (JAR) file that was uploaded.

To delete import records from the table, complete these steps:

Procedure

1. From the navigation tree, click **Configure System** > **Import Data**. The Import Data page is displayed.
2. On the Import Data page, select the import that you want to delete, and then click **Delete**.

Results

The import record is removed from the table on the Import Data page.

What to do next

Perform additional import management tasks, or click **Close**.

Making import and export JAR files portable

For import and export Java archive (JAR) files to be portable between two machines, certain configuration settings must be the same in both systems.

About this task

To ensure that the import and export JAR files are portable between two systems, verify that these configuration settings are the same in both systems:

- Keystore file
- Keystore password
- Hash algorithm

Chapter 16. Configuration of IBM Cognos reporting components

After you install the prerequisites for the IBM Security Identity Manager Cognos Business Intelligence server, configure the Framework Manager, and create a content store database. Then, configure the web gateway and web server.

During the database configuration process, ensure that you complete the points in the following note.

Note:
- IBM Security Identity Manager Cognos reports do not support Microsoft SQL Server database. Use DB2 database or Oracle database instead.
- Set the JAVA_HOME environment variable to point to the JVM used by the application server.
- You must use the enterprise database as IBM Cognos content store.
- Delete the existing data source and create a new data source to enable an option of generating DDL during creation of the content store database. For information about data source creation, see "Creating a data source" on page 224.

The following table describes the configuration process.

Table 34. Configure IBM Cognos reporting components

Task	For more information
Configure Framework Manager.	1. Access the IBM Cognos Business Intelligence documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp. 2. Search for **Configuring Framework Manager on a 64-bit computer**.
Create a content store in the database.	1. Access the IBM Cognos Business Intelligence documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp. 2. Search for **Start IBM Cognos Configuration** and complete the steps as per your operating system. 3. Search for **Create a content store database**.
Configure the web gateway.	1. Access the IBM Cognos Business Intelligence documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp. 2. Search for **Configure the gateway**.
Configure your web server.	1. Access the IBM Cognos Business Intelligence documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp. 2. Search for **Configure your web server**.

Setting report server execution mode

You must have a report server execution mode that is set to 32-bit mode for the report packages that do not use dynamic query mode.

Procedure

1. Start IBM Cognos® Configuration.
2. In the **Explorer** panel, click **Environment**.
3. Click the **Value** box for **Report server execution mode**.
4. Select **32-bit**.
5. From the **File** menu, click **Save**.

What to do next

Restart the IBM Cognos service. Complete the following steps:

1. Access the IBM Cognos Business Intelligence documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp.
2. Search for **Restarting the IBM Cognos service to apply configuration settings**.

Setting environment variables

You must set the database environment variables for a user before you start the IBM® Cognos® processes.

Procedure

1. Access the IBM Cognos Business Intelligence documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp.
2. Search for **Database environment variables**.

What to do next

Start the Cognos service from IBM Cognos Configuration to host the IBM Cognos portal. Complete the following steps:

1. Access the IBM Cognos Business Intelligence documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp.
2. Search for **Starting or stopping the Cognos service**.

Creating a data source

To work with the IBM Security Identity Manager Cognos reports, you must create a data source.

Before you begin

- You must use the data source name as ISIM.
- If you are working with DB2 database client, copy the file db2cli.dll from the DB2 client installation directory to the *<IBM Cognos installation directory>*/bin folder.
- The data source must be pointed to the IBM Security Identity Manager enterprise database. For example, IBM DB2. After the data synchronization, the data is in the IBM Security Identity Manager enterprise database.

Procedure

1. Access the IBM Cognos Business Intelligence documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp.
2. Search for **Creating a data source** and complete the steps.

What to do next

Note: Corrupted attribute names are displayed in the reports for `Arabic`, `Chinese`, `Hebrew`, `Japanese`, `Korean`, and `Russian` languages. Double-byte character set (DBCS) characters appear to be corrupted in the reports. Edit the data source so that the data flow is in Unicode format. Complete the following steps:

1. On the Work with Reports page, click **Launch** > **IBM Cognos Administration**.
2. Click **Configuration** to open the data source connection.
3. Click **ISIM**.
4. Under the **Actions** column, click **Set properties-ISIM**.
5. On the **Set properties-ISIM** window, click **Connection**.
6. In the **Connection String** field, click the pencil symbol to edit the connection string.
7. In the **Collation Sequence** field, type `@UNICODE`.
8. Click **OK**.
9. Run the report to verify that the text is no longer corrupted.

Chapter 17. Identity feed management

As administrator, you need to take a number of initial steps to take employee data from one or more human resources repositories. You use the data to populate the IBM Security Identity Manager registry with an equivalent set of users.

Overview

An *identity* is the subset of profile data that uniquely represents a person in one or more repositories, and additional information related to the person. For example, an identity might be represented by unique combination of the first, last, full name, and employee number of a person. The data might also contain additional information such as phone numbers, manager, and email address. A data source can be a customer's user repository or a file, a directory, or a custom source.

Use IBM Security Identity Manager to add a number of users to the system by reading a data source, such as a user repository, directory, file, or custom source. The process of adding users based on a user data repository is called an *identity feed*, or *HR feed*.

Reconciliation for an identity feed is the process of synchronizing the data between the data source and IBM Security Identity Manager. The initial reconciliation populates IBM Security Identity Manager with new users, including their profile data. A subsequent reconciliation both creates new users and also updates the user profile of any existing users that are found.

You can use several source formats to load identity records into the IBM Security Identity Manager user registry.

You need to anticipate the effect of missing information in the user record. For example, the record that you feed into IBM Security Identity Manager might not have an email address for the user. The user does not receive a password for a new account in an email and must call the help desk, or contact the manager.

Common sources for identity feeds

IBM Security Identity Manager supplies the following service types to handle many of the most common sources for identity feeds:

- Comma-Separated Value (CSV) identity feed
- DSML identity feed
- AD OrganizationalPerson identity feed (Microsoft Windows Active Directory)
- INetOrgPerson (LDAP) identity feed
- IDI data feed

You can populate initial content and subsequent changes to the content of the people registry from these sources:

Comma-Separated Value (CSV) file
> Use a comma-separated value (CSV) file. A CSV file contains a set of records separated by a carriage return/line (CR/LF) feed pair. Each record contains a set of fields separated by a comma. You can use a global identity policy to select the schema attributes that create a user ID.

227

Directory Services Markup Language (DSML) v1 file

Use a DSML v1 file to populate the people registry. A DSML file represents directory structural information in an XML file format. If you run the identity feed more than one time, duplicate people are modified according to the newest file. A global identity policy does not apply to a DSML file.

Windows Server Active Directory

From Windows Server Active Directory, importing only the information found in the `inetOrgPerson` schema portion of a Windows Server Active Directory user. You can use a global identity policy to select the schema attributes that create a user ID. The identity feed process uses all user objects in the specified base.

INetOrgPerson identity feed

Use an LDAP directory server. The data uses the objectclass implied by the person profile name specified in the service definition. You can use a global identity policy to select the schema attributes that create a user ID. The identity feed process ignores records that do not have the specified objectclass.

Custom identity sources

Use custom identity sources to populate initial content and subsequent changes to the content of the people registry. Depending on the identity source, you might use a global identity policy to select the schema attributes that create a user ID.

For example, use an IBM Security Directory Integrator identity feed to obtain more flexibility than a standard data feed provides. Additional capabilities include:

- Working with a subset of data, such as filtering users in a specified department
- Enabling additional attribute mapping beyond the standard mapping
- Enabling data lookups, such as the manager of an employee, obtained from another data source
- Changing detection on the data source
- Using databases and human resource systems such as DB2 Universal Database™ and SAP
- Controlling attributes; for example, updating status such as suspending a person
- Deleting identity records
- Initiating changes with IBM Security Directory Integrator, instead of using IBM Security Identity Manager reconciliations

For more information about providing customized identity feeds, see the information about IBM Security Directory Integrator integration in the IBM Security Identity Manager `extensions` directory.

Enabling workflow for identity feeds

Regardless of the method used, the IBM Security Identity Manager Server can be configured to call the workflow engine for identity feed records. Enabling the workflow engine results in enforcement of all applicable provisioning policies for incoming identities. The configuration results in slower feed performance. Persons are automatically enrolled in any applicable dynamic roles even if the workflow engine is not enabled for an identity feed. For initial loads, consider importing

identities into the system and then enabling applicable provisioning policies to improve identity feed performance.

Comma-Separated Value (CSV) identity feed

The Comma-Separated Value (CSV) identity feed provides capability for reading comma-separated value (CSV) file to add users to IBM Security Identity Manager.

CSV service type

This identity feed service type parses identity feeds with CSV file formats that comply with RFC 4180 grammar. The IBM Security Identity Manager parser has the following RFC enhancements:

- Trims leading and trailing white space from unquoted text in a field. In contrast, RFC 4180 regards all space to be significant, whether inside or outside of quotation mark delimiters.
- Allows quoted and unquoted text to be in the same field. In contrast, RFC 4180 does not allow both text types in the same field.
- Does not enforce the RFC 4180 restriction that all records have the same number of fields. However, the code that calls the CSV parser reports an error if a record has more fields than the CSV header has.
- Allows record termination to use carriage return (CR) or to use carriage return/line feed (CR/LF) to be compatible with both UNIX and DOS base files. In contrast, RFC 4180 terminates all records with carriage return/line feed (CR/LF).

