Application Configuration

Version: 8.4

Revised: September 2023

Copyright and Trademark Notices

Copyright © 2023 SailPoint Technologies, Inc. All Rights Reserved.

Contents

About Applications and Connectors

IdentityIQ uses configured applications to connect to your enterprise's target systems and read data from those applications into IdentityIQ. This data is used for creating/updating Identity Cubes, building IdentityIQ's entitlement catalog, and bringing information about users' accounts and group membership into IdentityIQ. Many, but not all, applications can also write information back to target systems, to reflect changes made within IdentityIQ.

Although the terms "application" and "connector" sometimes seem to be used interchangeably, it's important to understand the differences:

Connectors are how IdentityIQ provides the base functionality for communication with the target system. Connectors are in essence templates for connecting to target systems. They define how connectivity to the target system is accomplished, provide the ability to connect to a source, to read from and (in some cases) write to the source, and to create a normalized resource object representing the data read from the source.

Applications are configured instances of connectors. They contain specific information for connecting to specific target systems, such as a unique name, connection parameters for connecting to a particular system, and a specific account schema. For example, you may need to aggregate data from several different delimited files. Using the delimited file connector, you can create many unique delimited file applications, each pointing to a particular file, and using a specific schema that defines the data to be aggregated from the file.

Documentation For Specific Connectors

This guide provides general concepts and common practices that are applicable to all connectors. For more detailed information about specific connectors, use SailPoint's connector-specific guides. These guides are available online and in PDF format, and provide details such as supported features, connection parameters, account schema details, and more, for each connector.

Online connector documentation: documentation for the current version of each of the connectors supplied by SailPoint is available online on SailPoint's documentation portal. Guides for IdentityIQ connectors are located here.

PDF and legacy connector documentation: you can download PDF versions of the connector guides from SailPoint's Compass Community. Connector guide PDFs are updated with each patch to IdentityIQ. If you need connector documentation for older versions of IdentityIQ, use the Supported Connectors for IdentityIQ page on Compass; be sure to locate the correct documentation zip file for your specific version and patch.

Additional Resources for Connectors

These other SailPoint resources can be useful sources of additional information on connectors:

- Compass – in SailPoint's Compass Community, you can engage with peers and experts to ask questions and share answers, submit ideas, read wiki articles and technical white papers, watch webinars, and more.
- SailPoint's Customer Support knowledge base – you can search here for technical notes about specific connectors, from SailPoint's Customer Support team.

Application Concepts

Here are some general concepts you should be familiar with as part of planning your application onboarding and configuring your applications.

Aggregation

Aggregation is the process wherein information that resides on the target system is discovered and read in to IdentityIQ. The specific information that is brought into IdentityIQ through aggregation will vary depending on the type and purpose of the target system, what data is available in the target system, and how the application is configured.

For example, when reading from a system of record such as an Human Resources system, you might want to aggregate a lot of information about each user, such as full name, department, job title, manager, and location. For a business-specific system such as an Accounts Payable system, you may aggregate only a limited set of information, such as which groups your users belong to, and which features each user has access to.

While the details of what is aggregated is largely defined by your application configuration, the actual work of aggregation is accomplished by the **Account Aggregation** and **Account Group Aggregation** tasks. These tasks have many configuration options that allow you to fine-tune the behavior of the aggregation for each application. For more information, see:

- **Tasks overview** in the *IdentityIQ Tasks Guide*
- **Account Aggregation** in the *IdentityIQ Tasks Guide*
- **Account Group Aggregation** in the *IdentityIQ Tasks Guide*

Correlation

Correlation is the process of matching aggregated accounts to existing identities. When you aggregate from an authoritative source, IdentityIQ creates an Identity Cube for each account on that source. When you aggregate from a non-authoritative source, correlation logic helps match aggregated accounts to an authoritative Identity Cube.

Correlation logic can be implemented several ways:

- Through **direct mapping** of attributes – for example, the application's account attribute "mail" is mapped directly to the identity's attribute "email." Attribute mapping can be defined both in the application configuration feature and in the Rapid Setup feature.
- Using **conditions** that assign application accounts to existing identities by defining attribute conditions. For example, the root account on Unix typically does not have any identifying attributes that can help when trying to correlate it to an existing identity using direct attribute mapping, so you can use a condition, such as whether

the identity is a Unix application owner, to drive the correlation. Condition-based correlation can only be defined in the application configuration feature. The correlation configurations you define based on conditions can be saved and shared across multiple applications.

- Through **rules** – custom BeanShell rules let you create your own specialized logic for correlation. Rules for account and manager correlation are specified in the **Rules** tab of the application. Rules can be built outside of IdentityIQ and imported as XML objects, or can be defined using the rule editor that can be opened in the application configuration Rules tab. For more information on working with rules, see the **Rules and Scripts in IdentityIQ** section of the *IdentityIQ System Administration Guide*.

Resolving Uncorrelated Accounts

The correlation logic you define in the application does not always successfully match newly-aggregated accounts to Identity Cubes. For example, accounts belonging to employees who are no longer part of your organization, or mismatched accounts that use variations on an employee's name, can result in accounts without a clear association to an existing identity.

The **Identity Correlation** feature lets you manually correlate these accounts to specific identities. Manual correlations are permanently retained; IdentityIQ does not change these correlations during subsequent aggregations. If a manual correlation has been made in error, it can be changed in the Identity Warehouse on the relevant Identity Cube's **Application Account** tab. For more information, see the Identity Correlation section of the *IdentityIQ Identity Management Guide*.

Provisioning

Provisioning is the process of writing information (that is, the attributes and values you want to update) back to your target system, based on changes made within IdentityIQ. For example, an access request managed in IdentityIQ may create a new account for the user in a target system, or a policy violation may result in a user having certain access on the target system revoked. Provisioning propagates these changes to target systems, using a Provisioning Policy as the instruction or a template for what to include as part of provisioning; this includes the creation of new accounts when new access is granted on a target system where a user does not already have an account.

