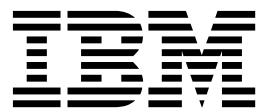IBM Security Identity Governance and Intelligence
Version 5.2.2.1
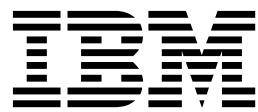
Configuration Topics

IBM

IBM Security Identity Governance and Intelligence
Version 5.2.2.1

Configuration Topics

IBM

Table of contents

Table list

Chapter 1. Dashboards for Service Center

Create and configure Dashboard home pages for users of Service Center. Configure the dashboard items for their home pages that give them a quick overview of conditions to monitor and work to be done.

When a user logs in to the Service Center, the Dashboard home page is shown. The information that is presented depends on the administrative roles that are assigned to the user. A single administrative role can include zero or many dashboard items in its configuration. The dashboard presents the set of all dashboard items for all administrative roles that are assigned to the user.

To configure a user to have access to dashboard items, use the Administrator Console to go through the following process.

1. Review existing dashboard items. In **Report Designer**, click **Manage > Dashboard**. Determine whether an existing dashboard meets your needs.
2. If necessary, create customized dashboard items by:
   - Creating or choosing an existing query and defining a dashboard that is based on its properties.
   - Copying and modifying the dashboard items that are provided with the product.

   See Dashboard.
3. Assign dashboard items to an entitlement.
   a. In **Report Designer**, click **Configure > Assignment**.
   b. Click **Report/Dashboard > Entitlement**.
   c. Filter for **Display Type** of **Dashboard**.
   d. Select a dashboard item.
   e. Select **Actions > Add** in the right pane to add the dashboard item to an existing entitlement.

   Note: To view all of the dashboard items and reports that are assigned to an entitlement,
   a. Click **Entitlement > Report/Dashboard**.
   b. Filter for **Display Type** of **Business Role**.
   c. Click an entitlement. The list of dashboard items that are associated with the entitlement is shown in the right pane.
   d. Use the **Action** menu to add and remove dashboard items.
4. Review the dashboard items that are assigned to the wanted Admin Role to see whether they correspond to the duties you want to assign to the user.
   a. In **Access Governance Core**, click **Configure > Admin roles**.
   b. Select an Admin Role in the left pane, then click the **Management** tab in the right pane.
   c. Review items where **Application** is set to **Report** and **Permission Type** is set to **dashboard**.
   d. Use the **Actions** menu to add and remove dashboard items for the Admin Role.
5. Assign the user to the wanted Admin Role.

a. In Access Governance Core, click **Configure** > **Admin Roles**.
   b. Select an Admin Role in the left pane, then click the **Users** tab in the right pane.
   c. Use the **Actions** menu to add and remove users.

Admin Roles are provided for the following roles. The configuration includes a default set of dashboard items for each role.

- Employee
- User Manager
- Application Manager

See Available dashboard items for the full list of available dashboard items.

Employee dashboard

The Employee dashboard shows dashboard items that are configured in the Employee admin role. When a user is assigned to the Employee admin role, these items are included in the Dashboard home page in the Service Center.

An employee persona is one that interacts with the system through access requests. Access to applications in the system is governed by entitlements. An employee typically needs the following views of access requests and assignments.

- Summary of access requests by status
- Summary of access requests by type
- Access request history
- Access requests that require special attention
- Entitlements assigned

An **Employee** admin role is provided as part of the product. Review it in **Access Governance Core** > **Configure** > **Admin Roles**. Use it as an example of how to configure roles. When a user logs in to the Service Center as a User that has the Employee role, the corresponding dashboard items are shown on the **Dashboard** home page.

If a user has more than one role, the dashboard items are combined. Dashboard items are not repeated.

See Service Center for information about dashboard item controls: Maximize, Refresh, Settings, Filter, Select columns, drill down.

The Dashboard page for an Employee admin role might include the following dashboard items.

Recent requests
   Chart (table) listing your recent requests in date order. Each line shows the following information.
   - **Subject** - Type of request
   - **Step Status** - Status of the request
   - **Request Entity Name** - Application that is the subject of the request
   - **Request Creation Date** - Date of the request

   Note: The provided Employee admin role includes this dashboard item.

My entitlements

Chart (table) listing your entitlements and the following information about each one.

- Entitlement name
- Entitlement type
- Application that is associated with the entitlement
- Permission type

The provided Employee admin role includes this dashboard item.

My requests

Chart (pie) showing a count of your requests by status.

- **Authorizable**
- **Completed**
- **Incompatibility**
- **Rejected**

Hover over a slice to see a count of access requests of that status.

You can also view your password change requests in **Self Care**.

Note: The provided Employee admin role does not include this dashboard item. It is helpful for the role.

Days until the next password expiration

Single number. Number of days before your password expires. Go to **Self Care** to work with your passwords.

Note: The provided Employee admin role does not include this dashboard item. It is helpful for the role.

Partial requests

Single number. Number of your requests that are partially completed.

Note: The provided Employee admin role does not include this dashboard item. It is helpful for the role.

Policy violation requests

Single number. Number of your requests that resulted in a policy violation.

Note: The provided Employee admin role does not include this dashboard item. It is helpful for the role.

Rejected requests

Single number. Number of your requests that were rejected.

Note: The provided Employee admin role does not include this dashboard item. It is helpful for the role.

Access request history

Chart (bar) showing a count of requests per request type. Bar sections show request status.

Hover over a bar section to see a count of requests.

Note: The provided Employee admin role does not include this dashboard item. It is helpful for the role.

User Manager dashboard

The User Manager dashboard shows dashboard items that are configured in User Manager admin role. When a user is assigned to the User Manager admin role, these items are included in the Dashboard home page in Service Center.

A user manager persona typically manages a team and has the following duties:
- Reviewing and approving access requests from the team
- Monitoring risks for team members
- Addressing risks
- Creating access requests to modify permissions and entitlements
- Reviewing access certifications

A **User Manager** admin role is provided as part of the product. Review it in **Access Governance Core** > **Configure** > **Admin Roles**. Use it as an example of how to configure roles. When a user logs in to the Service Center as a user with the User Manager role, the corresponding dashboard items are shown on the **Dashboard** home page.

If a user has more than one role, the dashboard items are combined.

See Service Center for information about dashboard item controls: Maximize, Refresh, Settings, Filter, Select columns, drill down.

The Dashboard page for a User Manager admin role includes the following dashboard items.

Delegation assignments
    Single number. Number of access requests within your team that are delegated to you.

    Go to the **Access Requests** module and click **User Manager** to view delegated requests.

User violations without mitigation
    Single number. Number of users in your team with violations. See the **User Violations** dashboard item for details.

Locked accounts
    Single number. Number of locked accounts in your team.

    Go to **Access Requests** module and click **User Manager** to create an access request to unlock the account.

Accounts expiring in next 30 days
    Single number. Number of application accounts that expire in the next 30 days.

Approval tasks
    Chart (pie) showing requests that are pending approval, by type. Click a slice to drill down to **Access Requests** > **User Manager** and work with individual access requests. Supervisors and reviewers for the campaign can drill down to view and manage access requests according to their role.

    Note:

This dashboard item is not configured through Report Designer. It is not automatically shown in the User Manager dashboard and is available to be shown in the dashboard for any Admin Role.

To see the **Approval tasks** dashboard item, a user must be assigned to an Admin Role that includes access to the **Daily Work** workflow. The **Daily Work** workflow must be configured in Process Designer.

Access certification status
Chart (table) listing access certification campaigns by status. To review a campaign, click it to drill down to **Access Certifier**. Review details there, including end date, percent completed, and supervisor name. You can review campaign items and take action on them there.

Note:

This dashboard item is not configured through Report Designer. It is not automatically shown in the User Manager dashboard and is available to be shown in any dashboard.

To see the **Access campaign status** dashboard item, a user must be assigned as a supervisor or reviewer in at least one access campaign.

Access campaigns are configured in **Access Governance Core** > **Configure** > **Certification Campaigns**. When you select a campaign, you can assign users in the **Supervisors** tab or the **Reviewers** tab in the right pane. See Certification campaigns.

Accounts expiring in next x days with OU scope
Chart (table) listing application accounts that expire in the next 30 days. It is a signal to review the accounts to determine whether they need to be removed or matched again.

Go to the **Access Requests** module and click **User Manager** to create an access request to extend the time for the account.

User violations
Chart (table) listing violations, showing user ID and severity. Violations occur when combinations of user permissions on activities and policies conflict with rules.

Go to the **Access Requests** module and click **User Manager** to create requests to modify permissions on activities.

Application Manager dashboard

The Application Manager dashboard shows dashboard items that are configured in the Application Manager admin role. When a user is assigned to the Application Manager admin role, these items are included in the Dashboard home page in the Service Center.

An application manager persona typically has the following duties:
- Managing application accounts and make sure that they are mapped correctly to users. Application accounts that are not matched to a user account are called orphans. They represent potential risks.
- Matching accounts. Adding and removing user accounts from an application account.
- Mapping business activities to permissions

An **Application Manager** admin role is part of the product. Review it in **Access Governance Core** > **Configure** > **Admin Roles**. Use it as an example of how to configure roles. When a user logs in to the Service Center as a user with the Application Manager admin role, the corresponding dashboard items are shown on the **Dashboard** home page.

If a user has more than one role, the dashboard items are combined.

See ../CrossIdeas_Topics/DESK/Help_Desk.dita for information about dashboard item controls: Maximize, Refresh, Settings, Filter, Select columns, drill down.

The Dashboard page for an Application Manager admin role includes the following dashboard items.

Accounts created in last x days
> Single number. Number of accounts created in the defined interval. It is a signal that new accounts might need to be reviewed, depending on their status.
>
> See **Unmatched Accounts** and **Account Matching Status** dashboard items.