Services that use CSV files

IBM Security Identity Manager has the following types of services that use CSV files as input:

- CSV identity feed
- Custom services that use the Manual Service Provider type. These custom services use a CSV file format for the reconciliation upload file. This service type can be used for both identity and account feeds.

  By default, all accounts defined in a CSV file for reconciliation of a manual service are marked as active in Security Identity Manager. To suspend a person or account using a manual service reconciliation, add the `erpersonstatus` or the `eraccountstatus` attribute to the CSV file (depending on whether the feed is for identities or accounts). A value of `0` (zero) indicates active. A value of `1` indicates inactive.

- Custom services that use the Directory Integrator Adapter Provider type that use the IBM Security Directory Integrator CSV connector. This service type can be used for both identity and account feeds.

CSV file format

A CSV file contains a set of records separated by a carriage return/line feed (CR/LF) pair (\r\n), or by a line feed (LF) character. Each record contains a set of fields separated by a comma. If the field contains either a comma or a CR/LF, the comma must be escaped with double quotation marks as the delimiter. The first record in the CSV source file defines the attributes provided in each of the following records. For example:

```
uid,sn,cn,givenname,mail,initials,employeenumber,erroles
```

The sn and cn attributes are required by the object classes used by IBM Security Identity Manager to represent a person. The identity feed process uses all objects in the file. The CSV file cannot contain binary attributes.

You might use a multi-valued attribute to specify a user who has membership in multiple groups. Groups might include Service Owner, Windows Local Management (a self-defined group), and Manager. If you include multi-valued attributes, they must be represented by using multiple columns with the same attribute name.

To specify multi-valued attributes, repeat the column the required number of times. For example:

```
cn, erroles, erroles, erroles, sn
cn1,role1, role2, role3, sn1
cn2,rolea,,,sn2
```

The record that you feed into IBM Security Identity Manager might not have an email address for the user. That user does not receive a notification email that contains the password for a new account, and must call the help desk or contact the manager.

CSV connector for IBM Security Directory Integrator

Information about the CSV connector for IBM Security Directory Integrator is available in the following product directory:

ISIM_HOME/extensions/*versionNumber*/examples/idi_integration/HRFeedCSV/ITDIFeedExpress

(For example, /opt/IBM/isim/extensions/6.0/examples/idi_integration/ HRFeedCSV/ITDIFeedExpress)

UTF-8 encoding in an identity feed file

Your identity feed file must be in UTF-8 format. You must use an editor that supports UTF-8 encoding.

- Windows

  The following are UTF-8 capable: Microsoft Word 97 or later, or the Notepad editor that is included with the Windows 2003 Server or Windows XP operating systems.

  To save a file in UTF-8 format using Notepad, click **File** > **Save As**. Then, expand the list of choices for the **Encoding** field and select UTF-8.

- Linux

  The Vim text editor (a version of the classic vi editor) is UTF-8 capable. To work with files in UTF-8 format using the Vim text editor, specify the following:

  ```
  :set encoding=utf-8
  :set guifont=-misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
  ```

  If your version of UNIX does not include this text editor, access this Web site:

  http://www.vim.org

Note: For the 7-bit ASCII code subset, the UTF-8 encoded Unicode format is identical to 7-bit ASCII format. For input files that contain 7-bit ASCII (ASCII character values between hex 20 to hex 7e), you can use a normal text editor to create the file. For files containing any other character values (including extended European characters), you must save the file in UTF-8 format.

For an exact list of the 7-bit ASCII characters as supported by UTF-8, access this Web site and click the **Basic Latin** link in the first column:

http://www.unicode.org/charts

Directory Services Markup Language (DSML) identity feed

The Directory Services Markup Language (DSML) identity feed provides capability for reading a DSML file to add users to IBM Security Identity Manager.

DSML service type

The IBM Security Identity Manager Server allows for integration of various human resource (HR) type data feeds. You can add large numbers of individuals to the IBM Security Identity Manager Server without manually adding each individual. An identity record in HR data becomes an instance of a person object in IBM Security Identity Manager. One type of HR type data feed is the DSML Identity Feed service. The service can receive the information in one of two ways: a reconciliation or an unsolicited event notification through an event notification program.

The mechanisms that handle HR data in IBM Security Identity Manager requires that the HR data be in an XML format. The format uses the standard schema defined by the Directory Services Markup Language (DSML version 1). See the DSML website at http://www.oasis-open.org for DSMLv1. When sending asynchronous notifications, an XML message format defined by the Directory Access Markup Language (DAML version 1) is used. DAML is an XML specification defined by IBM that allows specification of add, modify, and delete operations.

DSML file format

DSML is an XML format that describes directory information. A *DSML file* represents directory structure information in an XML file format. The DSML file must contain only valid attributes of the IBM Security Identity Manager profile. The identity feed process uses all objects in the file.

The `erPersonPassword` attribute is used in an identity feed only during a Person create process, not in a Person modify process. If the value of the `erPersonPassword` attribute is set, then the IBM Security Identity Manager account password is set to that value when the person and account are created. The following statement sets a value for the `erPersonPassword` attribute:

```
<attr name="erpersonpassword"><value>panther2</value></attr>
```

If you select a DSML file format for an identity feed, specify a DSML file similar to this one:

```
<entry dn="uid=sparker">
<objectclass><oc-value>inetOrgPerson</oc-value></objectclass>
<attr name="givenname"><value>Scott</value></attr>
<attr name="initials"><value>SVP</value></attr>
<attr name="sn"><value>Parker</value></attr>
<attr name="cn"><value>Scott Parker</value></attr>
<attr name="telephonenumber"><value>(919) 321-4666</value></attr>
<attr name="postaladdress"><value>222 E. First Street Durham, NC  27788</value></attr>
</entry>
```

UTF-8 encoding in an identity feed file

Your identity feed file must be in UTF-8 format. You must use an editor that supports UTF-8 encoding.

- Windows

  The following are UTF-8 capable: Microsoft Word 97 or later, or the Notepad editor that is included with the Windows 2003 Server or Windows XP operating systems.

  To save a file in UTF-8 format using Notepad, click **File** > **Save As**. Then, expand the list of choices for the **Encoding** field and select UTF-8.

- Linux

  The Vim text editor (a version of the classic vi editor) is UTF-8 capable. To work with files in UTF-8 format using the Vim text editor, specify the following:

  ```
  :set encoding=utf-8
  :set guifont=-misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
  ```

  If your version of UNIX does not include this text editor, access this Web site:

  http://www.vim.org

Note: For the 7-bit ASCII code subset, the UTF-8 encoded Unicode format is identical to 7-bit ASCII format. For input files that contain 7-bit ASCII (ASCII character values between hex 20 to hex 7e), you can use a normal text editor to create the file. For files containing any other character values (including extended European characters), you must save the file in UTF-8 format.

For an exact list of the 7-bit ASCII characters as supported by UTF-8, access this Web site and click the **Basic Latin** link in the first column:

http://www.unicode.org/charts

JavaScript code within DSML identity feeds

The human resources database can also provide changes to the IBM Security Identity Manager server proactively as changes are detected.

The IBM Security Identity Manager server comes with a Java Naming and Directory Interface (JNDI) Service Provider. The provider can be used as a programming interface to deliver the changes to the server. These changes are received by the server as an event notification of change. This feature is called event notification. When using an event notification program to import HR data, add, modify, and delete operations are available.

JNDI service provider for DAML

Before using the JNDI Service Provider for DAML, you need to understand both the JNDI interface specification and LDAP. The JNDI Service Provider uses both concepts. This section provides links to information that you need to understand about the JNDI interface and LDAP.

JNDI The Java Naming and Directory Interface for accessing Directory type information from a Java program. See the website for Sun Microsystem at http://java.sun.com/products/jndi/tutorial/ for a tutorial on the JNDI.

LDAP Lightweight Directory Access Protocol. Information about this protocol can be obtained from many sources, such as the OpenLDAP Foundation at http://www.openldap.org.

The Java libraries required to use JNDI and DAML/DSML are contained within the `lib` directory of the IBM Security Identity Manager server directory.

Event notifications of HR data

HR data can be sent to the IBM Security Identity Manager server from another program as a DAML/HTTPS message.

The DAML/HTTPS message is sent to the IBM Security Identity Manager server as an HTTPS Post request. The Java Naming and Directory Interface JNDI Service Provider for DAML/HTTPS is provided for this purpose.

Initializing the context

For all operations with the JNDI SP for DAML, the first step is to initialize the context. The context must be initialized with all of the protocol properties needed to communicate with the IBM Security Identity Manager Server.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you imported these packages:
- `import javax.naming.*;`
- `import javax.naming.directory.*;`
- `import java.util.*;`

About this task

To initialize the context, modify the following environment variables:

```
Hashtable env = new Hashtable();
env.put (Context.INITIAL_CONTEXT_FACTORY,
"com.ibm.daml.jndi.DAMLContextFactory");
env.put(Context.SECURITY_PRINCIPAL,serviceUserName);
env.put(Context.SECURITY_CREDENTIALS, servicePassword);
env.put("com.ibm.daml.jndi.DAMLContext.CA_CERT_DIR", certDirLocation);
env.put(Context.PROVIDER_URL,providerURL);
env.put("com.ibm.daml.jndi.DAMLContext.URL_TARGET_DN", serviceDN);

DirContext damlContext = new InitialDirContext (env);
```

Results

When the context is initialized, a bind request is sent to the Security Identity Manager Server. If the environment variables are not correct, a `NamingException` is thrown.