Each application can have its own configurations that specify which attributes to include in provisioning, and how to set their values.

You can configure distinct provisioning policies for different actions, such as creating accounts, updating accounts, and deleting accounts.

Only some connectors support provisioning. For more information about specific connectors, refer to the IdentityIQ Connectors documentation on SailPoint's documentation portal.

For more information, see the *IdentityIQ Provisioning Guide*.

Attribute Synchronization

Attribute synchronization is an automated process of pushing changes in identity attributes (such as name, email, or department) to target systems. These identity attribute changes often come from aggregation from an authoritative source, but they also might come from edits made to identities directly in IdentityIQ.

Attribute synchronization is configured globally for IdentityIQ, in the **gear menu > Global Settings > Identity Mappings UI**.

When you set your identity mappings, you will select which configured application(s) are updated, and which attributes (defined in the configured application schema) store the values to be synchronized.

When an attribute change comes through aggregation, attribute synchronization is initiated through a refresh task that has the Synchronize Attributes option selected. See the Identity Refresh task for information about configuring and running this task.

For more information, see **Attribute Synchronization** in the *IdentityIQ Provisioning Guide*.

Rapid Setup

Rapid Setup provides an alternative way of defining application behavior and the events and processes for managing identities, in user-friendly, guided UI.

Rapid Setup lets you separate the technical and IT-centric steps of onboarding and configuring applications (such as defining connection parameters and schemas) from the business-centric steps of defining the business processes the application should follow (such as how to onboard applications and handle common identity management scenarios such as joiner, mover, leaver, and terminating identities). It provides preconfigured processes that follow best practices for managing identities.

Rapid Setup also provides a simple UI for defining attribute-based account and manager correlation.

For more information, see the *IdentityIQ Rapid Setup* guide.

Configuring an Application

An application in IdentityIQ an instance of a configured IdentityIQ connector, that includes all of the configuration and connection details required to connect IdentityIQ to a third party application for the purposes of governing and provisioning access to these applications. Applications can be read-only, or read-write.

There are two types of connector in IdentityIQ: application-type connectors that collect account information, and activity-type connectors that collect activity information. IdentityIQ uses the information from both types to maintain Identity Cubes.

To configure applications, from the menu bar, click **Applications > Application Definition**.

Application List Page

The Application List page displays all of the applications currently configured. The Application List contains the following information:

Column	Description
Name	The name of the application.
Host	Host where the application resides.
Type	The application type, such as LDAP or JDBC.
Modified	The date when the application was last modified.

Click on an existing application or click **New Application** to open the Edit Application page. See Using the Edit Application Page.

Common Application Configuration Details

The Details tab for each application includes options that are common to all applications.

Application Type

The Application Type list contains all the connectors that are enabled for your instance of IdentityIQ. Many of the application types are specific to particular target systems, such as Workday or Box. Others are more general-use types that can be used on various systems or sources, such as JDBC and SCIM 2.0.

The application type determines which configuration options appear in the application configuration UI. Some options are unique to the selected application type.

Note that some connectors may not be enabled by default in your instance and so may not appear in the application type list; if you do not see an application type listed here that you think you should have access to, contact your Customer Success Manager or Professional Services resource.

For more information about specific connectors, refer to the IdentityIQ Connectors documentation on SailPoint's documentation portal.

Owner

The owner specified here is typically the business owner of the system, who is familiar with the system, its entitlements, and who should have access it it. The owner responsible for access reviews for this application from application owner certifications, account group permissions certifications, and account group membership certifications. The application owner can also be selected as an approver in targeted certifications.

By default, the application owner is also responsible for approving access requests for entitlements associated with this application, although this behavior can be modified in the business process that manages access requests.

> **Tip:**
> An application owner can be either an individual or a workgroup; it is a best practice to set a workgroup as the owner of the application. Assigning ownership to a workgroup makes it possible for multiple people to share responsibility for managing application-related activities that must be performed by someone, but do not necessarily need to be owned or performed by a specific person. See **Workgroups** in the *Roles, Groups, and Populations* guide.

Revoker

If you want to designate a user or workgroup that is responsible specifically for handling requests to revoke access (which typically occur as a result of a certification decision or policy violation), enter that identity or workgroup here. If no user is specified in this field, all revocation requests are assigned to the application owner by default.

Authoritative Application

An authoritative application is a target system that represents *the primary and most trusted information* for employee information for your enterprise, such as a human resources application. When this option is selected, designating this application as an authoritative one, the IdentityIQ aggregation process will create an authoritative Identity Cube for each account in this application.

For applications that are not marked as authoritative, IdentityIQ will attempt to correlate aggregated accounts to existing authoritative identities.

Your organization can have multiple target systems designated as authoritative sources.

For more information, see the **Authoritative and Non-Authoritative Data Sources** section in the *IdentityIQ Application Management Guide*.

Using the Edit Application Page

Use the **Edit Application** page to configure applications for your enterprise.

> Note: Do not open multiple tabs or browsers. Making and saving a change in one tab might overwrite changes made in another.

The Edit Application page opens to the **Details** page and includes these tabs.

> Note: Some tabs only appear after an application has been defined and configured.

- Configuration Tab
- Correlation Tab
- Accounts Tab
- Risk Tab
- Activity Data Sources Tab
- Unstructured Targets Tab
- Rules Tab
- Password Policy Tab
- Tiers Tab. This tab is available only for Logical application types. See detailed information about configuring Logical applications in the IdentityIQ connectors documentation.

The information on each tab is determined by the type of application specified on the **Application Type** dropdown list. Use these tabs to define how each application interacts with IdentityIQ.

The **Edit Application** page also lists any extended attributes that were configured for your deployment of IdentityIQ; these are displayed at the bottom of the tab.

For each application enter or edit the following information:

Edit Application Details Fields

Name

The name of the application. This is the named used to identify the application throughout IdentityIQ.