Activities created in the last x days
> Single number. Number of business activities created in the defined interval. It is a signal that activities and permissions might need to be reviewed.
>
> Click the number to drill down to **Business Activity Mapping** > **Business activity Perspective**. Select an activity and use the **Actions** menu to add and remove permissions and rights.

Permissions created in the last x days
> Single number. Number of permissions created in the defined interval.
>
> Click the number to drill down to **Business Activity Mapping** > **Permission Perspective**. Click a permission and use the **Actions** menu to review and manage how the permission is associated to business activities.

Unmatched accounts
> Single number. Application accounts that have status **Unmatched**, which is a signal to match them to user accounts.

Account matching status
> Chart (bar) showing account status by application. For each account, the bar is subdivided by account status. Review application accounts to determine how they are matched to users. The status is one of the following values:
> - **Identity matched** - Accounts that are matched to a user
> - **Unmatched** - Accounts that are not matched to any user account
> - **Orphan** - Accounts that are not mapped to any users
>
> Hover over a bar section to see a count of accounts for that status.
>
> Click a bar section to drill down to the **User-Account Matching** application. A list is shown of all accounts for the selected application (bar) that have the selected status (bar section). In the list, for each account, the application account (**Application UID**) and the system master account (**Master UID**) are shown.
>
> Use the **Action** menu to work with the account.

Accounts expiring in next x days with application scope
Chart (table) showing application accounts that expire in the next 30 days within the Application scope. It is a signal to review the accounts to determine whether they need to be removed or matched again.

Business activity mapping status
Chart (pie) showing the status of how permissions are mapped to business activities. Each slice represents a status. That status is one of the following values:

- **Linked** - Permission is assigned to an activity.
- **Missing** - Permission is manually set to Missing. Cannot determine which activities to assign.
- **Ignored** - Permission is manually set to Ignored.
- **TBD** - Permission is not assigned to an activity and is not manually set to Ignored or Missing.

Hover over a slice to see the count of permissions for that status.

Click a slice to drill down to **Business Activity Mapping** > **Permission Perspective**. In **Permission Perspective**, click a permission and use the **Actions** menu to review and manage how the permission is associated to business activities.

Permissions are associated with activities (Linked). Users are assigned permissions.

To audit business policies like segregation of duties, all activities and users in the system must be mapped. Rules define how to assess individual users, based on the permissions they have on activities.

Dashboards with a business activity scope

More dashboards are displayed in the Service Center of users who have an administrative role that is assigned to particular activities in Process Designer.

The activities in Process Designer must have the following characteristics:
- Activities with a Formal Request functionality in a User Access Change context
- Activities with a Daily Work functionality in a Request Report context

When the Administrator configures any of these activities in Process Designer, the Administrator can set the **Enable dashboard** item to True in **Activity** > **Activity scope** > **Required Data**. This action enables the Dashboard view of the activity in the Service Center of the users who have the administrative role that is assigned to run the activity.

The dashboard provides a shortcut to Access Requests for:
- The User Manager, or similar administrative role, who must generate a request for User access.
  The dashboard displays the business activities of the users that are selected in the **Activity Scope** > **Beneficiary** tab of the activity configuration details (for example, **All Users**, **Actor**, **All Users belonging to an OU**, and more).
  To generate a request for User access, the manager can click a business activity in the dashboard to list the users who are associated with the business activity. The manager can then generate the request from this point.
- A user with an administrative role who must authorize generated requests or execute authorized requests.

The dashboard displays the business activities that are associated with the requests that await authorization or execution.

To quickly find out the requests that await action, the manager or operator can click the business activities in the dashboard to display lists of associated requests.

Chapter 2. External authentication by use of reverse proxy

You can use a reverse proxy to authenticate users with Active Directory or LDAP.

The product supports external authentication by using HTTP(S) header information that is supplied by a proxy.

The reverse proxy must support and use the WebSocket.

The setup of a reverse proxy to authenticate users with Active Directory or LDAP requires the following steps:

1. Configure reverse proxy server authentication for Active Directory or LDAP.
2. On the reverse proxy, configure the following HTTP(S) headers:
   - Set `IV-User` to the User ID of the authenticated user
   - Set `realm` to the string `"ideas"`
3. Enable Identity Governance and Intelligence for HTTP header-based authentication, as explained in "Configuring the IBM Security Identity Governance and Intelligence server."

Configuring the IBM Security Identity Governance and Intelligence server

Use the Custom File Management page to configure the IBM® Security Identity Governance and Intelligence server.

Procedure

1. From the top-level menu of the **Appliance Dashboard**, select **Configure** > **Manage Server Setting** > **Custom File Management**.
2. In the Custom File Management page, click the **All Files** tab.
3. Go to `directories/properties`.
4. Create a folder, and name it as **desk**. For more information about creating a folder, see Managing custom files.
5. Under the **desk** folder, create a subfolder and name it as **console**.
6. Create an xml file on your computer, and name it as `application.xml`.
7. Add the following contents in the `application.xml` file.

   ```
   <DESK>
   <REALM name="Ideas" label="IDEAS" isDefault="true" enableHeaderAuth="true"/>
   </DESK>
   ```

8. Upload the `application.xml` file in the **console** folder. For more information about uploading a file, see Managing custom files.
9. On the **Appliance Dashboard** of the IBM Security Identity Governance and Intelligence virtual appliance console, locate the **Server Control** widget.
10. Select **Identity Governance and Intelligence server**.
11. Click **Restart**. For more information, see Viewing the server control widget.
12. Log on to the IBM Security Identity Governance and Intelligence Central Administration console with administrative credentials.
    - User: `admin`
    - Password: `admin`

13. In the Access Governance Core console, from Realm **IDEAS**, click **Settings** > **Core Configurations** > **General**.
14. In the **Access** section, verify that the **Login User ID** check box is selected.
15. Use the `https://<hostname>/ideas/desk?realm=IDEAS` URL to call the desk console by bypassing the usual Login page.

Chapter 3. Customization features in IBM Security Identity Governance

IBM Security Identity Governance and Intelligence supports a wide range of customizations.

Customization features allow IBM Security Identity Governance and Intelligence users to change different characteristics.

Accessing resources for customization

Customizations work with specific resources.

You can access these resources with the indications in ../installing/tsk/ t_managing_custom_files.dita.

You must have virtual appliance administrator authority. From the virtual appliance dashboard, click **Configure** > **Custom File Management**. On the **All Files** tab, all the available directories are shown under the root folder named directories.

directories/connectors
> In this directory, you can copy the files for the Enterprise Connectors module.

directories/log
> Downloading is allowed.

directories/db
> Downloading and uploading are allowed.

directories/lib
> This directory can contain custom libraries. Downloading and uploading are allowed.

directories/lib/native
> This directory can contain SAP libraries. Downloading and uploading are allowed.

directories/properties
> Downloading and uploading are allowed.

To complete the customization tasks that are described in the following sections, you must download the SDK: directories/SDK/sdk.zip.

Logo and Banner Customization

You can customize the header of the console.

You can customize the following elements:
- The logo, logo.png. Its size must be 60 pixels by 222 pixels.
- The graphic banner, headerWave.png. Its size must be 65 pixels by 1366 pixels or 65 pixels by 1 pixel.
- The text parameters of the banner, including the font and the color.

The following images show examples of the header.

Demo   IDEAS Central Administration

CrossIdeas IDEAS Central Administration
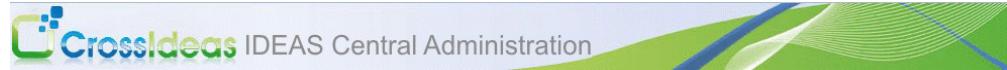
Graphics requirements for the logo and banner

You can customize the banner that is used for a single realm or all realms.

- Stop the application server before you start the customization process.
- Start the application server after you complete the customization process.
- Override mode is used to set the banner for all realms.
- Realm-specific mode is used to set the banner for one realm.

The following graphics show the exact dimensions required.

The graphics format must always be PNG.

logo.png

60px

CrossIdeas

222 px

headerWave.png

65px

1366 px
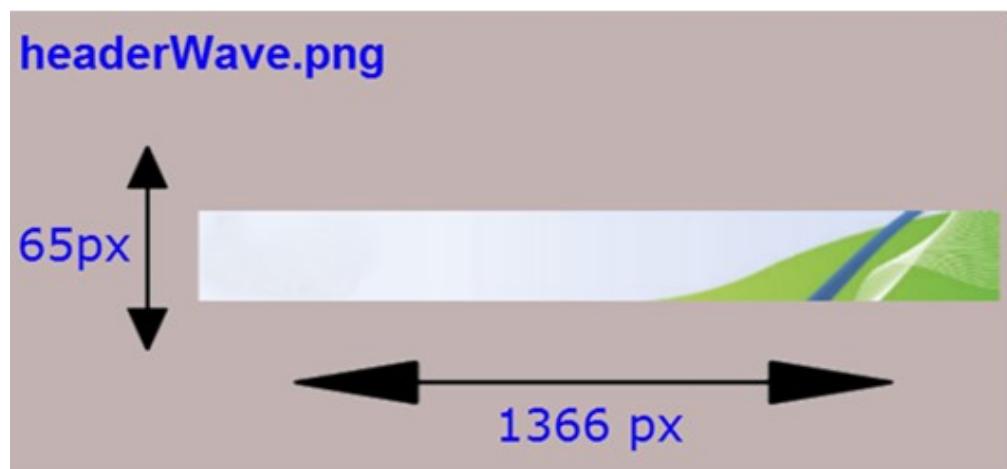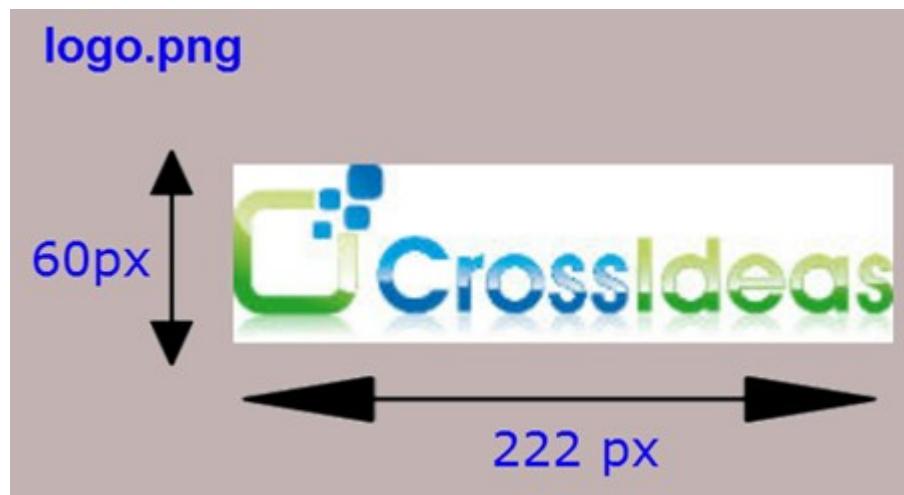
Note: The size of `headerWave.png` can also be 65 x 1 pixels.