What to do next

After initialization, you can do these tasks:
- Add a person entry
- Modify a person entry
- Remove a person entry

Adding a person entry

The attributes for adding a person are the same as the attributes used in the file reconciliation method.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you have initialized the context.

About this task

The attributes for adding a person are the same as the attributes used in the file reconciliation method. The DN for the new person entry must include at a minimum a unique attribute used to identify the person, such as the uid. The JavaScript placement rule defined for the DSML Identity Feed service is used to determine the organizational unit to which the person entry is added. If organization information is not provided, the person is added to the root of the organization. (The DN is specified through the `createSubcontext` / `destroySubcontext` / `modifyAttributes` methods in the following example).

The `objectclass` attribute must be defined and must match the LDAP object class that is mapped to the person type you want to add. This class is typically `inetOrgPerson`, but other objectclasses can be used by defining them through the Entity Configuration feature in the IBM Security Identity Manager Server. Add the required objectclass as a new entity, with `"Entity Type"` = `"Person"`.

To add a person, complete these steps:

Procedure

1. Define the DN of the person you want to add.
2. Create an Attributes object to contain the list of Attribute objects for the new user.
3. Call `createSubContext` on the context.

Results

After creating the DN and the attributes for the person, a call to `createSubcontext` is made with the JNDI context.

Example

```
BasicAttributes ba = new BasicAttributes(true);
ba.put(new BasicAttribute("objectclass","inetorgperson"));
ba.put(new BasicAttribute("uid", uid));
ba.put(new BasicAttribute("cn", "JoeSmith"));
ba.put(new BasicAttribute("mail", uid + "@acme.com"));

damlContext.createSubcontext("uid="+ uid, ba);
```

What to do next

You can do these tasks:

- Add another person entry

- Modify the information of a person entry
- Remove a person entry

Modifying a person entry

To modify a person entity, you must create a list of modification items and then call `modifyAttributes` on the context.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you have initialized the context.

About this task

To modify the attribute values for a person (including adding new attributes, or deleting existing ones), complete these steps:

Procedure

1. Define the DN of the person you want to modify.
2. Create a list of `ModificationItems` containing the required changes.
3. Call `modifyAttributes` on the context.

Results

After defining the DN of the person, a call to `modifyAttributes` is made with the JNDI context.

Example

```
Vector mods = new Vector();
// Add a new attribute (or additional value ifit already exists)
mods.add(new ModificationItem(DirContext.ADD_ATTRIBUTE, new BasicAttribute("roomnumber", "102")));
// Modify an existing Attribute
mods.add(new ModificationItem(DirContext.REPLACE_ATTRIBUTE, new BasicAttribute("title","Consultant")));
// Modify an existing Attribute to a multi-valued value Attribute
newOuAt = new BasicAttribute("ou");
newOuAt.add("Research Department");
newOuAt.add("DevelopmentDivision");
mods.add(new ModificationItem(DirContext.REPLACE_ATTRIBUTE, newOuAt));
// Delete one existing attribute
mods.add(new ModificationItem(DirContext.REMOVE_ATTRIBUTE,new BasicAttribute("initials", null)));
String dn = "uid=" + uid;
damlContext.modifyAttributes(dn,(ModificationItem[])mods.toArray(new ModificationItem[mods.size()]));
```

What to do next

You can do these tasks:
- Add person entry
- Remove a person entry

Removing a person entry

To remove a person, define the DN for the person and then call `destroySubContext` on the context.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Ensure that you have initialized the context.

About this task

To remove a person entry, complete these steps:

Procedure

1. Define the DN of the person you want to remove.
2. Call `destroySubContext` on the context.

Results

After defining the DN of the person, a call to `destroySubContext` is made with the JNDI context.

Example

```
damlContext.destroySubcontext("uid=" + uid);
```

What to do next

You can do these tasks:
* Add person entry
* Modify the information of a person entry

Sample driver for event notifications of HR data

This Java test program and sample compiler add 100 person entries to the IBM organization.

Purpose

This program assumes that a tenant with the short name of `ibm` exists, containing an organization that is named IBM Security Identity Manager. In the organization, there is a DSML Identity Feed service with the following attributes:
* Service name - dsmltest
* UID - dsml
* Password - dsml

This information is all specified in the `serviceDN`, `serviceUID`, and `servicePassword` lines in the following sample program.

The location of IBM Security Identity Manager Server is specified in the `providerURL` line.

Sample program

This sample program does not use a client certificate (it is not using two-way SSL authentication). A copy of the CA certificate for the server certificate that is installed in IBM Security Identity Manager Server must exist in the directory `\certificates` (cerDirLocation line).

```java
// TestDSML.java
import java.io.*;
import java.util.*;
import javax.naming.*;
import javax.naming.directory.*;

public class TestDSML {
  // Service DN.This is constructed of four parts:
  // "erservicename=dsmltest" specifies the name of the Service
  // "ou=itim" is the Organization
  // "ou=ibm" is the Tenant
  // "dc=com" is the base of the LDAP tree for IBM Security Identity Manager.
 static final String DEFAULT_SERVICEDN =
     "erservicename=dsmltest, ou=itim, ou=ibm, dc=com";
 static final String DEFAULT_HOST =
     "localhost:4443";

public static void main(String arg[]) {
  // number of people to process
  int noOfPeople = Integer.getInteger("count", 100).intValue();
  // required operation ("add", "del", "mod")
  String op = System.getProperty("op", "add").toLowerCase();

  String certDirLocation = "\\certificates";    //where to get the CA certificates
  // URL to use.
  // Use "/enrole/unsolicited_notification" to specify the Unsolicited Notification Servlet,
  // which is the servlet used for DSML requests -
  String host = System.getProperty("host", DEFAULT_HOST);
  String providerURL = "https:// " + host + "/enrole/unsolicited_notification";
  // Target DN
  String serviceDN = System.getProperty("servicedn", DEFAULT_SERVICEDN);

  String serviceUID = "dsml";      // user id defined for the service
  String servicePassword = "dsml";   // password defined for the services

  // create and fill the environment table
  Hashtable env = new Hashtable();
  env.put (Context.INITIAL_CONTEXT_FACTORY,
    "com.ibm.daml.jndi.DAMLContextFactory");
  env.put(Context.SECURITY_PRINCIPAL, serviceUID);
  env.put(Context.SECURITY_CREDENTIALS, servicePassword);
  env.put("com.ibm.daml.jndi.DAMLContext.CA_CERT_DIR", certDirLocation);
  env.put(Context.PROVIDER_URL, providerURL);
  env.put("com.ibm.daml.jndi.DAMLContext.URL_TARGET_DN", serviceDN);

  DirContext damlContext = null;
  try {
    // generate connection request
    damlContext = new InitialDirContext (env);
  }
  catch (NamingException e) {
    System.out.println("Error connecting to server at \"" + providerURL + "\": " + e.getMessage());
    return;
  }
  for (int i = 1; i&lt;=noOfPeople; i++) {
    String sn = "smith" + i;
    String uid = "jsmith" + i;
    String dn = "uid=" + uid;

    try {
     if (op.startsWith("add")) {
      BasicAttributes ba = new BasicAttributes(true);
      ba.put(new BasicAttribute("objectclass","inetorgperson"));
      ba.put(new BasicAttribute("uid", uid));
      ba.put(new BasicAttribute("cn", "Joe Smith"));
      ba.put(new BasicAttribute("mail", uid + "@acme.com"));
      ba.put(new BasicAttribute("sn"));

     damlContext.createSubcontext(dn, ba);
     }
     else if (op.startsWith("del")) {
      damlContext.destroySubcontext(dn);
     }
     else if (op.startsWith("mod")) {
      Vector mods = new Vector();
      // Add a new attribute (or extra value if it already exists)
      mods.add(new ModificationItem(DirContext.ADD_ATTRIBUTE, new BasicAttribute("roomnumber", "102")));
      // Modify an existing Attribute
      mods.add(new ModificationItem(DirContext.REPLACE_ATTRIBUTE, new BasicAttribute("title", "Consultant")));
      // Modify an existing Attribute to a multi-valued value
```

```
        Attribute newOuAt = new BasicAttribute("ou");
        newOuAt.add("Research Department");
        newOuAt.add("Development Division");
        mods.add(new ModificationItem(DirContext.REPLACE_ATTRIBUTE, newOuAt));
        // Delete one existing attribute
        mods.add(new ModificationItem(DirContext.REMOVE_ATTRIBUTE, new BasicAttribute("initials", null)));

        damlContext.modifyAttributes(dn, (ModificationItem[])mods.toArray(new ModificationItem[mods.size()]));
      }
    }
    catch (Exception e)  {
      System.out.println("Error, DN \"" + dn + "\": " + e.getMessage());
      e.printStackTrace();
      }
    }
  }
}
```

Sample compiler

A sample Windows XP script to compile the preceding test program is:

```
@rem compileDsmlTest.cmd - compile DSML Test Program
setlocal
rem locationof the lib directory containing the jar files from the
rem IBM Security Identity Manager installation lib directory, as listed below
set LIB=C:\ITIM\lib
set APP=TestDSML

rem Library files from IBM Security Identity Manager lib directory -
set AGENTLIB=%LIB%\enroleagent.jar
set CLASSPATH=.;%AGENTLIB%;%LIB%\jlog.jar

javac -classpath %CLASSPATH% -d . %APP%.java
endlocal
```

Importing HR data with reconciliation

HR data can be imported into the IBM Security Identity Manager Server from a file written in DSML, with the DSML Identity Feed service provider.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

In a cluster environment, the DSML file is present on all cluster member machines at the same location. In a reconciliation, the DSML file can be found regardless of which cluster member initiates the reconciliation.