> Note: IdentityIQ does not support applications names that start with a numeric value or that are longer than 31 characters.

Owner

The owner specified here is typically the business owner of the system, who is familiar with the system, its entitlements, and who should have access it it. The owner responsible for access reviews for this application from application owner certifications, account group permissions certifications, and account group membership certifications. The application owner can also be selected as an approver in targeted certifications.

Application ownership can be assigned to an individual identity or a workgroup. If the application ownership is assigned to a workgroup, all members share certification responsibilities, are assigned certification request associated with the application and all can take action on those requests.

Application Type

The application type, for example, LDAP or JDBC.
The **Application Type** dropdown list contains the types of application to which IdentityIQ can connect. This list can grow and change as new application types are developed for IdentityIQ, and as existing application types are updated or deprecated.

Description

A brief description of the application.

If your instance of IdentityIQ is configured to support multiple languages, you can use the language selector to enter description in multiple languages. The dropdown list displays any languages supported by your instance of IdentityIQ. The description displayed throughout the product is dependent on the language associated with the user's browser. If only one description is entered, that is the description used by default.

> Note: You must Save the description before changing languages to enter another description.

Revoker

The default IdentityIQ user or workgroup to be assigned revocation requests associated with entitlements on this application. If no user is specified in this field, all revocation requests are assigned to the to application owner by default.

Proxy Application

A proxy application is an application that handles the processing (aggregation and provisioning) on behalf of your application. Configuring a proxy application is optional.

Here are some examples of proxy applications:

- Multiplex applications: In this case you define an application and, most often, a build map rule that sorts the data out in multiple sub-applications. In that case, the sub-applications have the main application as the proxy.
- Similar to the multiplex applications are the connectors for legacy identity management systems such as, BMC, Novell/NetIQ, IBM Tivoli, and Sun/Oracle Waveset.
- The Cloud Gateway connector tunnels all aggregation and provisioning requests to the gateway in another network. The gateway then acts on behalf of IdentityIQ. All applications that live in the remote network need to have the cloud gateway connector set as the proxy.

Profile Class

This is an optional class used to associate this application with a larger set of applications, for role modeling purposes.

For example, you might set a profile class of XYZ on all of the applications where any user that has read account privileges should be assigned the role XYZ Account Reader. You can then create a single profile for that role instead of a separate profile for each instance of the applications. During the correlation process any user with read account privileges on any of the applications with the profile class XYZ is assigned the role XYZ Account Reader.

Scope

This field is only visible if scoping is enabled for your instance of IdentityIQ. When a scope is assigned, only the owner of the application or users that control the designated scope can work with this application.

Objects associated with this application, such as entitlements in a certification request, are visible to a user with any or no controlled scope, but if a new object is being created (for example, a certification schedule), this application does not appear on the select list unless the creator controls the scope assigned.

Depending on configuration settings, objects with no scope assigned might be visible to all users with the correct capabilities.

Authoritative Application

An authoritative application is a target system that represents *the primary and most trusted information* for employee information for your enterprise, such as a human resources application. When this option is selected, designating this application as an authoritative one, the IdentityIQ aggregation process will create an authoritative Identity Cube for each account in this application. Note that your organization can have multiple authoritative applications.

Case Insensitive

Use to cause case insensitive comparisons of account attribute values when evaluating provisioning policy.

Native Change Detection

Select this option if this application should be included when IdentityIQ performs native change detection during aggregation.

For more detailed information about configuring native change detection, see Native Change Detection Configuration.

Native Change Definitions

Native Change Operations

Select which operations are included when detecting native changes. If no operations are selected, native change detection is disabled.

Attributes to Detected

Indicates which attributes are compared when accounts are modified. If the **Entitlement** option is selected, all entitlement attributes are included. If you select **User Defined**, enter the name of the attributes to compare in the **Attribute Names** box.

Maintenance: Take an application offline for maintenance

Maintenance Enabled

When maintenance is enabled, the application is excluded from provisioning and aggregation during the defined maintenance period. See Application Maintenance Windows for more information.

Maintenance Expiration

The date at which the maintenance will end. If no date is defined, this application will be in maintenance indefinitely.

After adding the application information, click **Save** to save your changes and return to the Application List page.

Configuration Tab

The information displayed on the Configuration tab changes depending on the application type specified.

> Note: The terms account group and application object are use interchangeably in this document but have the same meaning. Some application can have multiple application objects. An account group can be the name of one of those objects.

Settings Tab

The **Settings** tab contains the information that IdentityIQ uses to connect and interact with the target system. **Each application type requires different connection information and the fields on this tab are changed accordingly.**

For more information on connection parameters, see Application Connection Parameters.

For more information about specific connectors, refer to the IdentityIQ Connectors documentation on SailPoint's documentation portal.

If your enterprise is going to use partitioning for account aggregations, identity refreshes, and manager certification, you must enable that function here. Each application type requires different partitioning information.

This is also where you enable an application for data merging and delta aggregation.

Enter the information on this tab as required by the application type being configured. Click **Test Connection** to verify the information is correct.

Adding Object Types

For most application types you see account and group object types. Certain application types, however, enable you to create multiple application object types, each with their own schema. These application object types are sometimes referred to as account groups and those term might be used interchangeably in discussion around this feature.

Click at the top of the page to add a new object type. This function is only available for if the application type is associate with a connector that is enable to handle multiple application object types, or multiple schema.

This button is also displayed if you recently upgraded your instance of IdentityIQ and the application type now supports multiple schemas. In that case you must add the supported application object type here and then run the Account Group Aggregation task to import the new information.

Multiple application object types can be directly correlated, for example an application object type is also an attribute in the schema of another, or they can be indirectly associated, for example they are both objects (schemas) in the same application. These objects and their associations are tracked throughout IdentityIQ and appear in place such as reports, policy violations, searches, and certifications.