Override customization

You can maintain copies of original files to help override customization later.

Stop the application server before you start this process.

1. In this example process, the following JAR file is used.
   `WhiteLabel<suffix_string>.jar`
2. Make a copy of the file before you modify the original file.
3. Open the file `WhiteLabel<suffix_string>.jar`. Find the folder
   `com\crossideas\toolkit\web\resource\images\crossideas`.
4. Edit the contents of the folder for each environment:
   - For the **Administration Console** environment, replace the files `logo.png` and
     `headerWave.png` with the customized files in the `com\crossideas\toolkit\`
     `web\resource\images\crossideas` folder. Do not change the names.
   - For the **Service Center** environment, complete the following steps:
     a. Replace the `sc-logo.png` file with the customized files in the
        `com\crossideas\toolkit\web\resource\images\crossideas` folder to
        change the Service Center logo. Do not change the names.
     b. To change the application name and the color of the banner, edit the
        `com/crossideas/toolkit/web/resource/ExternalCustomization.css` file:

        ```
        / NOTE: EDIT WITH EXTREME CARE.
         Purpose of this file is to allow external style customizations.

        1. To change the background-color of header do following
        .igi_header {
        background-color: red !important;
        }
        2. To change the product name
        .app-name:after{
        content: "My Demo product";
        }
        3. To change the logo  upload sc-logo.png (refer knowledge center)
        /
        ```
5. Save the file `WhiteLabel<suffix_string><MyName>.jar`.
6. Move the saved file to the `directories/lib` folder accessible as described in
   "Accessing resources for customization" on page 11.

Customizing a realm

The consolidated process for customizing a realm includes customizing the realm
stylesheet and the graphics for the logo and banner.

In this process, you open an existing `.JAR` file. Then, you modify and save it under
a new name.

In this example process, the following `.JAR` file is used: `WhiteLabelCustom.jar` in
the `customization` folder of SDK.

1. Stop the application server.
2. If you have not yet done so, download SDK.
3. Make a copy of the `.JAR` file.
4. Open the `.JAR` file.

5. Create the directory <realm_name> in com\crossideas\toolkit\web\resource\
   images\crossideas. The new directory is the name of the realm to be
   customized. Use all lowercase letters.

6. Save the customized files `logo.png` and `headerWave.png` in the new
   <realm_name> directory.

7. Make a copy of \com\crossideas\toolkit\web\resource\
   `DefaultStyleExample.stylesheet.xml`. An example of this file is in
   `WhiteLabelCustom.jar`.

8. Rename the copy as `DefaultStyle<Realm_name>.stylesheet.xml`. For
   <Realm_name>, substitute the name of the realm to be customized.

   Note: You must capitalize the first letter of the realm name in this path.

9. Edit the renamed file.

10. Save the modified `.JAR` file under a new name:`WhiteLabelCustom<MyName>.jar`.

11. Move the modified `.JAR` file to `directories/lib`. It is accessible. See
    "Accessing resources for customization" on page 11.

12. Optional. Repeat steps 5to 9 for every realm to be customized in the `.JAR` file.

13. Start the application server.

- *<suffix_string>* is an existing string of characters in the product.
- *<MyName>* is a string of characters that you define.

Customizing the realm stylesheet

The realm stylesheet is in the form `DefaultStyle<realm_name>.stylesheet.xml`.

For the <realm_name> token, use the name of the realm. Capitalize the first letter of
the realm.

For example, for realm DEMO, use the following form:
`DefaultStyleDemo.stylesheet.xml`

The file includes three attributes:
- `App.Header`: the image file to be used as background in the header and the
  background color
- `App.Logo`: the image file to be used as the logo
- `Title.Menu`: the font and color of the label that indicates the name of the
  application

The following code is a full example.

```
<ss>
    <s n="App.Header" t="nextapp.ech.app.ContentPane">
            <p n="backgroundImage" t="FillImage">
            use <fi r="x" t="r"> to repeat image in x axis
            (i.e. homogeneous background) 65 x 1 pixels is enough
            or <fi r="0" t="r"> for fix image 65 x 1366 pixels
                    <i
r="/com/crossideas/toolkit/web/resource/images/demo/headerWave.png" />
            </fi>
        </p>
        <p n="background" t="Color">#427ab3</p>
    </s>
    <s n="App.Logo" t="nextapp.echo.app.Button">
        <p n="height" t="Extent">60px</p>
        <p n="width" t="Extent">222px</p>
        <p n="LayoutData" t="nextapp.echo.app.Layout.GridLayoutData">
                <p n="insets" t="Insets">2px 2px 2px 2px</p>
        </p>
        <p n="backgroundImage" t="FillImage">
            <fi r="0" t="r">
                    <i
r="/com/crossideas/toolkit/web/resource/images/demo/logo.png" />
            </fi>
        </p>
        <p n="background" t="Color">#427ab3</p>
    </s>
    <s n="Title.Menu" t="nextapp.echo.app.Label">
        <p n="font" t="Font">
                <f bo="0" sz="26px">
                    <tf>Trebuchet MS </tf>
                </f>
        </p>
        <p n="foreground" t="Color">#004666</p>
        <p n="insets" t="Insets">0px 0px 0px 0px</p>
    </s>
</ss>
```

The following sample files are included with the product.

DefaultStyleDemo.stylesheet.xml
> This file is included in the following folder. `com\crossideas\toolkit\web\resource`

logo.png and **headerWave.png**
> These files are included in the following folder. `com\crossideas\toolkit\web\resource\images\demo`

Labels globalization and customization of the Identity Governance and Intelligence GUIs

You can globalize or customize the user interface modules.

You can customize configuration files. The following sample shows how to do it:

```
...
......
LoginWin.InvalidLogin.Title=Invalid Login
LoginWin.InvalidLogin.Message=Incorrect UserID or Password
LoginWin.InvalidLogin.CfgError=Configuration file not found
......
reamiwin.titolo=Realm Selection
reamiwin.label.reame=Realm
reamiwin.button.reame=Realm Login
......
cambiapwdwin.titolo=Change Password
cambiapwdwin.utente=User
cambiapwdwin.password=Password
cambiapwdwin.confermapwd=Confirm Password
......
...
```

Every configuration file is composed of a list of rows.

Every row is composed of a `<KEY>=<Value>` pair:

- `<KEY>` is a string that can include several items that are separated by a dot. For example:

  `<KEY>=ITEM1.ITEM2.ITEM3...ITEMn`

`<Value>` is used to specify the customized term.

Table 1. Information about configuration file formatting

	Note:
	• Do not use spaces before and after the '=' sign.
	• Do not modify the default configuration files
	.

Customization basic approach

You can customize some labels with properties files.

The customizable labels of IBM Security Identity Governance and Intelligence modules are defined in `CustomMessages_<language>.properties`. The suffix *<language>* is a standard code that is defined by ISO 639-1. See http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes.

Examples:

`CustomMessages_en.properties` for the English language

`CustomMessages_it.properties` for the Italian language

`CustomMessages_nl.properties` for the Dutch language

CustomMessages_<language>.properties is a generic file in directories/
properties/<module_name>/console/localization that is accessible in the SDK.
You must download it. See "Accessing resources for customization" on page 11.

Customization procedure

You can customize a set of labels.

1. Create a dedicated *<modulename>*/console/localization folder in the
   directories/properties directory accessible as described in "Accessing
   resources for customization" on page 11.
2. In the directory localization, create a text file named
   CustomMessages_*<language>*.properties.
3. Edit the file. Add rows that are formatted as *<KEY>=<Customized_Value>*.
4. Stop the application server.
5. Start the application server. It uses the updated labels.

Table 2. Note about the customization procedure

⚠	**CAUTION:** **Do not modify the key.** **The procedure must act only on the** `Customized_Value` **items.**

Customizable Identity Governance and Intelligence modules

You can customize a limited set of modules.

The following table lists the IBM Security Identity Governance and Intelligence
modules that you can customize. They are accessible in directories/properties.
See "Accessing resources for customization" on page 11. This directory is labeled
PREFIX in the Path and Log of labels column of the table.

Table 3. Customizable modules

Identity Governance and Intelligence Module	Module Name `MODULENAME`	Path and Log of Labels
Desk	Service Center `SERVICECENTER`	*<PREFIX>*/desk/console/localization/ CustomMessages_<language>.properties
ARM (Access Request Manager)	Access Provisioning `ACCESSPROVISIONING`	*<PREFIX>*accessprovisioning/console/localization/ CustomMessages_<language>.properties
ACCM (Account Matching)	User-Account Matching `USERACCOUNTMATCHING`	*<PREFIX>*/accountmatching/console/localization/ CustomMessages_<language>.properties
EOB (Easy onboarding)	Business activity mapping `BUSINESSACTIVITYMAPPING`	*<PREFIX>*/onboarding/console/localization/ CustomMessages_<language>.properties
REPORTS	Reports `REPORTS`	*<PREFIX>*/reports/console/localization/ CustomMessages_<language>.properties
AC (Access Certifier)	Access Certifier `ACCESSCERTIFIER`	*<PREFIX>*/accesscertifier/console/localization/ CustomMessages_<language>.properties

Table 3. Customizable modules (continued)

Identity Governance and Intelligence Module	Module Name `MODULENAME`	Path and Log of Labels
COMMON	Not applicable	*`<PREFIX>`*`/common/console/localization/` `CustomMessages_<language>.properties`

Introduction to realm and Service Center customization

To customize a realm or add an application on the Service Center, edit the `application.xml` file.