The DSML file must be present on the Security Identity Manager Server machine for a single server setup.

About this task

When you use the DSML Identity Feed Service to import HR data from a DSML file, only the add and modify person operations are done. The delete person operation is not available when importing identity record information from a DSML file.

Note: When processing identity record information from a DSML file, it is assumed that the data set reconciled does not represent the entire person population for the Security Identity Manager Server. Because of this assumption, the polling method can be used to add or modify persons, but not delete them. To delete persons, the event notification interface must be used.

To import the HR data with the DSML Identity Feed service type, complete these steps:

Procedure

1. Create an instance of the DSML Identity Feed service.
2. Configure the service to refer to a DSML file that contains the identity record data. Specify the full path name to the DSML file. Use the service test feature to verify that the file name is correct.
3. Reconcile the service.

Results

When reconciling the DSML Identity Feed service, the identity record entries are read from the DSML file. For each identity record entry, the objectclass is matched up to the appropriate person profile in IBM Security Identity Manager. If a match is made, the distinguished name (dn) is converted into a search filter. The search filter looks for an existing match to a person entry that exists in the organization that contains the service. If a single match is found, then the person entry is used as an update to the existing entry. If no match is found, the individual is added as a new person entry. Duplicate matches return an error and the entry is not added.

Example

These statements are a sample of a DSML entry for a person:

```
<entry dn="uid=jsmith">
   <objectclass>
         <oc-value>inetOrgPerson</oc-value>
   </objectclass>
   <attr name="sn"><value>smith</value></attr>
   <attr name="uid"><value>jsmith</value></attr>
   <attr name="mail"><value>jsmith@IBM.com</value></attr>
   <attr name="givenname"><value>John</value></attr>
   <attr name="cn"><value>John Smith</value></attr>
</entry>
```

What to do next

You can now add, modify, and delete identity information with the Security Identity Manager interface.

You can add more users, modify existing users with the DSML file, and deleting users.

DSML identity feed service form

The fields on the DSML identity feed service form specify information about the Directory Services Markup Language (DSML) identity feed. For example, you might select a service profile to import identity data with DSML. Complete the fields on the form to connect to the server where the service resides.

The following fields are available on the DSML identity feed service form:

Service name
Specify a name that helps you identify the service instance.

Description
Specify more information about the service instance.

User ID
>Specify the administrative user ID for the service instance.

Password
>Specify the administrative password for the service instance. If password authentication is used, enter a value. Otherwise, reconciliation later fails.

File name
>Specify the file name, including the path name, that contains the user information.
>
>**Note:** In cluster environments, the file must be stored at the same location on all cluster members.

Use workflow
>Select this check box to use workflow for this service instance and to determine whether to automatically create accounts for entries. This feature can be used for small incremental feeds, but not for importing large amounts of data.

Placement rule
>Specify a rule to be used for placing a user (person) in the organization tree. This rule is defined with a script. The context of the script is the identity information for the current user in the feed and the service that defines the feed itself.

Sample DSML file for reconciliation

Use this example as a model for creating the DSML file you want to use to import HR data with reconciliation.

Sample

The following DSML file is a complete sample XML for use in reconciliation:

```
<?xml version="1.0" encoding="UTF-8"?>
<dsml>

   <directory-entries>

   <entry dn="uid=janesmith">
      <objectclass>
         <oc-value>inetOrgPerson</oc-value>
      </objectclass>
      <attr name="ou"><value>Engineering</value></attr>
      <attr name="sn"><value>Smith </value></attr>
      <attr name="uid"><value>janesmith</value></attr>
      <attr name="mail"><value>j.smith@ibm.com</value></attr>
      <attr name="givenname"><value>Jane</value></attr>
      <attr name="cn"><value>Jane Smith</value></attr>
      <attr name="initials"><value>JS</value></attr>
      <attr name="employeenumber"><value>E_1974</value></attr>
      <attr name="title"><value>Research and Development</value></attr>
      <attr name="telephonenumber"><value>(888) 555-1614</value></attr>
      <attr name="mobile"><value>(888) 555-8216</value></attr>
      <attr name="homepostaladdress"><value>15440 Laguna Canyon Rd, Irvine, CA 92614</value></attr>
      <attr name="roomnumber"><value>G-114</value></attr>
      <attr name="homephone"><value>(888) 555-3222</value></attr>
      <attr name="pager"><value>(888) 555-7756</value></attr>
      <attr name="erAliases">
         <value>j.smith</value>
         <value>jane_smith</value>
         <value>JaneSmith</value>
      </attr>
      <attr name="erRoles">
         <value>Engineering</value>
         <value>Development</value>
      </attr>
   </entry>
   <entry dn="uid=johndoe">
```

```
                    <objectclass>
                        <oc-value>inetOrgPerson</oc-value>
                    </objectclass>
                    <attr name="ou"><value>Sales-West</value></attr>
                    <attr name="sn"><value>Doe</value></attr>
                    <attr name="uid"><value>johndoe</value></attr>
                    <attr name="mail"><value>j.doe@ibm.com</value></attr>
                    <attr name="givenname"><value>John</value></attr>
                    <attr name="cn"><value>JohnDoe</value></attr>
                    <attr name="initials"><value>JD</value></attr>
                    <attr name="employeenumber"><value>S_1308</value></attr>
                    <attr name="title"><value>Sales Engineer</value></attr>
                    <attr name="telephonenumber"><value>(888) 555-1620</value></attr>
                    <attr name="mobile"><value>(888) 555-8210</value></attr>
                    <attr name="homepostaladdress"><value>15440 Laguna Canyon Rd, Irvine, CA 92614</value></attr>
                    <attr name="roomnumber"><value>G-120</value></attr>
                    <attr name="homephone"><value>(888) 555-3228</value></attr>
                    <attr name="pager"><value>(888) 555-7750</value></attr>
                    <attr name="erAliases">
                        <value>j.doe</value>
                        <value>john_doe</value>
                        <value>JohnDoe</value>
                    </attr>
                    <attr name="erRoles">
                        <value>Sales</value>
                    </attr>
                </entry>

        </directory-entries>

        </dsml>copy from here to there
```

AD Organizational identity feed

AD Organizational identity feed provides capability for creating users based on user records from Windows Server Active Directory (AD).

This feed uses a directory resource as the source for the feed. Information from the AD `organizationalPerson` objectclass is mapped to the `inetOrgPerson` schema. This identity feed loads all user objects under a specified base.

AD Organizational service type

When you create a service instance for this identity feed, the following information is required:

- URL used to connect to the directory resource
- User ID and password to gain access to the resource
- Naming context, which is the search base in LDAP terminology, and defines where in the directory tree to begin the search
- Name attribute, which must be selected from the values that are provided

After creation, this service is set to reconcile a specific branch of the directory.

Customized attribute mapping

The **Attribute Mapping file name** option provides a way to customize the mapping of LDAP attributes to IBM Security Identity Manager attributes.

The format of the attribute mapping file is `feedAttrName=itimAttrName`. Lines that begin with a number sign (#) or semicolon (;) are interpreted as comments.

The attribute mapping file completely overrides the default mappings. All attributes that are needed from the feed source must be included in the mapping file.

These attributes must be included in the mapping file:

- Attributes that are specified as required in the person profile form
- Attributes that are specified as required in the LDAP schema for the target person profile

If an attribute from the feed source is not included in the attribute mapping file, the value is not set on the IBM Security Identity Manager attribute.

The following example shows that six attributes are mapped. All other LDAP attributes are ignored.

```
#feedAttrName=itimAttrName
cn=cn
sn=sn
title=title
telephonenumber=mobile
mail=mail
description=description
```

UTF-8 encoding in an identity feed file

Your identity feed file must be in UTF-8 format. You must use an editor that supports UTF-8 encoding.

- Windows

  The following are UTF-8 capable: Microsoft Word 97 or later, or the Notepad editor that is included with the Windows 2003 Server or Windows XP operating systems.

  To save a file in UTF-8 format using Notepad, click **File** > **Save As**. Then, expand the list of choices for the **Encoding** field and select UTF-8.

- Linux

  The Vim text editor (a version of the classic vi editor) is UTF-8 capable. To work with files in UTF-8 format using the Vim text editor, specify the following:

  ```
  :set encoding=utf-8
  :set guifont=-misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
  ```

  If your version of UNIX does not include this text editor, access this Web site:

  http://www.vim.org

Note: For the 7-bit ASCII code subset, the UTF-8 encoded Unicode format is identical to 7-bit ASCII format. For input files that contain 7-bit ASCII (ASCII character values between hex 20 to hex 7e), you can use a normal text editor to create the file. For files containing any other character values (including extended European characters), you must save the file in UTF-8 format.

For an exact list of the 7-bit ASCII characters as supported by UTF-8, access this Web site and click the **Basic Latin** link in the first column:

http://www.unicode.org/charts

inetOrgPerson identity feed

The inetOrgPerson identity feed supports LDAP directory server with the RFC2798 (inetOrgPerson LDAP objectclass).

This feed uses a directory resource as the source for the feed. This identity feed loads all `inetOrgPerson` objects under a specified base. Records that do not have `objectclass=inetOrgPerson` are ignored.

inetOrgPerson service type

When you create a service instance for this identity feed, the following information is required:
- URL used to connect to the directory resource
- User ID and password to gain access to the resource
- Naming context, which is the search base in LDAP terminology, and defines where in the directory tree to begin the search
- Name attribute, which must be selected from the values that are provided

After creation, this service is set to reconcile a specific branch of the directory.