Credential Cycling

A note is displayed at the top of this tab if the application is configured to use credential cycling. For those applications, the credentials are stored and maintained on a Privileged Access Management (PAM) module, and verification is performed using existing hook points that support the retrieval of passwords from application credential management solutions such as, CyberArk Application Identity Manager (AIM) or BeyondTrust PowerBroker Password Safe. Refer to the *IdentityIQ Privileged Account Management* guide for more information.

> Note: To enable credential cycling, BeyondTrust PowerBroker Password Safe application passwords must be configured in the JSON format:
> `{"bt_user":"MyUserName","bt_password":"MyPasswordValue"}`

Schema Tab

The Schema tab is used to define the attributes for each object type in the application being configured. Use the following fields to define attributes for use with the IdentityIQ application. The field content is dependent on the application being configured.

For more details on configuring schema information, see Application Schemas.

For more information about specific connectors, refer to the IdentityIQ Connectors documentation on SailPoint's documentation portal.

When initially configuring applications, click **Add New Schema Attribute** to define the attributes for each object. Most application types include a default set of schema attributes. For more dynamic application types (JBDC or DelimitedFile), schemas should be defined manually. Click **Edit** to display the Advanced Properties dialog.

> **Important:** A schema attribute name must not duplicate any extended attribute names that have been defined in your IdentityIQ instance. If an schema attribute name matches an extended attribute name, there is a risk that attribute values will not be updated correctly during an aggregation.

The connectors for some application types enable the automatic discovery of the base schema attributes for those applications. For those application types, click **Discover Schema Attributes** to automatically populate your schema tables. After using the automatic discovery function you must designate the Identity Attribute and Display Attribute for the application.

Click **Preview** to test the respective schema configuration. A pop-up sample table displays to indicate a successful configuration. These tables automatically update when you make changes so that you can use this feature before committing your changes. Only one table can be open at one time. Failures result in an error message specifying the point of failure, for example, a file path and name.

> Note: The Preview function does not apply to applications which do not support aggregation.

Fields	Descriptions
Native Object Type	LDAP default types are iNetOrgPerson and groupOfUniqeNames for groups. This is a required field. The type of object with which the attributes are associated. For example, User

Fields	Descriptions
	and Group for Active Directory LDAP or DBA_USER and DBA_ROLES for Oracle.
Identity Attribute	This is a required field. Do not change the identity attribute on connectors with pre-defined schema. The attribute that is used by the IdentityIQ application to identify the object.
Include Permissions	Select this function to automatically add directPermissions to the schema. This option is available for any application that has `DIRECT_PERMISSIONS` in the `featureString`, for example, Oracle, DelimitedFile, and sybase HR. With this option activated, IdentityIQ correctly pulls in permission data for identities.
Display Attribute	This is a required field. The attribute that is used as the object name as it appears throughout the IdentityIQ application.
Instance Attribute	The attribute that uniquely identifies a specific instance of an application. Instance Attributes are not supported for Managed Attributes.
Remediation Modifiable	Accounts that are remediation modifiable can have their values and permissions modified from the Certification Report page for the identity being certified. Specify the method of modification for this attribute: • **Select** – display a select list of all possible values or permissions for this attribute. • **Free text** – display a text field in which a certifier can enter any value.
Additional Group Attributes:	
Description Attribute	Used during group aggregation to indicate which of the group attributes is used to populate the corresponding ManagedAttribute description. The value set here overwrites any set during the Account Group Aggregation task.
Group Membership Attribute	The attribute that is used by the IdentityIQ application to identify the group.
Attributes:	
Name	Attribute names cannot begin with IIQ_. Attributes with names that begin with IIQ_ are considered internal, reserved attributes and are not displayed in the

Fields	Descriptions
	product. The name of the attribute.
Description	A brief description of the attribute.
Type	The type of attribute being defined. For example, string or boolean. Select from the dropdown list.
Properties: Click Edit to open the Advanced Properties dialog to edit the attribute properties.	
Managed	Specify attributes to promoted to a first-class object in the IdentityIQ database so that they can be associated with other objects with that value, for example a description or an owner. Any attribute can become managed: department, location, title, but the most common attribute to be managed is the one holding group memberships. Managed attributes can be viewed and managed from the Entitlement Catalog page.
Entitlement	Specify attributes to be used as entitlements on this application. Attributes specified as entitlements are used by IdentityIQ as follows: • As additional entitlements during certification. • When creating profiles based on exiting users on this application. Profiles are created on the IdentityIQ Modeler and are used to create roles. • In account group certifications. • In Lifecycle Manager.
Multi-Valued	Specify attributes for which multiple values might be returned during aggregation. Attributes flagged as multi-valued are stored as a list. Even objects that have a single value for a multi-value attribute are stored as a single-item list. Multi-valued attributes are used for queries throughout the product. Before multi-valued attributes are available for use in searches, they must be mapped on the Edit Account Attribute page. Refer to the IdentityIQ system configuration documentation for information on how to add or edit account attributes.

Fields	Descriptions
Correlation Key	Specify attributes that IdentityIQ can use to correlate activity discovered in the activity logs for this application with information stored in identity cubes. For example, activity logs might contain the full name of users instead of unique account ids. Therefore, correlation between the activity discovered by an activity scan and the identity cube of the user that performed the action must key off of the user's full name. Correlation Key is only used during activity aggregation. If activity aggregation is not being used, Correlation Key should not be selected.
Minable	Specify attributes for use during role and profile creation. When creating roles and profiles it is possible to mine applications for attributes and permissions to use in those objects rather than manually entering the values. Only attributes designated as minable are returned by those searches.
Remediation Modifiable	Attributes that are remediation modifiable can have their values and permissions modified from the Certification Report page for the identity being certified. Specify the method of modification for this attribute: • **Select** – display a select list of all possible values or permissions for this attribute. • **Free text** – display a text field in which a certifier can enter any value.