The file is in the `directories/properties` folder. You can access the file as described in "Accessing resources for customization" on page 11.

You can customize the following:
- Realm attributes
- Service Center

After you edit the file, restart the application server.

Customizing realm attributes

You can customize some realm attributes.

A realm is represented by the tag REALM. The following attributes can be modified.
- `name` is a string that sets the name of the realm. It is registered in advance in the core database.
- `label` is a string that hosts the name of the realm.
- `default` is a Boolean value that indicates whether this realm is the default Realm. If the value is `true`, the realm is the only one present.
- `enableHeaderAuth` is a Boolean value. If the value is `true`, then accessing the realm is based on the header. Log with `DN=IV-USER`. The default value is `false`.

The following code is an example:

```
<REALM name="DEMO" label="DEMO" default="true"  enableHeaderAuth="true">
...
</REALM>
```

Customizing the Service Center

An application is represented by the tag APPLICATION and specified in the REALM tag in the Service Center. You can customize some attributes.

You can customize the following attributes:
- `name` This attribute is a string. It sets the name of the application to be added to the Service Center. It is registered in advance in the core database.
- `label`: This attribute is a string. Set it to the name of the application on the Service Center.

- url: This attribute is a string. It sets the access URL of the Service Center. If the application runs on a server other than the server for the Service Center, add the string `http://` or `https://`.
- `isEndUser`: This attribute is a Boolean value. The default value `false` indicates an application user. Set it to `true` to indicate an administrative user.
- `showIfDisabled`: This attribute is a Boolean value. The default value of `true` means that the button for accessing the application is shown to a user who is not authorized to manage that application.
- `order`: This attribute is a number. Set it to the position of the application on the Service Center.
- `auth`: This attribute is a string. It indicates the authentication method. Set it to `ITOKEN` or `SAML`. The default value is `ITOKEN`.
- `openInNewWindow`: This attribute is a Boolean value. The default value `false` means that the applications you select in the Service Center are displayed in the same window. Change to `true` to display applications in a new window.
- `checkISIGPermission`: This attribute is a Boolean value. The default value `true` means that applications are displayed only after the user's authorization is verified.

The following attributes specify images that are used with the application. The images must be placed in a `.JAR` file in the folder for custom libraries.
- `img` specifies a PNG image. The image must be 75x75 pixels.
- `img_dis` specifies a PNG image. The image must be 75x75 pixels. The image is disabled.
- `img_rol` specifies a PNG image. The image must be 75x75 pixels. The image has a roll-over effect.

The following code sample shows how to set up customization.

```
REALM name="DEMO" label="DEMO">
<APPLICATION name="AD-DEMO" label="AD-DEMO" url="../custom" img="" img_dis="" img_rol="" order="10" isEndUser="true" auth="SAML" openInNewWindow="true"/>
 ...
</REALM>
```

Add a description for use under **Application**.

Set the text of the description in key`application.description.`*`application_name`*.

For example:

```
application.description.ad-demo=My first application<br>On IGI Service Center<br>Well done
```

Place the key in the following file: `customization/IDEASPlatformEnvCustom/properties/desk/console/localization/CustomMessages_<language>.properties`. This folder is accessible as described in "Accessing resources for customization" on page 11.

To avoid the specifying the attributes `img`, `img_dis_` and `imag_rol`, follow these steps:
1. Create three images: `Desk_application_name_btn.png`, `Desk_application_name_btn_dis.png`, and `Desk_application_name_btn_rol.png`.
2. Store images in the package `com\crossideas\toolkit\web\resource\images\desk\btnrow\75x75\`.
3. Create a `MyCustomDesk.jar` file that contains the package from step 2.

4. Move this last file to the `directories/lib` folder as described in "Accessing resources for customization" on page 11.

Time and date customization

You can customize the time and date format for all modules.

The time and date formats are compliant with the Java/Oracle time and date specification. See Java/Oracle specification.

For customizing time and date, you have to consider the file `application.xml`.

A fragment of the file `application.xml` is shown in the following code sample.

```
<DESK>
<CONFIG>
<DATEFORMAT>
<DEFAULT>....</DEFAULT>
<DATE_AND_TIME>....</DATE_AND_TIME>
<DATE_AND_TIME_SECONDS>....</DATE_AND_TIME_SECONDS>
<DATE_ONLY>....</DATE_ONLY>
<TIME_ONLY>....</TIME_ONLY>
</DATEFORMAT>
</CONFIG>
</DESK>
```

You can find an example of this file into the `sdk.zip` file.

For downloading the sdk.zip file, access through the Virtual Appliance User Interface, in `Configure>Manage Server Setting>Custom File Manegement`.

Browsing into the directory tree in **All Files** tab, you can find the folder **sdk**, where you can download `sdk.zip`.

The file `application.xml` is stored in:

`sdk.zip\customization\IDEASPlatformEnvCustom\properties\desk\console`.

Use the following procedure.
1. Stop the application server.
2. Update the fragment of `application.xml` file involved in the time and date customization (if not present, you can add it (see the previous code sample)).
3. In **All Files** tab, find the folder `properties`
4. If not present in `properties`, create the folder`desk`
5. If not present in `desk`, create the folder`console`
6. Save the file `application.xml` into `console` folder.
7. Start the application server.

The time and date formats are compliant with the Java/Oracle time and date specification. See Java/Oracle specification.

A fragment of `application.xml` file involved in the time and date customization is shown in the following code sample.

```
<DESK>
<CONFIG>
<DATEFORMAT>
<DEFAULT>yyyy:dd:MM</DEFAULT>
```

```
<DATE_AND_TIME>yyyyy.MMMMM.dd GGG hh:mm aaa</DATE_AND_TIME>
<DATE_AND_TIME_SECONDS>yyyyy.MMMMM.dd GGG hh:mm:ss aaa</DATE_AND_TIME_SECONDS>
<DATE_ONLY>yyyy:dd:MM</DATE_ONLY>
<TIME_ONLY>HH:mm:ss</TIME_ONLY>
</DATEFORMAT>
</CONFIG>
</DESK>
```

SAP Libraries

The ARCS module must access two SAP libraries to connect to SAP systems.

The ARCS module must access the following SAP libraries:

- `sapjco3.jar`
- `libsapjco3.so`

1. Log in as administrator of the virtual appliance.
2. Select **Configure** > **Custom File Management**. All directories are available in **All Files** under`directories`.
3. Put the libraries in the following location: `directories/lib/native`.

For more information, see "Accessing resources for customization" on page 11.

Chapter 4. Single sign-on overview

You can choose from several protocols to authenticate a user for the security applications that the user has rights to. After the first login, further authentication prompts by the system are eliminated as the user switches applications during the same session.

Identity Governance and Intelligence provides alternative options to set up single sign-on for the Service Center with IBM Security Identity Manager and other applications.

OpenID Connect (OIDC)
> Uses an external OIDC provider that works as a single point of authentication.

Lightweight Third-Party Authentication (LTPA)
> Uses the LTPA token generated directly by the application servers.

ISAM-based authentication
> Uses LTPA keys that are shared by ISAM and the application servers.

Authentication with OpenID Connect

The authentication process can be outlined in the steps that follow.

1. A user who requests to log in to the Identity Governance and Intelligence Service Center is redirected to an authentication form on the authorization endpoint of the OIDC provider.
2. The user submits the login information and upon authentication is redirected to its redirect end point.
3. At the same time, a session-like cookie is created on the OIDC provider for further requests to the authorization endpoint.

   The presence of this cookie assures that when the user wants to access IBM Security Identity Manager and is redirected to the authorization endpoint, the user does not need to be authenticated again.

Authentication with the LTPA token

A user who is authenticated in the login page of IBM Security Identity Manager can access the Service Center, or vice versa, with no need to authenticate again. The LTPA token - a cookie that is generated by the Application Server - is used to achieve this result under the following conditions:

- The Security Identity Manager and the Service Center user interfaces support the use of the LTPA cookie. This condition applies to the user interfaces of all the applications that are to be included in the single sign-on perimeter.
- The LTPA key is shared by the application servers.
- The user accounts that are enabled for single sign-on are registered in the Security Identity Manager LDAP user repository and in the Identity Governance and Intelligence database for the Service Center.

ISAM-based authentication

This solution fits best existing Security Identity Manager with ISAM for SSO installations. It is based on authenticated LTPA junctions for the Identity

Governance and Intelligence Service Center and for Security Identity Manager. The URLs that gain access to the respective junctioned application servers change accordingly.

- `https://< WebSEALHostAndPort >/igiauth/service?realm=ideas` for the Service Center
- `https://< WebSEALHostAndPort >/isim/itim/console` for the Security Identity Manager console.

The LTPA keys are shared by ISAM and the application servers.

Another unauthenticated junction is defined for the Identity Governance and Intelligence Administration Console so that it bypasses ISAM authentication. It is used by the Administration Console users who are authenticated directly by the application server and are out of the single sign-on context.

Configuring Identity Governance and Intelligence for single sign-on

There are two ways to do this task, depending on the type of solution that is selected to provide single sign-on authentication.