Customized attribute mapping

The **Attribute Mapping file name** option provides a way to customize the mapping of LDAP attributes to IBM Security Identity Manager attributes.

The format of the attribute mapping file is `feedAttrName=itimAttrName`. Lines that begin with a number sign (#) or semicolon (;) are interpreted as comments.

The attribute mapping file completely overrides the default mappings. All attributes needed from the feed source must be included in the mapping file. Attributes specified as required in the person profile form or LDAP schema for the target person profile must be in the mapping file. If an attribute from the feed source is not included in the attribute mapping file, the value is not set on the IBM Security Identity Manager attribute.

The following example shows that six attributes are mapped. All other LDAP attributes are ignored.

```
#feedAttrName=itimAttrName
cn=cn
sn=sn
title=title
telephonenumber=mobile
mail=mail
description=description
```

UTF-8 encoding in an identity feed file

Your identity feed file must be in UTF-8 format. You must use an editor that supports UTF-8 encoding.
- Windows

  The following are UTF-8 capable: Microsoft Word 97 or later, or the Notepad editor that is included with the Windows 2003 Server or Windows XP operating systems.

  To save a file in UTF-8 format using Notepad, click **File** > **Save As**. Then, expand the list of choices for the **Encoding** field and select `UTF-8`.
- Linux

  The Vim text editor (a version of the classic vi editor) is UTF-8 capable. To work with files in UTF-8 format using the Vim text editor, specify the following:

```
:set encoding=utf-8
:set guifont=-misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
```
If your version of UNIX does not include this text editor, access this Web site:

http://www.vim.org

Note: For the 7-bit ASCII code subset, the UTF-8 encoded Unicode format is identical to 7-bit ASCII format. For input files that contain 7-bit ASCII (ASCII character values between hex 20 to hex 7e), you can use a normal text editor to create the file. For files containing any other character values (including extended European characters), you must save the file in UTF-8 format.

For an exact list of the 7-bit ASCII characters as supported by UTF-8, access this Web site and click the **Basic Latin** link in the first column:

http://www.unicode.org/charts

IBM Security Directory Integrator (IDI) data feed

The IBM Security Directory Integrator (IDI) identity feed is used to support data feeds from custom identity sources, and to provide greater flexibility over the standard data feeds.

The IDI data feed is provided for instances where the other HR feeds are not sufficient. Use an IDI data feed to define custom identity feeds.

Use of this data feed requires knowledge of IBM Security Directory Integrator (IDI).

This data feed is used to provide greater flexibility over the standard data feeds. Examples of this flexibility include:
- Ability to work with a subset of data, such as filtering users in a specified department
- Additional attribute mapping beyond the one-to-one mapping provided by the standard feeds
- Data lookups, such as to derive a supervisor or manager from another data source
- Change detection on the data source
- Databases and HR systems, such as DB2®, Oracle, PeopleSoft, and SAP
- Control over attributes, such as updating status or suspending a person
- Deletion of people
- Changes driven by IBM Security Directory Integrator instead of by IBM Security Identity Manager reconciliations (used for deletions, updates, and change detection)

UTF-8 encoding in an identity feed file

Your identity feed file must be in UTF-8 format. You must use an editor that supports UTF-8 encoding.
- Windows

  The following are UTF-8 capable: Microsoft Word 97 or later, or the Notepad editor that is included with the Windows 2003 Server or Windows XP operating systems.

To save a file in UTF-8 format using Notepad, click **File** > **Save As**. Then, expand the list of choices for the **Encoding** field and select `UTF-8`.

- Linux

  The Vim text editor (a version of the classic vi editor) is UTF-8 capable. To work with files in UTF-8 format using the Vim text editor, specify the following:

  ```
  :set encoding=utf-8
  :set guifont=-misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
  ```

  If your version of UNIX does not include this text editor, access this Web site:

  http://www.vim.org

Note: For the 7-bit ASCII code subset, the UTF-8 encoded Unicode format is identical to 7-bit ASCII format. For input files that contain 7-bit ASCII (ASCII character values between hex 20 to hex 7e), you can use a normal text editor to create the file. For files containing any other character values (including extended European characters), you must save the file in UTF-8 format.

For an exact list of the 7-bit ASCII characters as supported by UTF-8, access this Web site and click the **Basic Latin** link in the first column:

http://www.unicode.org/charts

Identity information with IBM Security Directory Integrator

You can use IBM Security Directory Integrator to import identity information into IBM Security Identity Manager and to manage accounts on external resources in the IBM Security Identity Manager data store. Identity data can come from a human resources repository or another source, such as a company-wide directory. An identity record in HR data becomes an instance of a person object in IBM Security Identity Manager. Integration with IBM Security Directory Integrator requires network connectivity with the IBM Security Identity Manager system and a new service type to manage data feeds.

Advantages of using IBM Security Directory Integrator include:

- Avoiding the need for custom programming to manipulate raw personal information data into a form that can be imported into IBM Security Identity Manager. IBM Security Directory Integrator can be used to parse data from a comma-separated file or a database and feed the result into IBM Security Identity Manager as personal information data or changes to that data. Previously, a Directory Services Markup Language (DSML) file or custom Java Naming and Directory Interface (JNDI) client was required.

- Managing identity data in which IBM Security Identity Manager can act as a DSMLv2 client to retrieve person data from IBM Security Directory Integrator in reconciliation by running searches against IBM Security Directory Integrator, which acts as a DSMLv2 server. IBM Security Identity Manager can also act as a DSMLv2 server, accepting requests from a DSMLv2 client such as IBM Security Directory Integrator, with the JNDI service provider.

  Note: DSMLv2 is deprecated in IBM Security Identity Manager Version 5.0 in favor of the remote method invocation (RMI)-based IDI adapter framework. DSMLv2 continues to be supported in this release.

- Providing advantages in account management. See additional documentation in the `extensions` directory.

See additional documentation provided by the IBM Security Directory Integrator product. For examples of customizing schemas and importing data in an identity

data feed, navigate to the *ISIM_HOME*/extensions/*versionNumber*/examples directory. *ISIM_HOME* is the Security Identity Manager installation directory. *versionNumber* is the Security Identity Manager version; for example, 6.0.

Bulk loading identity data: a scenario

A typical scenario for the use of IBM Security Directory Integrator might be an administrator who is interested in bulk loading identity data into IBM Security Identity Manager.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

An instance of Security Directory Integrator must be running.

About this task

This scenario includes the following high-level tasks:

Procedure

1. Setting up the Security Directory Integrator configuration, including a DSMLv2 event handler and an assembly line with a connector to the wanted data source.
2. Starting the Security Directory Integrator event handler.
3. Setting up a Security Identity Manager service to communicate with the Security Directory Integrator configuration.
4. Running the reconciliation to initiate the communication.

Results

These events occur after the reconciliation:

1. Security Identity Manager sends a search request message to Security Directory Integrator, which searches the enterprise data store for the identity data.
2. Security Directory Integrator sends the data back to Security Identity Manager, which processes the data. This processing includes evaluation of the position in the organization tree in which to place people and evaluation of role membership. Processing also includes evaluation of a supervisor relationship, possibly evaluation of provisioning policy, and insertion of data into the Security Identity Manager data store. Evaluation of the provisioning policy could result in account management actions.
3. The identity information is loaded into Security Identity Manager from the enterprise data store.

What to do next

You can now add, modify, and delete identity information with the Security Identity Manager interface.

For additional scenarios on the use of Security Directory Integrator, see the `extensions` directory for these descriptions:
- Identity feed with JNDI

- End user account management
- Account event notification

Identity feeds that retain group membership

Ensure that identity feeds retain a user's membership in both customized and default groups.

All default IBM Security Identity Manager groups initially have no members, except for the administrator group, which contains one user whose account is named `itim manager`. When you load the first identity records into IBM Security Identity Manager, some individuals might become members of the manager group.

Table 35. Group membership after initial identity feed

Group name	Membership
Administrator	1 member with an account named `itim manager`
Manager	Zero or more, depending on whether the initial identity feed has an identity record that indicates the user has a managed relationship.
Service owner	Zero
Help desk assistant	Zero

The first help desk assistant and first service owner is a user that the administrator explicitly adds to the group. Alternatively, a user automatically gains membership in the service owner group if you specify the user as owner of a service. If you specify the user as the manager of another user, a user automatically gains membership in the manager group.

A user who is a member of a customized group must also be a member of the default group of the same category. Otherwise, processing results are unpredictable.

If the incoming identity record for a user initially indicates membership in a customized group, Security Identity Manager includes the user as a member of both the customized group and the default group of the same category. Security Identity Manager interprets a subsequent identity feed that includes the same user as a modification of the existing Security Identity Manager user. If the subsequent identity feed specifies that the user has membership only in the customized group, and not also in the default group of the same category, the user is removed from membership in the default group. To avoid this problem, ensure that both initial and subsequent identity feeds specify that a user has membership in both a customized and the default group of the same category.

Map of inetOrgPerson to Windows Server Active Directory attributes

The IBM Security Identity Manager `inetOrgPerson` attributes map to Windows Server Active Directory attributes. The differences are shown in **boldface** type.