Provisioning Policies Tab

Provisioning Policies are used to define application object attributes that must be managed due to a Lifecycle Manager request. With a provisioning policy in place, when a role or entitlement is requested the user must input specified criteria into a generated form before the request can be completed. A policy can be attached to an IdentityIQ application object or role and is used as part of the provisioning process.

For more details on Provisioning Policies, see Provisioning Policies.

For applications that support multiple application objects, each object is displayed in a separate table containing the provisioning policies those objects support. Not all application objects support all of the provisioning policies listed below.

In order to be able to provision to a DN with a backslash (\) to an Active Directory application through the Cloud Gateway you will need to set the following properties in `catalina.sh` or `catalina.bat` on the Cloud Gateway instance:

```
set CATALINA_OPTS=%CATALINA_OPTS%
-Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true
set CATALINA_OPTS=%CATALINA_OPTS%
-Dorg.apache.catalina.connector.CoyoteAdapter.ALLOW_BACKSLASH=true
```

Setting the dependencies between applications and accounts implies ordering in provisioning.

IdentityIQ includes the following types of provisioning policies:

- Create
- Update
- Delete
- Enable Account
- Disable Account
- Unlock Account
- Change Password
- CreateGroup
- UpdateGroup

Click an existing provisioning policy or click **Add Policy** to create a new one using the Provisioning Policy Editor or to reference an existing policy. Only one of each policy types is supported.

Use the **Application Dependencies** dropdown list to create the list of applications where this application is dependent for provisioning. If no account is detected on an application where this application is dependent, an account request is added to the provisioning plan and the provision policy for this application is processed as expected.

The Provisioning Policy Editor panel contains the following information:

Field Name	Description
Name	The name of your provisioning policy.
Description	A brief description of the provisioning policy.
Owner	The owner of the provisioning policy. This is determined by selecting from the following: • **None** – no owner is assigned to this provisioning policy.

Field Name	Description
	- **Application Owner** – identity assigned as owner of the application in which the provisioning policy resides. - **Role Owner** – identity assigned as owner of the role in which the provisioning policy resides. - **Rule** – use a rule to determine the owner of this provisioning policy. - **Script** – use a script to determine the owner of this provisioning policy
Edit Provisioning Policy Fields Panel Use the Edit Provisioning Policy Fields panel to customize the look and function of the form fields generated from the provisioning policy.	
Name	The name of the field.
Display Name	The name displayed for the field in the form generated by the provisioning policy.
Help Text	The text you wish to appear when hovering the mouse over the help icon.
Type	Select the type of field from the dropdown list. Choose from the following: - **Boolean** – true or false values field - **Date** – calendar date field - **Integer** – only numerical values field - **Long** – similar to integer but is used for large numerical values - **Identity** – specific identity in IdentityIQ field - **Secret** – hidden text field - **String** – text field
Multi Valued	Choose this to have more than one selectable value in this field of the generated form. Click the plus sign to add another value.

Field Name	Description
Read Only	Determine how the read only value is derived: • **Value** – value based on the selection from the drop-down list • **Rule** – value is based on a specified rule • **Script** – value is determined by the execution of a script
Hidden	Determine how the hidden value is derived: • **Value** – value based on the selection from the drop-down list • **Rule** – value is based on a specified rule • **Script** – value is determined by the execution of a script
Owner	The owner of this provisioning policy field. This is determined by selecting from the following: • **None** – no owner is assigned to this provisioning policy. • **Application Owner** – identity assigned as owner of the application in which the provisioning policy resides. • **Role Owner** – identity assigned as owner of the role in which the provisioning policy resides. • **Rule** – use a rule to determine the owner of this provisioning policy. • **Script** – use a script to determine the owner of this provisioning policy
Required	Choose whether or not to have the completion of this field a requirement for submitting the form.
Review Required	Choose whether or not to require the person who is approving the workflow item to approve this field.
Refresh Form on Change	Select this option to have the form associated with this policy refresh to reflex changes to this policy.
Display Only	Set this field as display only.
Authoritative	Boolean that specifies whether the field value should completely replace the current value rather than be merged with it; applicable

Field Name	Description
	only for multi-valued attributes
Value	Determine how the value is derived. Select from the following: • **Literal** – value is based on the information you provide • **Rule** – value is based on a specified rule • **Script** – value is determined by the execution of a script
Validation	Gives the ability to specify a script or rule for validating the user's value. For example, a script that validates that a password is 8 characters or longer.

Correlation Tab

Use the correlation tab to configure how application accounts are assigned to identities within IdentityIQ using account and identity information.

To configure Account Correlation you can select an existing correlation configuration from the list or create a new configuration using the correlation wizard. The correlation wizard walks you through both attribute and condition based correlation.

In the manager correlation section, configure how assigned managers should be resolved to identities using existing information.

Attribute Based Correlation

Use attributes of the application's account to find identities based on attribute values stored on Identity objects. This is how accounts are typically correlated to Identities. For example, you can correlate the application's account attribute "mail" with an identity's attribute "email."

Condition Based Correlation

Assigns application accounts to existing identities by defining attribute conditions. Service and Administrator accounts might be handled using condition based correlation. For example, the root account on Unix typically does not have any identifying attributes that can help when trying correlate it to an existing identity. In cases where the account owner is known because they are the application owner, a direct mapping can be used.

To configure Manager Correlation you must select two attributes, the Application Attribute and the Identity Attribute.

Application Attribute

The name of the application's account attribute that holds the reference to the manager.

Identity Attribute

The name of the identity attribute to use when searching for managers.

For example, if the application has an attribute `managerEmail` with the value set as the email address of the manager of every user with an account on the application, and you have an identity attribute `email` configured within IdentityIQ with the value set as the email address for every identity cube, you would correlate the application attribute `managerEmail` with the identity attribute `email` to perform manager correlation.

Accounts Tab

> Note: This tab is only visible after an application is defined and configured.