Using external authentication
> If your authentication method is based on ISAM or on an OIDC Provider, the sign-on process uses the name of the account that is owned by the user.
> - If the account is not `Ideas`, the name of the account becomes the authentication code. This name and the user ID become the authentication parameters that are used to grant access to the multiple applications.
> - If the account is `Ideas`, one of the configured attributes of `Ideas` becomes the authentication code. This name and the user ID become the authentication parameters that are used to grant access to the multiple applications.
>
> The Identity Governance and Intelligence administrator must select the account name in the **Settings** page of Access Governance Core. With the `Ideas` account, the administrator must also select one of the attributes that are listed for this account.

Using internal authentication
> If your authentication method is based on the use of LPTA keys, the keys are passed between Identity Governance and Intelligence and Security Identity Manager by the Identity Governance and Intelligence virtual appliance. To enable the exchange, you must ensure that two conditions be true for the application servers that run the Identity applications.
> - They are well synchronized
> - They run in the same realm

See also:
- "Configuring access for single-sign-on using external authentication" on page 25
- "Configuring access for single-sign-on using internal authentication" on page 25

Configuring access for single-sign-on using external authentication

Take these steps to select the login credentials for users to the Service Center if your single sign-on authentication system is based on Open IDConnect (OIDC) or on IBM Security Access Manager (ISAM).

About this task

Follow these steps to define the login credentials if you are using single sign-on authentication for the Service Center and Security Identity Manager, or other applications. The credentials that Identity Governance and Intelligence sends to the single sign-on service provider to authenticate a user are

- The user's ID
- The name of an account that the user can access

Procedure

1. Log in to the Administration Console and go to Access Governance Core.
2. Select **Settings** > **Core Configurations** > **General** and scroll down to the **Access** section.
3. Select **Login User ID**. The Login User ID window is displayed, where you are asked to provide the account information.
4. Scroll the **Account** field and select an account in the list.

   If you select the `Ideas` account, scroll the **Attribute** field to select an attribute of this account that is used to authenticate the user.

   If you select another account, no attribute selection is requested and the **Attribute** field disappears.

   If you do not make a selection, the `Ideas` account and the `code` attribute are used by default.
5. Click **OK** to close the Login User ID window and click **Save** to save your settings. User access to the Service Center is now configured for single sign-on.

Configuring access for single-sign-on using internal authentication

Take these steps to specify that you run a single sign-on authentication system that is based on the use of Lightweight Third-Party Authentication keys to authenticate users to the Service Center, Security Identity Manager, and other applications.

About this task

If your governance and identity system is set up for a single sign-on authentication method that runs on the use of Lightweight Third-Party Authentication keys, users of the Service Center, Security Identity Manager, and any other applications within the single sign-on environment need to provide their credentials only one time within the browser session.

Follow these steps to define in Identity Governance and Intelligence that you use this authentication method.

Procedure

1. Log in to the Administration Console and go to Access Governance Core.
2. Select **Settings** > **Core Configurations** > **General** and scroll down to the **Access** section.

3. Select **Internal authorization**. After their first login to one of the applications in the single sign-on environment, users are not prompted for further authentication to gain access to the other applications in that environment. The virtual appliance does the transfer of the LTPA tokens that include the user credentials to the other applications. The tokens are valid for the lives of the browser sessions.

Chapter 5. Configuring iToken to implement a custom single sign-on

The iToken provides a way to implement a custom single sign-on (SSO) between Identity Governance and Intelligence modules. It can be also used by custom applications that use Identity Governance and Intelligence as an authentication system.

About this task

An iToken object has a lifetime of one only minute. It can be used only one time for an authentication action.

The `type` attribute specifies the type of process to use with the activity.

Note: Java™ classes and the related methods that are used for iToken management are in the Security API and delivered through the Software Development Kit. For related information, see Application programming interfaces.

To implement a custom single sign-on with iToken, complete these steps:

Procedure

1. Authenticate on Identity Governance and Intelligence. In the basic scenario, a user is already logged in to Identity Governance and Intelligence, and the custom application is integrated. In this scenario, the Security Context is already instantiated. For related information, see Application programming interfaces.

   ```
   SecurityContext sc = new SecurityContext(properties);
   sc.login(userid,password,realm);
   ```

2. Get the iToken from the Security Context with the method `getInternalToken()`.

   ```
   String iToken = sc.getInternalToken();
   ```

3. Send the iToken to the custom application with HTTP POST request by using a `.jsp` and setting the following parameters:

   ```
   <form NAME="AccessPTC" METHOD="POST" ACTION="<%=URLDecoder.decode(url)%>">
     <input type="hidden" name="itoken" value="<%=itoken%>">
     <input type="hidden" name="realm" value="<%=realm%>">
     <input type="hidden" name="realmwork" value="<%=realmWork%>"></form>
   ```

   It gets the realm value from the Security Context:

   ```
   String realm = sc.getRealmLogin();
   ```

   Usually the `realm` and the `realmwork` are the same. In some situations, `realm` is the realm where you logged in; `realmwork` is the distinct realm where you want to work.

4. From the custom application side, get the iToken that is embedded in the request. The custom application must get the iToken that is sent in the previous step.

   ```
   String itoken = request.getParameter("itoken");
   String realm = request.getParameter("realm");
   ```

5. Create the Security Context.

   ```
   SecurityContext sc = new SecurityContext(properties);
   ```

6. Log in using the iToken from step 4.

```
SecurityContext sc = new SecurityContext(securityProperties);
 sc.loginInternalToken(internalToken, realm);
```

Chapter 6. Integration with IBM Security Identity Manager

Use a special connector to integrate IBM Security Identity Governance and Intelligence with IBM Security Identity Manager.

For instructions and further documentation, see *Integration between IBM Security Identity Manager and IBM Security Identity Governance* at http://www-01.ibm.com/support/docview.wss?uid=swg21968516.

Chapter 7. Desktop Password Reset Assistant

The Desktop Password Reset Assistant (DPRA) enables Windows users to reset and change their Windows passwords from their desktops.

The Desktop Password Reset Assistant uses the Identity Governance and Intelligence password validation.

Note:
- Identity Governance and Intelligence password services must be configured by the administrator. For information about setting up security questions and enabling the password service, see Configuring the password service in Access Governance Core. In the **Configure Password Service** tab, these options must be enabled:
  – Enable password service
  – Allow user to change password immediately if you want to change password immediately
  – Send system-generated password to known user email address if you want to reset the password
- Security questions and answers must be configured by the user. For more information, see Updating my security questions.
- The Feedback task must be enabled. To enable feedback, go to the Task Planner module. Click **Manage** > **Tasks**. In the left navigation pane, select the Feedback task. Then click **Actions** > **Start**.

After resetting the password, users must check the email that is registered in Identity Governance and Intelligence to retrieve the new password.

After changing the password, users must wait until the request is complete in the Identity Governance and Intelligence workflow before they can log in with the new password.

Unlocking accounts and password synchronization are not supported.

For information about configuring the Desktop Password Reset Assistant to work with Identity Governance and Intelligence, see the *Desktop Password Reset Assistant Installation and Configuration Guide* at http://www.ibm.com/support/knowledgecenter/SSIGMP_1.0.0/com.ibm.itim_pim.doc/dpr/install_config/dpr_html_mstr.htm.

The Desktop Password Reset Assistant is distributed as a Credential Provider for Windows Vista, Windows 7, Windows 8, and Windows 2008 server and 2012 server. It is visible as a link under the password prompt when you are logged out or when the computer is locked.

Chapter 8. Secure authentication for DAML- based adapters

You can configure the DAML- based adapters for one-way or two-way secure authentication with signed certificates.

Configuring certificates for one-way SSL authentication

In this configuration, Identity Brokerage and the DAML-based adapter use SSL.

About this task

Client authentication is not set on either application. The Identity Brokerage server operates as the SSL client and initiates the connection. The adapter operates as the SSL server and responds by sending its signed certificate to the Identity Brokerage. Identity Brokerage uses the installed CA certificate to validate the certificate that is sent by the adapter.

Procedure

1. On the adapter, complete these steps:
   a. Start the certTool utility.
      1) Open the **Command Line Tool** and go to the following directory: *AGENT_INSTALLED_DIR*\Agents\adapter_nameAgent\bin\
      2) Run the command **CertTool -agent adapter_nameAgent**

         adapter_nameAgent is the name of the adapter agent, such as, NotesAgent or WinLocalAgent.
   b. Install a CA certificate in the adapter registry
      1) To configure the SSL-server application with a self-signed certificate
         a) Create a keystore for the adapter. See "Creating a PKCS12 keystore file for the adapter" on page 35.
         b) Create a self-signed certificate for the adapter. See "Creating a self-signed certificate for the adapter" on page 35.
         c) Install the created self-signed certificate in the adapter registry. See "Installing the certificate and key from a PKCS12 file" on page 36.
      2) To configure the SSL-server application with a signed certificate from a certificate authority.
         a) Create a certificate signing request (CSR) and private key. This step creates the certificate with an embedded public key and a separate private key and places the private key in the PENDING_KEY registry value.
         b) Submit the CSR to the certificate authority by using the instructions that are supplied by the CA. When you submit the CSR, specify that you want the root CA certificate to be returned with the server certificate.
2. On the virtual appliance where the Identity Brokerage is installed, complete one of these steps, take the following actions:
   - If you generated the self-signed certificate with the key management utility of another application, take the following actions:
      a. Extract the certificate from the keystore of that application. See "Extracting a CA certificate for the adapter" on page 36.

33

b. Add it to the truststore of the virtual appliance. See "Importing the Identity Brokerage CA certificate in the virtual appliance" on page 37.
- If you used a signed certificate from a well-known CA, take the following actions:
   a. Ensure that the virtual appliance where the Identity Brokerage is installed, stored the root certificate of the CA (CA certificate) in its truststore.
   b. If the truststore does not contain the CA certificate, extract the CA certificate from the adapter and add it to the truststore of the virtual appliance. See "Extracting a CA certificate for the adapter" on page 36.