Table 36. Map of `inetOrgPerson` and Windows Server Active Directory `organizationalPerson` attributes

IBM Security Identity Manager `inetOrgPerson` attributes	Windows Server Active Directory `organizationalPerson` attributes
cn	cn

Table 36. Map of `inetOrgPerson` and Windows Server Active Directory `organizationalPerson` attributes (continued)

IBM Security Identity Manager `inetOrgPerson` attributes	Windows Server Active Directory `organizationalPerson` attributes
departmentNumber	**department**
description	**comment**
employeeNumber	**employeeID**
givenName	givenName
homePhone	homePhone
homePostalAddress	homePostalAddress
initials	initials
internationaliSDNNumber	internationallSDNNumber
jpegPhoto	**thumbnailPhoto**
l	l
mail	mail
manager	manager
mobile	mobile
o	o
ou	ou
pager	pager
physicalDeliveryOfficeName	physicalDeliveryOfficeName
postalAddress	postalAddress
postalCode	postalCode
postOfficeBox	postOfficeBox
preferredDeliveryMethod	preferredDeliveryMethod
registeredAddress	registeredAddress
secretary	**assistant**
seeAlso	seeAlso
sn	sn
st	st
street	**streetaddress**
telephoneNumber	telephoneNumber
teletexTerminalIdentifier	teletexTerminalIdentifier
telexNumber	telexNumber
title	title
uid	**< - intentionally blank - >**
userPassword	userPassword **Note:** Encryption by the directory server prevents IBM Security Identity Manager from using the value of this attribute.
x121Address	x121Address

User passwords provided by an identity feed

Encryption by the directory server prevents IBM Security Identity Manager from using the `userPassword` attribute in the `inetOrgPerson` schema to provide user password data in an `inetOrgPerson` identity feed from LDAP or a Windows Server Active Directory identity feed.

Other identity feeds that use CSV, DSML, or IBM Security Directory Integrator-based formats can provide a password for a new user. Given the identity feed value, IBM Security Identity Manager uses the `erPersonPassword` attribute to create a password for a new user's IBM Security Identity Manager account. The `erPersonPassword` attribute is used only to create a password for a new IBM Security Identity Manager user. If the user exists, the value of the `erPersonPassword` attribute cannot be used to change the IBM Security Identity Manager user's login password.

In any identity feed where the `erPersonPassword` is not provided, IBM Security Identity Manager generates a new password for a new use. The application sends the generated password by email to the new user. If the email address of the user is not populated, the user must contact the help desk to obtain a password. Depending your site requirements, the new user's password might also be sent to the user's manager.

The password value that IBM Security Directory Integrator provides must be encoded in base64 format.

These identity feed attributes provide a value in clear text that is the password for a new user:
- CSV column name: `erPersonPassword`
- DSML tag: `erPersonPassword`

Attributes in an identity feed that are not in a schema

You can include some attributes in an identity feed that are not contained in the identity feed object class (`organizationalPerson` for Windows Server Active Directory; `inetOrgPerson` for IBM Security Identity Manager).

For example, the `erRoles` attribute determines a user's membership in a IBM Security Identity Manager group. The `erRoles` attribute is not in either the `organizationalPerson` or the `inetOrgPerson` schema. Based on the value of the `erRoles` attribute in an initial identity feed, a user might become a member of a customized group. The user might also become a member of a default Help Desk Assistant group.

A repeated identify feed might not contain a value for an attribute that was previously specified for the user, for both `organizationalPerson` and `inetOrgPerson` schemas. The identity feed process deletes that attribute for the IBM Security Identity Manager user.

If the incoming identity record for a user initially indicates membership in a customized group, Security Identity Manager includes the user as a member of both the customized group and the default group of the same category. Security Identity Manager interprets a subsequent identity feed that includes the same user as a modification of the existing Security Identity Manager user. If the subsequent identity feed specifies that the user has membership only in the customized group,

and not also in the default group of the same category, the user is removed from membership in the default group. To avoid this problem, ensure that both initial and subsequent identity feeds specify that a user has membership in both a customized and the default group of the same category.

For the Windows Server Active Directory feed, this problem also occurs for any `inetOrgPerson` attribute that is not also contained in the `organizationalPerson` schema. For an `inetOrgPerson` identity feed, the problem occurs for any `inetOrgPerson` attribute that is not supported by the identity feed.

Supported formats and special processing of attributes

IBM Security Identity Manager provides special processing for manager and secretary attributes, and for the `erRoles` attribute.

Supported formats and special processing for manager and secretary attributes

The manager and secretary attributes refer to another person entry within IBM Security Identity Manager.

Note: The Windows Server Active Directory identity feed maps the Windows Server Active Directory assistant attribute to the secretary attribute.

Internally, IBM Security Identity Manager uses a special format for the Distinguished Name (DN) of person directory entries. The format is inconvenient and difficult to specify in the identity feed data. So the identity feed code allows these attributes to be specified in more useful formats. IBM Security Identity Manager supports three formats for the values:

- A search filter (containing an equal (=) operator, but not `erglobalid`) that is a comma-separated list of attribute=value pairs.
- A simple name (not containing an equal (=) operator), which is assumed to be the value of the naming attribute for the person object class (that is, `cn`).
- A full IBM Security Identity Manager DN (containing an equal (=) operator and `erglobalid`). The expression must exactly match the IBM Security Identity Manager LDAP DN of one of the currently defined person objects.

For the first two cases, IBM Security Identity Manager converts the value to an LDAP search filter. The process does a subtree search of the organization to find a unique matching person. If the search returns zero matches, or more than one match, then the value is considered invalid, and is removed from the list. A suitable warning message is written to the IBM Security Identity Manager log.

A potential issue can occur with both the manager and secretary attributes if they reference a person who is also defined in the same feed. In this case, it is possible that when the attribute value is processed as above, the person that it references is not yet been created. This issue can occur even if the manager or secretary person is defined earlier in the identity feed file. The cause is multithreaded and asynchronous processing done by IBM Security Identity Manager during an identity feed. This situation results in deleting the attribute from the person, because the attribute references an invalid person. A warning is written to the logs.

There are two solutions to this reference dependency issue. First, run the identity feed a second time, after all processing completes from the first run. This second feed is much faster, because only changed entries cause in any significant

processing during the feed. Alternatively, define these people (managers and secretaries) in a separate identity feed file. Run that identity feed first, then run the main feed after the first feed fully completes. This separate, first feed might also contain entries that reference managers that are defined in the same feed. You might need to run the separate, first feed twice, or split the feed again.

Asynchronous workflow activities to create or modify people might still be running, even after the identity feed status seems to be complete. In this case, you must wait for an additional interval of time after the first feed seems to be complete, before submitting the second feed.

Supported formats and special processing for erRoles attribute values

The erRoles attribute is used to specify the list of roles to which a person belongs. In IBM Security Identity Manager, groups are equivalent to roles that IBM Security Identity Manager, as an enterprise product, provides. IBM Security Identity Manager uses the erRoles attribute to specify the groups to which a user belongs. For example, specifying an identity feed attribute erRoles with a value of Help Desk Assistant causes the user to belong to the Help Desk Assistant group. The erRoles attribute can be multi-valued.

These formats are supported:
- A simple name (not containing an equals (=) operator), which is assumed to be the value of the erRoleName attribute. IBM Security Identity Manager does a subtree search to find a unique matching static role. The name is not valid if zero or more than one role is a match.
- A full IBM Security Identity Manager DN, which must exactly match the IBM Security Identity Manager LDAP DN of one of the currently defined static roles.

Any invalid value is removed from the value list. If this results in zero remaining values, the attribute is removed from the attribute list. A suitable warning message is written to the log.

Modifiable schema classes and attributes

You can modify some IBM Security Identity Manager schema classes and attributes.

You can create new classes with names that begin with the characters er, a prefix that previously was reserved for IBM Security Identity Manager schema classes and attributes.

The IBM Security Identity Manager schema classes and attributes that you can modify have a unique object identifier (OID) prefix. An OID is a string of numbers that identifies a unique class in an LDAP schema. The IBM Security Identity Manager schema classes and attributes that remain read-only have the following OID prefix:

1.3.6.1.4.1.6054.1.**1**

Person naming and organization placement

When the IBM Security Identity Manager Server imports HR data, the server creates Distinguished Names (DN) for each identity record. The server also places the person in a specific organizational unit based on the information provided.

To uniquely identify and place each individual, each entry (or person) must organize its data in a way that the IBM Security Identity Manager Server can recognize as individual pieces (attributes). The IBM Security Identity Manager Server must also be configured to recognize attributes that are passed. Recognition is done by matching the objectclass attribute against the defined person profiles. By default, the LDAP standard `inetOrgPerson` objectclass is expected.

Placement of the person

The IBM Security Identity Manager Server determines where to place in the organization chart. The server uses a placement rule defined in the DSML Identity Feed service.

A person might be defined as a member of the marketing department in the identity source. The placement rule instructs the server to place the person in the marketing department in the IBM Security Identity Manager organization chart. This rule is used for initial placement of persons during an add operation and for moving a person to a different location during a modify operation.

Note: Organization names returned by placement rules must be unique within the context of the service unless an organization path is used to specify an organization container. If an organization path is provided by the placement rule, the organization name must be unique within that organization container.

Placement rules are written with JavaScript that returns the organization path in a distinguished name (DN) format. This information is used to search for an organizational unit in which to place a person. This DN indicates the required organization path relative to the organization base. The syntax of this path can be represented with the following pseudo BNF notation:

```
orgDn ::= orgRdn | orgRdn "," orgDn
orgRdn ::= prefix '=' name
prefix ::= 'l' | 'o' | 'ou'
name ::= string
```

where `string` is the textural value, `l` is location, `o` is organization, and `ou` is the organizational unit, business partner organization, or Admin Domain.