The Accounts Tab lists the following information for the selected account:

- Account ID
- Account Name
- Status
- Last Refresh
- Identity Name

Click the down arrow next to an account name to view details about the account, such as the last login time and date.

Risk Tab

The application Risk tab provides a current application risk score and a detailed view of the raw and compensated risk score for each category used to derive that score. This page also provides a list of the top composite score contributors providing further information on how the score was derived and providing clues on the areas of highest risk. These scores are based on the latest information discovered by IdentityIQ.

IdentityIQ uses a combination of base access risk and compensated scoring to determine the overall application risk scores, or composite risk score, used throughout the application.

All scores are calculated by first determining the percentage of accounts that have the qualities tested by the component score. For example, if 10 out of 100 accounts are flagged as service accounts, then the raw percentage is ten percent (.10). This number is then multiplied by a sensitivity value which can be used to increase or decrease the impact of the original percentage. The default sensitivity value is 5 making the adjusted percentage fifty percent (.50). This final percentage is then applied to the score range of 1000 resulting in a component score of 500.

After the component score is calculated a weight, or compensating factor, is applied to each component score to determine the amount each contributes to the overall risk score for the application. For example, a few violator accounts might increase risk more than many inactive accounts.

Service, Inactive, and Privileged component scores look for links that have a configured attribute. For example, the component `service` with a configured value `true`.

The Dormant Account score looks for a configured attribute that is expected to have a date value, for example `lastLogin`. This algorithm has an argument, `daysTillDormant`, that defaults to thirty (30). If the last login date is more than thirty (30) days prior to the current date, the account is considered dormant and is factored into the risk score.

The Risky Account score looks for links whose owning identity has a composite risk score greater than a configured threshold. The default threshold is five hundred (500).

The Violator Account score looks for links whose owning identity has a number of policy violations greater than a configured threshold. The default threshold is ten (10).

Activity Data Sources Tab

The Activity Data Source tab is used to configure the data sources from which activity information is collected. The information collected from these sources is normalized and then stored by IdentityIQ and used to monitor activity information for users and applications. Activity information is collected and correlated using the Activity Aggregation task. Activity information displayed on the or returned by activity searches is based on the information stored by IdentityIQ since the last aggregation and correlation tasks were run.

The Activity Data Sources table contains the following information:

Name

A descriptive name for the activity data source from which the activity data is collected.

Type

The activity data source type, for example JDBC Collector, Log File, CEF Log File or Windows Event Log Collector.

Modified

The date when the activity data source was last modified.

Right-click a data source and select **Edit** or click **New Activity Data Source** to access the Activity Data Source Configuration page.

Right-click a data source and select **Delete** to remove an activity data source.

See Activity Data Source Configuration.

Unstructured Targets Tab

Unstructured target information is used to define unstructured data sources from which the connector is to extract data. Unstructured data is any data that is stored in a format that is not easily readable by a machine. For example, information contained in an Excel spread sheet, the body of an email, a Microsoft Word document, or an HTML file is considered unstructured data. Unstructured targets pose a number of challenges for IdentityIQ connectors, because not only is the data stored in a format that is hard to extract from, the systems and directory structures in which the files reside are often difficult to access.

The most common unstructured data type supported by IdentityIQ is an operating system's file system permissions.

This target collector requires a the IdentityIQ Service to be installed on a machine that has visibility to the directory or share to include in the target scan. See the **Installation** documentation for information on installing and registering the IQService.

The unstructured targets defined on this tab are used by the Target Aggregation task to correlate targets with permissions assigned to identities and account groups for use in certifications.

Each of the Target Source Types require unique details or attributes for their configuration, but share some basic information.

The Unstructured Targets tab contains the following basic information for each of the Target Source Types:

Name

The name of this unstructured target configuration.

Description

A brief description of this configuration.

Rules

Specify the rules used to transform and correlate the targets.

> Note: Click the [...] icon to launch the Rule Editor to make changes to your rules if needed.

Creation Rule

The rule used to determine how the unstructured data extracted from data source is transformed into data that can be read by IdentityIQ.

Correlation Rule

The rule used to determine how to correlate account information from the application with identity cubes in IdentityIQ.

Provisioning Action

The provisioning action used to pass information to the affected applications.

Rules Tab

These are the rules that can be customized to handle data the manipulation requirements of the aggregation or provisioning processes for each application. Rules are specific to connectors and are used throughout the product. You can write more than one of each type and select the rule to use from dropdown lists.

A file containing an example of each rule type is included in the IdentityIQ installation package. The `examplerules.xml` file is located in the *`IdentityIQ_HOME`*`/WEB-INF/config directory`.

Many rule types apply to all applications and are called by the aggregation process. Other connectors may include additional rule options that are specific to the connector type.

For an overview of developing and using rules in IdentityIQ, see the *IdentityIQ System Administration* guide.

Aggregation Rules

These rules define behavior when aggregating data from the application. Aggregation Rules are used during part of the aggregation process that occurs after the connector has created valid ResourceObjects for the accounts or groups being aggregated, which occurs after the defined connector rules have all been run.

Correlation Rule

This rule type is used to define identity correlation for applications whose accounts cannot be correlated to Identities through a simple attribute match in a correlation configuration. A correlation rule is required if the logic to correlate accounts is more complex than a correlation configuration will support (for example, correlating based on multiple attributes taken together).

Creation Rule

This rule is only run when a new identity is being created during the aggregation. Some examples of operations which might be performed in a creation rule include setting an identity's password, changing the name of aggregated identities, or setting an attribute to all caps. Typically, this rule is only used for authoritative applications, though one could be specified for a non-authoritative application if needed.

Manager Correlation Rule

A manager correlation specifies how one identity should be linked to another in a Manager – Direct-report relationship. This can often be accomplished with a simple attribute mapping correlation on the Correlation tab, but when a more complex matching algorithm is required, this rule type can be used.