Configuring certificates for two-way SSL authentication

In this configuration, the Identity Brokerage application and adapter use SSL.

Before you begin

Configure the adapter and the Identity Brokerage application for one-way SSL authentication. See "Configuring certificates for one-way SSL authentication" on page 33.

About this task

The adapter uses client authentication. After the adapter sends its certificate to the server, the adapter requests identity verification from the Identity Brokerage application. Thevirtual appliance server is where Identity Brokerage is installed. It sends its signed certificate to the adapter. Both applications are configured with signed certificates and corresponding CA certificates.

Table 4. Difference between a signed and an exported CA certificate

If you use signed certificates from a CA	If you use the default CA certificates exported from the virtual appliance keystore file
• The CA provides a configured adapter with a private key and a signed certificate. • The signed certificate of the adapter provides the CA certification for the Identity Brokerage application.	• The CA certificate is extracted from the virtual appliance keystore file. • The CA certificate of the adapter provides the CA certification for the Identity Brokerage.

Procedure

1. If you use signed certificates from a CA, follow these steps.
   a. Install the CA certificate in the keystore of the virtual appliance. See "Importing the Identity Brokerage CA certificate in the virtual appliance" on page 37.
   b. Export the certificate from virtual appliance to temporary folder. See "Exporting the virtual appliance CA certificate for Identity Brokerage" on page 37.
   c. On the adapter, add the CA certificate that is extracted from the keystore of the virtual appliance. See "Importing the virtual appliance CA certificate in the adapter registry" on page 38.
2. If you use the default CA certificates that are exported from the virtual appliance keystore file, follow these steps.

a. Create a CA certificate for Identity Brokerage. See "Exporting the virtual appliance CA certificate for Identity Brokerage" on page 37.

b. Import the virtual appliance CA Certificate in the adapter registry. See "Importing the virtual appliance CA certificate in the adapter registry" on page 38.

c. Stop and restart the adapter service. See "Start, stop, and restart the adapter service" on page 38.

d. Restart the virtual appliance.

Results

After you configure the two-way certificate, each application has its own certificate and private key. Each application also has the certificate of the CA that issued the certificates.

Creating a PKCS12 keystore file for the adapter

Before you begin

If you do not have it, you must get the IBM Java Runtime Environment package where the **ikeyman** tool is included.

Procedure

1. Access the directory where the **ikeyman** tool is located. For example, *IBM_SOFTWARE_INSTALLED_DIR*\Java\jre\bin.
2. Start the **ikeyman.exe** file (for Windows operating systems) or **ikeyman** (for UNIX and Linux operating systems).
3. From the **Key Database File** menu, select **New**.
4. Select the key database type of **PKCS12**.
5. Enter the keystore file name. For example, agentkeys.p12.
6. Enter the location. For example, *AGENT_INSTALLED_DIR*/Agents/keys.

   Note: Ensure that location that you specify exists.
7. Click **OK**.
8. Type a password for the keystore. The default password is secret.
9. Click **OK**.

Creating a self-signed certificate for the adapter

A self-signed certificate contains information about the owner of the certificate and the signature of the owner. This type of certificate is typically used in a testing environment.

Procedure

1. Access the directory where the **ikeyman** tool is located. For example, *IBM_SOFTWARE_INSTALLED_DIR*\Java\jre\bin.

   Note: If you do not have it, you must get the IBM Java Runtime Environment package where the **ikeyman** tool is included.
2. Start **ikeyman.exe** for Windows operating systems or **ikeyman** for UNIX and Linux operating systems.

3. From the **Key Database File** menu, select **Open**.

4. Access the keystore file that was created previously: *AGENT_INSTALLED_DIR*/
   Agents/keys/agentkeys.p12.

5. Enter the keystore password. The default password is secret.

6. Select **Create** > **New Self Signed certificate**.

7. Set the Key Label to **agentAdapter**.

8. Use your system name (DNS name) as the Common Name (workstation
   name).

9. Enter the name of your organization.

10. Click **OK**.

Installing the certificate and key from a PKCS12 file

If the certTool utility did not generate a CSR to obtain a certificate, you must
install both the certificate and private key.

About this task

Store the certificate and private key in a PKCS12 file. The CA sends a PKCS12 file
that has a .pfx extension. The file might be a password-protected file and it
includes both the certificate and private key.

Procedure

1. Copy the PKCS12 file to the bin directory of the adapter. For Windows
   operating systems, *AGENT_INSTALLED_DIR*\Agents\adapter_nameAgent\bin

2. At the **Main Menu** prompt for the certTool, type C to display the following
   prompt:

   ```
   Enter name of PKCS12 file:
   ------------------------------------------------------------------------
   ```

3. At **Enter name of PKCS12 file**, type the name of the PKCS12 file that has the
   certificate and private key information and press **Enter**. For example,
   agentAdapter.p12.

4. At **Enter password**, type the password to access the file and press **Enter**.

Results

After you install the certificate and private key in the adapter registry, the certTool
displays **Main Menu**.

Extracting a CA certificate for the adapter

Use a CA certificate to verify the origin of a signed digital certificate.

About this task

When an application receives a signed certificate from another application, it uses a
CA certificate to verify the originator of the certificate. You can configure many
applications. For example, you can configure web browsers with the CA certificates
of well-known certificate authorities. This type of configuration can eliminate or
reduce the task of distributing CA certificates across the security zones in a
network.

Procedure

1. Access the directory where the **ikeyman** tool is located. For example, *IBM_SOFTWARE_INSTALLED_DIR*\Java\jre\bin.

   Note: If you do not have it, you must get the IBM Java Runtime Environment package where the **ikeyman** tool is included.

2. Start **ikeyman.exe** for Windows operating systems or **ikeyman** for UNIX and Linux operating systems.

3. From the **Key Database File** menu, select **Open**.

4. Access the keystore file that was created previously: *AGENT_INSTALLED_DIR*/Agents/keys/agentkeys.p12.

5. Enter the keystore password. The default password is secret.

6. Extract the server certificate for client use by selecting **Extract Certificate**.

7. Select **Binary DER data** as the data type.

8. Enter the certificate file name: agentAdapter.der.

9. Enter the location as *AGENT_INSTALLED_DIR*/Agents/keys.

10. Click **OK**.

11. Copy the agentAdapter.der certificate file to the workstation on which the Identity Brokerage is installed.

Importing the Identity Brokerage CA certificate in the virtual appliance

After you create a CA certificate for the adapter, you must import the adapter CA certificate in the virtual appliance where the Identity Brokerage is installed.

Procedure

1. Copy the CA certificate file, agentAdapter.der, to the C:\keys directory on the workstation on which Identity Brokerage is installed.

2. Log in to the Virtual Appliance Dashboard.

3. Select **Configure** > **Certificates** .

4. Choose trust and click **Edit**.

5. Select **Signer** and click **Upload**.

   a. Define the **Label**, which is the alias of the certificate.

   b. Browse for the file name of the exported IBM Security Directory Integrator server certificate: C:\keys\agentAdapter.der. When you open the certificate file, the type and name of the certificate are displayed.

   c. Click **Save**.

Results

The updated list of certificates is displayed.

Exporting the virtual appliance CA certificate for Identity Brokerage

To establish a secure communication between Identity Brokerage and the adapter, export the virtual appliance CA certificate for Identity Brokerage.

Procedure

1. Log in to the Virtual Appliance Dashboard.

2. Access **Configure** > **Certificates**.

3. Select **key** and click **Export**.
4. Convert **.pem** to **.der** type with an external tool.
5. Rename the certificate file. For example, `ibclient.der`.
6. Save the file in a temporary folder. IBM Security Directory Integrator uses this file when you import the certificate.

Importing the virtual appliance CA certificate in the adapter registry

Procedure
1. Start the certTool utility.
   a. Open the **Command Line** tool and go to the *AGENT_INSALLED_DIR*\Agents\ adapter_nameAgent\bin\ directory.
   b. Run the command **CertTool -agent adapter_nameAgent**.

      adapter_nameAgent is the name of the adapter agent. For example, `NotesAgent` or `WinLocalAgent`
2. Install a CA certificate in the adapter registry.
   a. Copy the certificate file to the bin directory of the adapter. For Windows operating systems,*AGENT_INSALLED_DIR*\Agents\adapter_nameAgent\bin
   b. At the **Main Menu** prompt for the certTool, type F to display the following prompt:

      `Install a CA Certificate:`
   c. At **Enter name of certificate file**, type the name of the CA certificate file that is extracted from the virtual appliance and press **Enter**. For example, `ibclient.der`.

Results

After you install the certificate in the adapter registry, the **certTool** displays the **Main Menu.**

Start, stop, and restart the adapter service

When you edit the adapter properties file, you must stop and restart the adapter service for the changes to take effect.

Select the appropriate method based on your operating system.

For AIX®, HP-UX, Linux, or Solaris operating systems
1. From the command line, access the directory that contains the ITIMAd script file.

   Note:
   - For AIX and HP-UX, the adapter installer copies the ITIMAd script file to the `timsol` directory.
   - For Linux or Solaris, the adapter installer automatically copies the ITIMAd script file to the `/etc/init.d/` directory when the adapter is installed.
2. Run the following commands:

Table 5. UNIX based and Linux commands

	AIX	**HP-UX**	**Linux and Solaris**
To start the adapter service	`ITIMAd startsrc`	`ITIMAd start`	`ITIMAd start`

Table 5. UNIX based and Linux commands (continued)

	AIX	HP-UX	Linux and Solaris
To stop the adapter service	`ITIMAd stopsrc`	`ITIMAd stop`	`ITIMAd stop`
To restart the adapter service	`ITIMAd restartsrc`	`ITIMAd restart`	`ITIMAd restart`

Note: On the Solaris operating system, the ITIMAd script file creates the `itimadpid` file in the adapter directory. This file is not created on other operating systems.