Note: The prefixes noted here are the default values. If the customer uses a different schema, then these prefixes are the values mapped in entity configuration.

Example

To illustrate, examine the following organization chart:

```
IBM (organization)
   Marketing (organizational unit)
   Facilities (organizational unit)
      Irvine (location)
```

The path for the Marketing department is `ou=Marketing, o=IBM`. The path for the Irvine Facilities department is `l=Irvine, ou=Facilities, o=IBM`.

The JavaScript function returns a string in this format, but omits the organization. The attributes of the identity record from the identity source can be retrieved from the JavaScript code to construct the path. Because of the programming flexibility provided by JavaScript code, the information used from the identity source can be manipulated in several ways. Programming constructs such as switch statements can be used to map specific organization names to different paths in the server.

String manipulation can be used to tokenize or concatenate names to derive paths. For example, a string of `IBM/Facilities/Irvine` can be tokenized and reconstructed in DN format as `l=Irvine, ou=Facilities, o=IBM`.

The following example demonstrates one use of this scripting capability. The identity source for the Acme organization uses the attributes `div` for division, `bu` for business unit, and `dept` for department. The logical layout of the organization is as follows:

```
organization
   division
      business-unit
         department
```

In the IBM Security Identity Manager Server, this structure is mapped to organizations and organizational units and looks like this example:

```
organization
   organizational unit (division)
      organizational unit (business-unit)
         organizational unit (department)
```

The following JavaScript code can be used for the placement rule to make this conversion:

```
return "ou=" + entry.dept[o] + ",ou=" + entry.bu[o]  + ",ou=" + entry.dw[o];
```

Note: All identities in this feed are assumed to be within the Acme organization.

For an organization that uses a multi-valued `ou` attribute, the placement rule might be:

```
var ou =entry.ou;
var filt = '';
for (i = 0, i < ou.length, ++i)
{
  if (i==0)
      filt = ''ou='' + ou[i];
  }
else
    {
filt = filt + '',ou='' + ou[i];
    }
}
return filt;
```

The IBM Security Identity Manager Server evaluates this script when adding a person to place that person in the organization. During a modify request, this script is evaluated. If the value is different from the current placement of the person, the person is moved to the new location based on the returned path.

Creating an identity feed service

Create a service instance for an identity type, such as CSV or DSML.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Before you can create a service in IBM Security Identity Manager, you must create a service type. Alternatively, you can use one of the service types that was automatically created when the IBM Security Identity Manager Server was installed. You can create a service type by installing the adapter profile. You can also add new schema classes and attributes for the service to your LDAP directory. Before you can create a service for an adapter, the adapter must be installed, and the adapter profile must be created.

About this task

The service name and description that you provide for each service are displayed on the console. Therefore, it is important to provide values that make sense to your users and administrators.

To create an identity feed service instance, complete these steps:

Procedure

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, click **Create**. The Create a Service wizard is displayed.
3. On the Select the Type of Service page, select an identity feed service type, and then click **Next**.

   If the table contains multiple pages, you can:
   - Click the arrow to go to the next page.
   - Type the number of the page that you want to view and click **Go**.
4. On the Service Information page, specify the appropriate values for the service instance.
5. Click **Test Connection** to validate that the data in the fields is correct, and then click **Finish**.

Results

For the inetOrgPerson identity feed, a successful test connection message confirms that all required fields are filled and that the specified target can be reached. It does not guarantee that reconciliation of the LDAP resource is successful or produces the wanted results.

A message indicates that you successfully created the service instance for the specific identity feed service type.

What to do next

Schedule reconciliation, or run a reconciliation immediately with the task list associated with the service.

When the Select a Service page is displayed, click **Refresh** to refresh the **Services** table and display the new service instance.

Performing an immediate reconciliation on an identity feed service

Initiate a reconciliation activity immediately on an identity feed service. During a reconciliation, the IBM Security Identity Manager Server requests the identity record information from the specified file.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Set up a suitable identity feed service.

Procedure

To run a reconciliation now, complete these steps:
1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▶) next to the identity feed service, and then click **Reconcile Now**.

Results

A message indicates that you successfully submitted a reconciliation request to run immediately.

What to do next

To view the results of the reconciliation, click **View my request**, or click **Close**.

Creating a reconciliation schedule for an identity feed service

Schedule a reconciliation to run at a specific interval. During a reconciliation, the IBM Security Identity Manager Server requests the identity record information from the specified file.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Set up a suitable identity feed service.

Procedure

To create a reconciliation schedule for an identity feed service, complete these steps:

1. From the navigation tree, click **Manage Services**. The Select a Service page is displayed.
2. On the Select a Service page, complete these steps:
   a. Type information about the service in the **Search information** field.
   b. In the **Search by** field, specify whether to search against services or business units.
   c. Select a service type from the **Search type** list.
   d. Select a status from the **Status** list, and then click **Search**. A list of services that matches the search criteria is displayed.

      If the table contains multiple pages, you can:
      - Click the arrow to go to the next page.
      - Type the number of the page that you want to view and click **Go**.
3. In the **Services** table, click the icon (▸) next to the identity feed service, and then click **Set Up Reconciliation**. The Manage Schedules page is displayed.
4. On the Manage Schedules page, complete the following steps:
   a. Specify whether a policy evaluates the accounts that the reconciliation returns.
   b. Click **Create**. The Set Up Account Reconciliation notebook is displayed.
5. On the General page, type information about reconciliation schedule.
6. On the Schedule page, select a schedule interval for the reconciliation. The fields displayed depend on the scheduling option that you select.
7. Optional: On the Query page, specify an LDAP search filter for account attributes to include in a query. Select this option if you want to do a "supporting data only" reconciliation.
8. Click **OK** to save the new schedule and close the page.

Results

A message indicates that you successfully created a reconciliation schedule.

What to do next

Select another services task, or click **Close**. When the Select a Service page is displayed, click **Refresh** to refresh the **Services** table.

Chapter 18. IBM Security Identity Manager utilities

IBM Security Identity Manager provides utilities to configure database schema, LDAP, and commonly used system properties.

System configuration tool (runConfig)

To configure commonly used system properties, you might use the system configuration tool (runConfig) to change property files that contain IBM Security Identity Manager system settings. You might also change WebSphere Application Server settings for IBM Security Identity Manager.

For example, when you change a computer that provides the mail server at your site, the IP address for the mail server might change. Use the system configuration tool to specify the new address for the mail server.

For more information about this utility, see the *IBM Security Identity Manager Installation and Configuration Guide*.

runConfig command

The runConfig command starts the system configuration tool that IBM Security Identity Manager provides.

Command description

To manually start the system configuration tool, run this command:

ISIM_HOME/bin/**runConfig**

If the EJB User or System User user IDs or passwords are changed on the operating system, use an additional `install` argument to force the update on the WebSphere Application Server. If these user IDs or passwords are changed, run this command:

ISIM_HOME/bin/**runConfig install**

If you have problems with the system configuration tool, check the *ISIM_HOME*/install_logs/runConfig.stdout log file for more information. The system configuration requires several minutes to complete. It requires additional time if the `install` argument is used.

Database configuration tool (DBConfig)

To configure the IBM Security Identity Manager database, you can use the database configuration tool, (DBConfig).

The database configuration tool creates the database schema and default data that IBM Security Identity Manager requires. Run this tool *only* if the command failed to configure the database during installation. If the IBM Security Identity Manager database tables were previously configured, running the DBConfig command prompts the user. Then, the user can choose to drop all previously existing IBM Security Identity Manager tables, or to exit without configuring the database.

DBConfig command

The DBConfig command starts the database configuration tool that IBM Security Identity Manager provides.

Command description

To manually start the database configuration tool, run this command:

ISIM_HOME/bin/**DBConfig**

After you change a field value, click **Test** to ensure that the connection to the database is active. When the database test is successful, the **Test** button changes to **Continue**. After you click **Continue**, the database configuration requires several minutes to complete.

If you have problems with the database configuration tool, check the *ISIM_HOME*/install_logs/dbConfig.stdout log file for more information.

Directory server configuration tool (ldapConfig)

To configure the directory server for IBM Security Identity Manager, you can use the directory server configuration tool (ldapConfig).

Do not run the directory server configuration tool, unless the LDAP configuration fails during the ldapConfig installation process. The directory server configuration tool creates the LDAP schema and default data for IBM Security Identity Manager. Running the directory server configuration tool after the directory server is configured restores the default values that IBM Security Identity Manager uses. If you changed the value of any of these IBM Security Identity Manager attributes, the value is overwritten with the default value. For example, ldapConfig resets the password for the user ID named itim manager to the default password of "secret".

See *IBM Security Identity Manager Installation and Configuration Guide*.

ldapConfig command

The ldapConfig command starts the directory server configuration tool that IBM Security Identity Manager provides.

Command description

To manually start the directory server configuration tool, run this command:

ISIM_HOME/bin/**ldapConfig**

Click **Test** to ensure that the connection to the directory server can be established. When the test for a connection to the directory server is successful, the fields in the Identity Manager Directory Information section become active.

If you have problems with the directory server configuration tool, check the *ISIM_HOME*/install_logs/ldapConfig.stdout log file for more information. The directory server configuration requires several minutes to complete.

Chapter 19. High Availability and Disaster Recovery

Configure IBM Security Identity Manager for hot-standby applications.