Customization Rule

This is an open-ended rule which is available to all application types in IdentityIQ for manipulating or customizing the aggregated data before the account (or group) is saved to the database. This rule type is most often used for manipulating data on application types which do not offer Connector Rules, since for those applications, this rule is the only account-manipulation rule offered.

Managed Entitlement Customization Rule

The Managed Entitlement Customization rule is called during creation of Managed Entitlements in an aggregation or refresh task, or in the Missing Managed Entitlements Scan task. This rule can modify the Managed Entitlement as it is being created, to set fields such as owner, requestable, or descriptions before they are saved. "Managed Entitlements" are "ManagedAttribute" objects, which are Entitlement Catalog entries.

Provisioning Rules

These rules run during the processing of provisioning requests. Some are connector specific and some apply for all connectors, as indicated in their descriptions. Provisioning-related rules which apply to all application types are:

Before Provisioning

The BeforeProvisioning rule is executed immediately before the connector's provisioning method is called. This gives customer the ability to customize or react to anything in the ProvisioningPlan before the requests are sent to the underlying connectors used in provisioning. This rule is not connector-specific; it runs for all applications regardless of connector type.

After Provisioning

An application's AfterProvisioning rule is executed immediately after the connector's provisioning method is called, but only if the provisioning result is in a committed or queued state. This gives customers the ability to customize or react to anything in the ProvisioningPlan that has been sent out to specific applications after the provisioning request has been processed. This rule is not connector-specific; it runs for all applications regardless of connector type.

Schema Rules

Schema rules vary by connector and are used for customization. Schema rules allow you to segregate business logic account and group objects respectively, avoiding the need to check whether the resourceobject represents an account or non-account object.

Connector Rules

Connector rules are used during aggregation from specific connectors, such as Delimited File, JDBC, and SAP. Connector rules generally run before Aggregation rules in the aggregation process. These rules are used to implement pre-processing of data, implement post-processing of data, and manipulate, merge, or otherwise transform the incoming data as it is being read. Connector rules vary by connector type.

Password Policy Tab

Use the password policy tab to select and create password policies which apply to specified applications.

The password policy panel contains the following:

Name

The name of your password policy.

Description

A brief description of the password policy.

Click an existing password policy to edit it or click **Create New Policy** to configure one from scratch.

Password Policy Options:

Password Policy Name

The name of your password policy.

Password Policy Description

A brief description of the password policy.

Minimum number of characters

The minimum number of characters required for a valid password.

Maximum number of characters

The maximum number of characters allowed for a valid password.

Minimum number of letters

The minimum number of letters required for a valid password.

Minimum number of character type constraints to meet

The minimum number of character types (digits, upper case, lower case, special) allowed for a valid password.

Minimum number of digits

The maximum number of numerical digits allowed for a valid password.

Minimum number of uppercase letters

The minimum number of uppercase letters required for a valid password.

Minimum number of lowercase letters

The minimum number of lowercase letters required for a valid password.

Minimum number of special characters

The minimum number of special characters required for a valid password.

Number of repeated characters allowed

The maximum number of consecutive repeated characters allowed in a valid password. For example, if this option is set to 2, "cloudd" and "cclooud" are valid, but "clouddd," "clooud" and "cccloud" are invalid. This value also sets the maximum number of occurrences of repeated characters allowed in a valid password. For example, if this option is set to 2, "happy123" is valid, however, "happy22" and "happpy123" are not.

In this example, when "cclooudd" is an invalid password, the following error message is displayed: Password should not contain more than 2 occurrence(s) of the repeated characters. When "clouddd" is an invalid password, the following error message is displayed: Password should not contain more than 2 consecutive repeated characters.

Setting this value to zero has the same effect as leaving this field blank, allowing any number of repeated and consecutive characters.

To prevent the use of any **consecutive** repeated characters, set this value to 1. Setting the value to 1 does not prevent using a character more than once, as long as the characters are not consecutive. For example, with a value of 1, "kitkat" is valid but "kitten" is not.

Password history length

The number of past passwords that cannot be used again.

Triviality check against old password

Ensure that the shorter of the old and new password is not a substring of the other.

Both passwords are changed to upper case prior to the check.

Minimum number of characters by position

The minimum number of unique characters by position the new password. Can be used to ensure that not just the first or last character is being changed.
Select **Case sensitive check** to ensure that more than just the case is changing in the new password.

Validate passwords against the password dictionary

Select this option to disallow the use of any password defined in the password dictionary. The password dictionary is a configurable list of terms unavailable for use as passwords. The `passwordDictionary.xml` file located in `IdentityIQ/WEB-INF/config/`.

Validate passwords against the identity's list of attributes

Select to disallow the use of Identity attribute values as passwords.

Validate password against the account's display name

Select to disallow the use of the account's display name as the password (exact match by default).

Enter a **Minimum word length** to define the minimum length of a substring of the account's display name allowed in the password.

Validate password against account ID

Select to disallow the use of the account's ID as the password (exact match by default).

Enter a **Minimum word length** to define the minimum length of a substring of the display name of the account allowed in the password.

Validate passwords against the identity's account attributes

Select to disallow the use of Identity link attribute values as passwords.

Enter a **Minimum word length** to define the minimum length of a substring of the account's ID allowed in the password.

Configure Password Filter

Select a filter that selects the identities to which this password policy applies. Select from the following filters:

- **All** – all identities have this password policy applied
  Match List – only identities whose criteria match that specified in the list. The criteria is configured using the tools provided. Add identity attributes, application attributes and application permissions. Customize further

by creating attribute groups to which this password policy applies.

> Note: If Is Null is selected, the associated value text box is disabled. When the is null match is processed, the term matches users on the chosen application who have a null value for that attribute/permission.

- **Filter** – use an XML filter or compound filter to determine the identities to which this password policy applies.
- **Script** – use a BeanShell script to determine the identities to which this password policy applies.
- **Rule** – use a rule to determine the identities to which this password policy applies.
- **Population** – select a population to which this password policy applies.