- The file contains the process ID of the adapter service. Do not modify or delete this file.
- When you start the adapter service, the ITIMAd script creates the `itimadpid` file.
- When you stop the dispatcher service, the ITIMAd script deletes the `itimadpid` file.

For Windows® operating systems

1. In the Control Panel, select **Administrative Tools** > **Services**.
2. In the Services window, you can start and stop the adapter service. The service name is `IBM Tivoli Directory Integrator (TIM Adapters)`.

For Linux for System z or z/OS operating systems

1. Access the `timsol` directory.
2. Run the following commands:

Table 6. Linux for System z and z/OS commands

	Linux for System z	z/OS
To start the adapter	`% ./ITIMAd start`	`% ./ITIMAd start`
To verify whether the `ibmdisrv` or the `ibmdisrv_ascii` process is running	`% ps -ef | grep ibmdisrv`	`% ps -ef | grep ibmdisrv_ascii`
To stop the adapter	`% ./ITIMAd stop`	`% ./ITIMAd stop`
To verify whether the `ibmdisrv` or `ibmdisrv_ascii` process is not running	`% ps -ef | grep ibmdisrv`	`% ps -ef | grep ibmdisrv_ascii`

Chapter 9. Secure authentication for RMI-based adapters

You can configure the RMI-based adapters for one-way or two-way secure authentication with signed certificates.

Note: The SSL configuration applies only to those adapters that are installed with an external IBM Security Directory Integrator.

Configuring certificates for one-way SSL authentication

Use one-way SSL communication when the client must authenticate the server.

About this task

One-way SSL communication on the IBM Security Directory Integrator server does not require the truststore. However, you must configure the truststore for the Remote Method Invocation (RMI) SSL initialization to succeed.

Procedure

1. Create a keystore for the IBM Security Directory Integrator server. See "Creating a keystore for the IBM Security Directory Integrator server" on page 42.

2. Optional: Create a truststore for the IBM Security Directory Integrator server. See "Creating a truststore for the IBM Security Directory Integrator server" on page 43.

   Note: Do not perform this task if you use the same file for keystore and truststore.

3. Create a self-signed certificate for the IBM Security Directory Integrator server. See "Creating a self-signed certificate for the IBM Security Directory Integrator server" on page 44.

4. Create a CA certificate for the IBM Security Directory Integrator server. See "Extracting a CA certificate for the IBM Security Directory Integrator" on page 44.

5. Import the CA certificate that is extracted from the IBM Security Directory Integrator into the virtual appliance where the Identity Brokerage is installed. See "Importing the Identity Brokerage CA certificate in the virtual appliance" on page 45.

6. Configure the IBM Security Directory Integrator to use keystores. See "Configuring the IBM Security Directory Integrator to use the keystores" on page 45.

   Note: You can modify the `solution.properties` file for steps 6, 7, and 8 in a single operation. When you do so, do not stop and restart the adapter service at the end of steps 6 and 7.

7. Configure the IBM Security Directory Integrator to use truststores. See "Configuring the IBM Security Directory Integrator to use the truststores" on page 46

8. Enable the adapter service to use SSL. See "Enabling the adapter service to use SSL" on page 46

9. Stop and restart the adapter service. See "Start, stop, and restart the adapter service" on page 38

10. Restart the virtual appliance.

Configuring certificates for two-way SSL authentication

Use two-way SSL communication when the client must authenticate the server.

Procedure

1. Create a keystore for the IBM Security Directory Integrator server. See "Creating a keystore for the IBM Security Directory Integrator server."

2. Optional: Create a truststore for the IBM Security Directory Integrator server. See "Creating a truststore for the IBM Security Directory Integrator server" on page 43.

   Note: Do not perform this task if you use the same file for keystore and truststore.

3. Create a self-signed certificate for the IBM Security Directory Integrator server. See "Creating a self-signed certificate for the IBM Security Directory Integrator server" on page 44.

4. Create a CA certificate for the IBM Security Directory Integrator server. See "Extracting a CA certificate for the IBM Security Directory Integrator" on page 44.

5. Import the CA certificate that is extracted from the IBM Security Directory Integrator into the virtual appliance where the Identity Brokerage is installed. See "Importing the Identity Brokerage CA certificate in the virtual appliance" on page 45.

6. Configure the IBM Security Directory Integrator to use keystores. See "Configuring the IBM Security Directory Integrator to use the keystores" on page 45.

   Note: You can modify the `solution.properties` file for steps 6, 7, and 8 in a single operation. When you do so, do not stop and restart the adapter service at the end of steps 6 and 7.

7. Configure the IBM Security Directory Integrator to use truststores. See "Configuring the IBM Security Directory Integrator to use the truststores" on page 46

8. Enable the adapter service to use SSL. See "Enabling the adapter service to use SSL" on page 46

9. Create a CA certificate for the Identity Brokerage. See "Exporting the virtual appliance CA certificate for Identity Brokerage" on page 47.

10. Import the virtual appliance CA certificate in the IBM Security Directory Integrator truststore. See "Importing the virtual appliance CA certificate in the IBM Security Directory Integrator truststore" on page 47.

11. Stop and restart the adapter service. See "Start, stop, and restart the adapter service" on page 38

12. Restart the virtual appliance.

Creating a keystore for the IBM Security Directory Integrator server

You must create a keystore to hold the certificates that the SSL server uses to authenticate with clients.

About this task

A keystore is a database of private keys and the associated certificates that authenticate the corresponding public keys. Digital certificates are stored in a keystore file. A keystore also manages certificates from trusted entities.

Procedure

1. Access the *ITDI_HOME*\jvm\jre\bin directory.
2. Start **ikeyman.exe** for Windows operating systems or **ikeyman** for UNIX and Linux operating systems.
3. From the **Key Database File** menu, select **New**.
4. Select the key database type of **JKS**.
5. Enter the keystore file name. For example, tdikeys.jks.
6. Enter the location. For example, *ITDI_HOME*/keys.

   Note: Ensure that location that you specify exists.
7. Click **OK**.
8. Type a password for the keystore. The default password is secret.
9. Click **OK**.

Creating a truststore for the IBM Security Directory Integrator server

You must create a truststore on the SSL server to hold trusted certificates so that clients can authenticate with the server.

About this task

A truststore is a database of public keys for target servers. The SSL truststore contains the list of signer certificates (CA certificates) that define which certificates the SSL protocol trusts. Only a certificate from one of these listed trusted signers can be accepted. Do not do the following task if you use the same file for keystore and truststore.

Procedure

1. Access the *ITDI_HOME*\jvm\jre\bin directory.
2. Start **ikeyman.exe** for Windows operating systems or **ikeyman** for UNIX and Linux operating systems.
3. From the **Key Database File** menu, select **New**.
4. Select the key database type of **JKS**.
5. Type the keystore file name. For example, tdikeys.jks.
6. Type the location. For example, *ITDI_HOME*/keys.

   Note: Ensure that location that you specify exists.
7. Click **OK**.
8. Type a password for the keystore. The default password is secret.
9. Click **OK**.

Creating a self-signed certificate for the IBM Security Directory Integrator server

A self-signed certificate contains information about the owner of the certificate and the signature of the owner. This type of certificate is typically used in a testing environment.

About this task

A self-signed certificate is a signed certificate and also a CA certificate. To use self-signed certificates, you must extract the CA certificate from the self-signed certificate to configure SSL. See "Extracting a CA certificate for the IBM Security Directory Integrator."

You can purchase a certificate from a well-known authority, such as VeriSign. You can also use a certificate server, such as the one included with the Microsoft Windows 2003 Advanced Server, to generate your own certificates.

Procedure

1. Access the *ITDI_HOME*\jvm\jre\bin directory.
2. Start **ikeyman.exe** for Windows operating systems or **ikeyman** for UNIX and Linux operating systems.
3. From the **Key Database File** menu, select **New**.
4. Access the keystore file that was created previously: *ITDI_HOME*/keys/tdikeys.jks.
5. Enter the keystore password. The default password is secret.
6. Select **Create** > **New Self Signed certificate**.
7. Set the Key Label to **tdiserver**.
8. Use your system name (DNS name) as the Common Name (workstation name).
9. Enter the name of your organization.
10. Click **OK**.

Extracting a CA certificate for the IBM Security Directory Integrator

Use a CA certificate to verify the origin of a signed digital certificate.

About this task

When an application receives a signed certificate from another application, it uses a CA certificate to verify the originator of the certificate. You can configure many applications. For example, you can configure web browsers with the CA certificates of well-known certificate authorities. This type of configuration can eliminate or reduce the task of distributing CA certificates across the security zones in a network.

Procedure

1. Access the *ITDI_HOME*\jvm\jre\bin directory.
2. Start **ikeyman.exe** for Windows operating systems or **ikeyman** for UNIX and Linux operating systems.
3. From the **Key Database File** menu, select **New**.

4. Access the keystore file that was created previously: *ITDI_HOME*/keys/
   `tdikeys.jks`.
5. Enter the keystore password. The default password is `secret`.
6. Extract the Server certificate for client use by selecting **Extract Certificate**.
7. Select **Binary DER data** as the data type.
8. Enter the certificate file name: `idiserver.der`.
9. Enter the location as *ITDI_HOME*\keys.
10. Click **OK**.
11. Copy the `idiserver.der` certificate file to the workstation on which the
    Identity Governance and Intelligence is installed.

Importing the Identity Brokerage CA certificate in the virtual appliance

After you create a CA certificate for the Identity Brokerage, you must import the
Identity Brokerage CA certificate in the virtual appliance where the Identity
Brokerage is installed.

Procedure

1. Copy the SSL server CA certificate file, `idiserver.der`, to the `C:\keys` directory
   on the workstation that communicates with the Identity Brokerage adapters.
2. Log in to the Virtual Appliance Dashboard.
3. Select **Configure** > **Certificates**.
4. Select **Identity Governance and Intelligence key store** and click **Edit**.
5. Select **Signer** and click **Upload**.
   a. Define the **Label**, which is the alias of the certificate.
   b. Browse for the file name of the exported IBM Security Directory Integrator
      server certificate: `C:\keys\idiserver.der`. When you open the certificate file,
      the type and name of the certificate are displayed.
   c. Click **Save**.