IBM Security Identity Manager allows separate nodes of a cluster to function as hot-standby nodes. Typically the nodes are geographically separate for a disaster recovery application.

You also use hot-standby nodes for testing applications on the node without interfering with other nodes.

Implementation of this capability is discussed on the IBM developerWorks site at Setting up HA and DR environments for ITIM 5.x. You can also use the setup instructions with IBM Security Identity Manager Version 6 and Version 7.0 VA.

The `enroleStartup.properties` file contains properties to support this capability. See the **Reference** > **Supplemental property files** for enroleStartup.properties. Note the following changes for version 6.0.0.4:

- The `enroleStartup.properties` file can be modified starting in this release. It was not modifiable in previous releases.
- The following new and changed properties play a role in High Availability and Disaster Recovery applications:

```
enrole.startup.MessageListeners.attributes
enrole.appServer.standby
enrole.appServer.standby.inactiveMessageListeners
enrole.appServer.standby.inactiveStartupInitializer
```

Chapter 20. Integrating with IBM® Control Desk

This section introduces the Security Identity Manager integration for IBM® Control Desk offering and refers to instructions for installing and configuring this package.

Instructions for setting up the integration are provided in Chapter 19 of the Redbook Tivoli Integration Scenarios.

The Redbook is based on earlier versions of both products but the instructions have been tested on the current versions. Use the prerequisites in this section for integrations with IBM Security Identity Manager version 6.0.x. Note that the following terminology is different:

- IBM Security Identity Manager is referred to as Tivoli Identity Manager.
- IBM® Control Desk is referred to as Tivoli Service Request Manager.
- Some paths have a version embedded and should be changed to reflect the current version. For example, the `tim_51` directory would be `tim_60` for integration with 6.0.x versions.

Prerequisite software

This section describes the prerequisite software products for the IBM Security Identity Manager integration for IBM® Control Desk.

Before you install the Security Identity Manager integration for IBM® Control Desk, the following products must be installed and running on one of the specified operating systems:

- IBM Security Identity Manager Version 6.0 on Windows, AIX, HP-UX, or Solaris
- IBM SmartCloud Control Desk Version 7.5 on Windows, AIX, Linux
- IBM Maximo® Administration Machine with Base Services on Windows

The IBM® Control Desk product must be supported by a web application server and a database server. See the IBM SmartCloud® Control Desk Wiki and search `control desk system requirements` for a list of supported software.

Adapter attributes

This section describes the adapter attributes.

Attribute descriptions

The IBM Security Identity Manager Server communicates with the IBM® Control Desk service provider using attributes that are included in transmission packets that are sent over a network. The combination of attributes which are included in the packets, depends on the type of action that the Security Identity Manager Server requests from the IBM® Control Desk service provider.

Table 37 on page 262 contains a list of the attributes that are used by the IBM® Control Desk service provider, and gives a brief description and the data type for the value of the attribute.

Table 37. Attributes, descriptions, and corresponding data types

Attribute	Directory server attribute	Description	Data format
Userid	eruid	Specifies the user ID of the account.	String
Password	erpassword	Specifies the account password.	String
Status	eraccountstatus	Specifies the status of the account (ACTIVE, INACTIVE).	String
Type	ermaximousertype	Specifies the type of the Maximo user.	String
Defsite	ermaximodefsite	Specifies the default site of the account.	String
Storeroomsite	ermaximostoresite	Specifies the storeroom site of the account.	String
Querywithsite	ermaximoquerysite	Specifies whether or not to use the insert site as a display filter.	Boolean
Emailpswd	ermaximoemailpswd	Specifies whether or not to e-mail the password to the user on account creation.	Boolean
Sysuser	ermaximosysuser	Specifies whether or not the account is a system account.	Boolean
Screenreader	ermaximoscreen	Specifies whether or not the account requires a screen reader.	Boolean
Firstname	ermaximofirstname	Specifies the first name of the person supporting the user account.	String
Lastname	ermaximolastname	Specifies the last name of the person supporting the user account.	String
Phonenum	ermaximophone	Specifies the primary phone number for the person.	String
PhoneType	ermaixmophonetype	Specifies the type of the primary phone number for the person.	String
Email	ermaximoemail	Specifies the primary e-mail address for the person.	String
Memo	ermaximomemo	Specifies the memo for the person.	String
Addressline1	ermaximoaddress	Specifies the address of the person.	String
City	ermaximocity	Specifies the city of the person.	String
Stateprovince	ermaximostate	Specifies the state of the person.	String
Postalcode	ermaximozip	Specifies the zip of the person.	String
Country	ermaximocountry	Specifies the country of the person.	String
Groupname	ermaximogroupname	Specifies the name of the group.	String
GroupDescription	ermaximogroupdescription	Specifies the description of the group.	String

IBM® Control Desk service provider attributes by action

The following lists are typical IBM® Control Desk service provider actions that are organized by their functional transaction group. The lists include more information about required and optional attributes sent to the IBM® Control Desk service provider to complete that action.

System Login Add

A System Login Add is a request to create a user account in the domain with the specified attributes.

Table 38. Add request attributes

Required attributes	Optional attribute
eruid ermaximoemailpswd	All other supported attributes

System Login Change

A System Login Change is a request to change one or more attributes for the specified users.

Table 39. Change request attributes

Required attributes	Optional attribute
eruid	All other supported attributes

System Login Delete

A System Login Delete is a request to remove the specified user from the IBM® Control Desk registry.

Table 40. Delete request attributes

Required attributes	Optional attribute
eruid	None

System Login Suspend

A System Login Suspend is a request to disable a user account. The user is not removed, and the attributes are not modified.

Table 41. Suspend request attributes

Required attributes	Optional attribute
eruid eraccountstatus	None

System Login Restore

A System Login Restore is a request to activate a user account that was previously suspended. After an account is restored, the user can access the system with the same attributes before the Suspend function was called.

Table 42. Restore request attributes

Required attributes	Optional attribute
eruid eraccountstatus	None

Reconciliation

The Reconciliation request synchronizes user account information between IBM Security Identity Manager and the service provider.

Table 43. Restore request attributes

Required attributes	Optional attribute
None	None

Index

A

Access card customization
    Request Access wizard 76
access limit customization
    Request Access wizard 81
access types
    changing 114
    creating 113
    deleting 115
    overview 113
access wizard
    account status 86
    compliance information 86
    view 86
accesses
    service based 90
account attribute
    sort 86
account attributes 109
account defaults 108
    add 109
    change 110
    remove 110
account request
    customizing group description 148
account validation logic 201
accounts 175
    policy enforcement 207
activities
    manage 90
    select 90
adapter attributes
    Tivoli Service Request Manager 261
adapter profile 104
administrative console, customization 36
adoption policies 117
approval details
    entities 90
    operations 90
auditingremoving attributes from
    attributesmaximum length for
        auditing 169

B

badge customization
    Request Access wizard 79
banner content customization 33

C

capabilities, Identity Service Center 43
change
    global adoption policies 118
configuration
    files 3
configuration files
    merge customized 48
configure
    default user interface 50

D

configuring
    Justification field 41
console interface
    configuration files 32
    title bar 39
content store, creation 223
create
    global adoption policies 117
css customization, migration 20
customization 31
    account attribute 86
    account status 88
    compliance information 88
    due date notification period 91
    file locations 43
    help files 93
    home page 58
    Identity Service Center files 47
    labels 91
    locale 94
    login page 51
    page header 55
    user interface 1
    user scope 61
customize
    login page 51

D

data source
    IBM Security Identity Manager
        Cognos reports 224
database configuration tool
    (DBConfig) 257
database schema 257
DBConfig command 258
default email message 165
default service types 97
delete, global adoption policies 118
dependencies, export 214
dependent accesses
    group based access 91
    role based access 91
direct-access URL, administrative console
    tasks 36
directory server configuration tool,
    ldapConfig 258
DSML file
    sample
        reconciliation 240
DSML Identity Feed 236, 239
    JavaScript 232
    placement rule, using 252
due date notification period
    customize 91

E

email notification 121
entities 177

entities (continued)

    adding 167
    adding lifecycle rules 189
    adding operations 180
    categories 167
    changing 168, 169
    changing lifecycle rules 190
    changing operations 181
    deleting 170
    deleting lifecycle rules 191
    deleting operations 182
    mapping attributes 167
    overview 167
    running lifecycle rules 191
environment variables settings 224
erRoles attribute 251
event notification, HR feed 233
events 185
export
    deleting 218
    dependencies 214
    full 216, 218
    JAR file 216, 218
    objects 213, 216, 218
    partial 216, 218

F

file customization for user interface 47
file locations for customizable files 43
footer content, customization 35
form designer 129, 130, 131, 132, 133,
   135, 136, 137, 138, 139, 140, 141, 142,
   143, 144, 145, 146, 147, 149
    constraints 159
    control types 152
    interface changes 163
    interface description 149
    properties 159
form templates 163
    modification 130, 131, 132, 133, 135,
      136, 138, 139, 140, 141, 142, 143, 144,
      145, 146, 147, 149, 152, 159
    opening 129, 137
    removal 147
    resetting 149
forms 163
    customization 129, 130, 131, 132, 133,
      135, 136, 137, 138, 139, 140, 141, 142,
      143, 144, 145, 146, 147, 149, 152, 159
    removal 147
Framework manager
    configuration 223

G

global adoption policies
    change 118
    create 117
    delete 118

265

IBM ®

Printed in USA