Results

The updated list of certificates is displayed.

Configuring the IBM Security Directory Integrator to use the keystores

You can configure the IBM Security Directory Integrator to use the keystores.

Procedure

1. Access the *ITDI_HOME*\timsol directory.
2. Open the IBM Security Directory Integrator `solution.properties` file in an
   editor.
3. Edit the following lines under **client authentication**:
   ```
   javax.net.ssl.keyStore=ITDI_HOME\keys\tdikeys.jks
   {protect}-javax.net.ssl.keyStorePassword=secret
   javax.net.ssl.keyStoreType=JKS
   ```
   a. Remove the comment, if necessary.
   b. Set the location, password, and type of keystore to match the keystore you
      created in "Creating a keystore for the IBM Security Directory Integrator
      server" on page 42.
4. Save your changes.

5. Stop and restart the adapter service. See "Start, stop, and restart the adapter service" on page 38.

   Note: You can modify the `solution.properties` file in a single operation. Do not stop and restart the adapter service after you configure the IBM Security Directory Integrator to use the keystores and truststores. You can stop and restart the adapter after you enable the adapter service to use SSL. See "Enabling the adapter service to use SSL."

Configuring the IBM Security Directory Integrator to use the truststores

To configure IBM Security Directory Integrator to use the truststores, take these steps:

Procedure

1. Access the *ITDI_HOME*\timsol directory.
2. Open the IBM Security Directory Integrator `solution.properties` file in an editor.
3. Edit the following lines under **client authentication**:

   ```
   javax.net.ssl.trustStore=ITDI_HOME\keys\tditrust.jks
   {protect}-javax.net.ssl.trustStorePassword=secret
   javax.net.ssl.trustStoreType=JKS
   ```

   a. Remove the comment, if necessary.
   b. Set the location, password, and type of truststore to match the truststore you created in "Creating a truststore for the IBM Security Directory Integrator server" on page 43.
4. Save your changes.
5. Stop and restart the adapter service. See "Start, stop, and restart the adapter service" on page 38.

   Note: You can modify the `solution.properties` file in a single operation. Do not stop and restart the adapter service after you configure the IBM Security Directory Integrator to use the keystores and truststores. You can stop and restart the adapter after you enable the adapter service to use SSL. See "Enabling the adapter service to use SSL."

Enabling the adapter service to use SSL

You can enable the adapter service to use SSL.

Procedure

1. Access the *ITDI_HOME*\timsol directory.
2. Open the IBM Security Directory Integrator `solution.properties` file in an editor.
3. Edit the following lines based on the preferred type of secure communications.

   For no SSL
   ```
   com.ibm.di.dispatcher.ssl=false
   com.ibm.di.dispatcher.ssl.clientAuth=false
   ```

   For one-way SSL
   ```
   com.ibm.di.dispatcher.ssl=true
   com.ibm.di.dispatcher.ssl.clientAuth=false
   ```

For two-way SSL
```
com.ibm.di.dispatcher.ssl=true
com.ibm.di.dispatcher.ssl.clientAuth=true
```

4. Save your changes.

5. Stop and restart the adapter service. See "Start, stop, and restart the adapter service" on page 38.

Exporting the virtual appliance CA certificate for Identity Brokerage

To establish a secure communication between Identity Brokerage and the adapter, export the virtual appliance CA certificate for Identity Brokerage.

Procedure

1. Log in to the Virtual Appliance Dashboard.
2. Select **Configure** > **Certificates**.
3. Select **key** and click **Export**.
4. Convert `.pem` to `.der` type with an external tool.
5. Rename the certificate file. For example, `ibclient.der`.
6. Save the file in a temporary folder. IBM Security Directory Integrator uses this file when you import the certificate.

Importing the virtual appliance CA certificate in the IBM Security Directory Integrator truststore

IBM Security Identity Governance and Intelligence uses the virtual appliance CA certificate to authenticate to the IBM Security Directory Integrator server.

About this task

After you export the virtual appliance CA certificate, you must import it into the IBM Security Directory Integrator server truststore. After it is stored in the truststore, the SSL server can recognize the credentials of the client and authenticate the client.

Procedure

1. Copy the SSL Client CA certificate file, `ibclient.der`, that you created in "Exporting the virtual appliance CA certificate for Identity Brokerage." Put the copied file in the *ITDI_HOME*\keys directory on the workstation where the IBM Security Directory Integrator is installed.
2. Access the *ITDI_HOME*\jvm\jre\bin directory.
3. Start the **ikeyman.exe** file (for Windows operating systems) or **ikeyman** (for UNIX and Linux operating systems).
4. From the **Key Database File** menu, select **Open**.
5. Select the key database type of **JKS**.
6. Type the keystore file name: tditrust.jks.
7. Type the location: *ITDI_HOME*\keys and click **OK**.
8. Select **Signer Certificates** from the list and click **Add**.
9. Select **Binary DER data** as the data type.
10. Use **Browse** to select the ibclient.der file that is stored in *ITDI_HOME*\keys directory.
11. Use ibclient as the label.

12. Click **OK** to continue.

Start, stop, and restart the adapter service

When you edit the adapter properties file, you must stop and restart the adapter service for the changes to take effect.

Select the appropriate method based on your operating system.

For AIX®, HP-UX, Linux, or Solaris operating systems

1. From the command line, access the directory that contains the ITIMAd script file.

   Note:
   - For AIX and HP-UX, the adapter installer copies the ITIMAd script file to the timsol directory.
   - For Linux or Solaris, the adapter installer automatically copies the ITIMAd script file to the /etc/init.d/ directory when the adapter is installed.
2. Run the following commands:

Table 7. UNIX based and Linux commands

	AIX	**HP-UX**	**Linux and Solaris**
To start the adapter service	`ITIMAd startsrc`	`ITIMAd start`	`ITIMAd start`
To stop the adapter service	`ITIMAd stopsrc`	`ITIMAd stop`	`ITIMAd stop`
To restart the adapter service	`ITIMAd restartsrc`	`ITIMAd restart`	`ITIMAd restart`

Note: On the Solaris operating system, the ITIMAd script file creates the itimadpid file in the adapter directory. This file is not created on other operating systems.
- The file contains the process ID of the adapter service. Do not modify or delete this file.
- When you start the adapter service, the ITIMAd script creates the itimadpid file.
- When you stop the dispatcher service, the ITIMAd script deletes the itimadpid file.

For Windows® operating systems

1. In the Control Panel, select **Administrative Tools** > **Services**.
2. In the Services window, you can start and stop the adapter service. The service name is IBM Tivoli Directory Integrator (TIM Adapters).

For Linux for System z or z/OS operating systems

1. Access the timsol directory.
2. Run the following commands:

Table 8. Linux for System z and z/OS commands

	Linux for System z	**z/OS**
To start the adapter	`% ./ITIMAd start`	`% ./ITIMAd start`

Table 8. Linux for System z and z/OS commands (continued)

	Linux for System z	**z/OS**
To verify whether the `ibmdisrv` or the `ibmdisrv_ascii` process is running	`% ps -ef	grep ibmdisrv`	`% ps -ef	grep ibmdisrv_ascii`
To stop the adapter	`% ./ITIMAd stop`	`% ./ITIMAd stop`
To verify whether the `ibmdisrv` or `ibmdisrv_ascii` process is not running	`% ps -ef	grep ibmdisrv`	`% ps -ef	grep ibmdisrv_ascii`

Index

A

adapters
    service, enabling SSL   46
Application Manager dashboard   5
authentication
    one-way SSL configuration   33
    two-way SSL configuration   34

C

certificates
    extracting
        CA for Tivoli Directory
        Integrator   36, 44
      origin verification   36, 44
      self-signed   35, 44
client authentication   34
communication
    SSL one-way   41
    SSL two-way   42
configuration
    one-way SSL authentication   33
configure
    Security Identity Governance
      server   9
configuring
    keystores, Security Directory
      Integrator   45
    Security Directory Integrator
      for keystores   45
      for truststores   46
    truststores, configuring Security
      Directory Integrator   46

D

Dashboards
    configuring for Service Center   1
    for Application Manager persona   5
    for Employee persona   2
    for User Manager persona   4
    with business activity scope   7

E

external authentication   9

I

ISAM   23, 25
iToken
    single sign-on   27

K

keystore
    creating   43
    directory integrator usage   43

keystore *(continued)*
    server authentication to clients   43

L

LTPA   23, 25

O

OIDC   25
one-way SSL authentication
    certificate validation   33
    configuration   33
OpenID Connect   23

P

passwords
    protected file, see PKCS12 file   36
personas
    Application Manager   5
    Employee   2
    User Manager   4
PKCS12 file
    certificate and key installation   36
protocol
    SSL
      certificate management   43
      client authentication   43
      keystore   43
      truststore   43
      two-way configuration   34

Q

Quick Insights
    configuring for   1
    dashboard items for   1
    for Employee persona   2
    for User Manager persona   4

R

restarting
    services   38, 48
reverse proxy   9

S

self-signed certificates   35, 44
server
    adapter
      communication with the
        server   34
      SSL communication   34
service
    SSL, enabling for adapter   46

Service Center
    configuring dashboards for   1
single sign-on
    configuring   25
    iToken   27
    protocols   23
SSL
    adapter service, enabling   46
    creating a keystore   43
    creating truststores   43
    one-way communication   41, 42
    two-way configuration   34
SSL certificates
    self-signed   35, 44
SSO   23
starting
    services   38, 48
stopping
    services   38, 48

T

truststores
    client authentication to server   43
    creating   43
two-way configuration
    certificate and private key   34
    SSL
      client   34

51

IBM ®

Printed in USA