Application Connection Parameters

Connection parameters define how the application will connect to and communicate with the target system. For example, for a delimited file application, you will need to specify the name and location of the delimited file, as well as what the delimiter is, and any file encoding. For a JDBC application, your connection parameters will include the database URL, the JDBC driver, and a username and password for the connection.

The required configuration parameters vary by connector, based on the specific requirements of the target system.

For more information about specific connectors, refer to the IdentityIQ Connectors documentation on SailPoint's documentation portal.

Application Schemas

For every application, you must define the specific data you want to aggregate from the target system. The target system may have dozens or hundreds of attributes for each user, not all of which may be relevant for your identity governance program, and which therefore do not need to be brought into IdentityIQ. The schemas you define for the application specify which attributes to include when aggregating data from the target system.

Most connectors include **predefined** default schemas, which you can use as a starting point for defining the schemas you need. When a connector supports a predefined schema, you will see the attributes listed and defined for you on the **Configuration > Schema** tab. You can add or remove attributes from a predefined schema as needed.

Account Schemas

Every configured application must include an account schema, which defines which data about accounts to read from the target application and identifies the accounts you want to manage.

The account schema must designate one of the attributes as the **Identity Attribute**, which is the unique identifier for the account on the source.

Group Schemas

For many applications, account entitlements are memberships in groups. Many connectors also support the use of group schemas, allowing the application to aggregate additional details about the group structures from the target system.

Once a group schema is defined, you will need to connect the account schema to the group schema so IdentityIQ can recognize that the account entitlements identify group memberships. This is done by setting the **Type** for the entitlement attribute in the account schema to a value that matches the Native Object Type of the group schema. This value is often "group"; however, some connectors support multiple group schemas or offer more flexible options for schema definitions, so in some cases this may be a value other than "group."

Values for the entitlement attribute on accounts will be mapped to the Identity Attribute selected for the group schema to associate the groups with those users.

Some connectors support only a single group schema, and others support multiple group schemas.For more information about specific connectors, refer to the IdentityIQ Connectors documentation on SailPoint's documentation portal.

Properties of Attributes in Account and Group Schemas

Attribute properties determine how attributes are used and managed in IdentityIQ. For example, you may want some attributes to support multiple values, or to be included in the Entitlement Catalog. Select **Edit** to open the Advanced Properties dialog to edit the attribute properties for accounts and groups.

Entitlement

Marking an attribute as an entitlement indicates that this is an access right you want to track for your identities (for example, to use in certifications). If you want this attribute to be able to be requested, to have an owner, and to have a description and display name, you must mark also mark it as Managed.

Managed

Attributes designated as Managed can be viewed and managed from the Entitlement Catalog page. Managed attributes can be made requestable, can be assigned an owner (for approvals or entitlement certifications), and can have display names and descriptions that will help users identify and understand them. They can also be used in policies and risk calculations.

When you do a group aggregation, all groups read from the aggregation are automatically included in the entitlement catalog as managed attributes.

Multi-valued

For some attributes, multiple values might be returned during aggregation (for example, an attribute indicating group membership). These should be marked as Multi-valued. Values for attributes flagged as multi-valued are stored as a list. Even objects that have a single value for a multi-value attribute are stored as a single-item list.

Correlation Key

The Correlation Key flag is only used for activity and unstructured data aggregation. If activity aggregation is not being used, Correlation Key should not be selected. This flag specifies attributes that IdentityIQ can use to correlate activity discovered in the activity logs for this application with information stored in Identity Cubes. For information about correlating aggregated accounts to existing identities, see **Correlation** in Application Concepts.

For example, activity logs might contain the full name of users instead of unique account ids. Therefore, correlation of the activity discovered by an activity scan and the Identity Cube of the user that performed the action must key off of the user's full name.

Minable

Attributes that you want to use for role and profile creation should be marked as minable. This allows the Role Mining feature to mine applications for attributes and permissions when creating roles and profiles, rather than requiring manual entry of the values. Only attributes designated as minable are returned by those searches.

Remediation Modifiable

Attributes that are remediation modifiable can have their values and permissions modified as part of a certification, for the identity being certified. Options are:

- Select – in the certification, display a select list of all possible values or permissions for this attribute.
- Free text – in the certification, display a text field in which a certifier can enter any value.
- Readonly – the value cannot be modified.

Discovering Schemas

Some connectors support the discovery of schemas. For those application types, you can select **Discover Schema** Attributes to automatically populate your schema information in the Schema page.

Previewing Data for a Schema

Each defined schema offers a **Preview** button; this is useful for checking the schema against a small sampling of the source system's data before aggregating.

Avoiding Conflicts Between Schema Attributes and Extended Attributes

If your IdentityIQ instance makes use of extended attributes, you should ensure that the names of your extended attributes names do not duplicate any attribute names in your application's schema(s). If an extended attribute name matches a schema attribute name, there is a risk that attribute values will not be updated correctly during an aggregation.

For more information on extended attributes, refer to the *IdentityIQ System Configuration Guide*.

Provisioning Policies

Provisioning policies in an application configuration define the set of attributes which are needed to complete a provisioning request, whether that request is to create an account, modify an account, add a role to an identity, etc. In some cases, values for the attributes can be auto-calculated based on the field's definition in the provisioning policy; in other cases, the fields are presented to a user who can provide their values. When auto-calculation is possible, it is encouraged as a best practice, because it is typically more efficient and less error-prone than user-provided values.

Provisioning policies are implemented as **forms** that specify which attributes to include in provisioning requests, and how the values of those attributes are provided – manually by an IdentityIQ user, or in an automated manner, either through a static value, or a value that IdentityIQ calculates using a script or rule.

For example, if you are provisioning a new account on an LDAP application, the provisioning policy should provide the minimum required attributes (such as user name and email) to successfully create a new account on the LDAP application.

In addition, optional attributes can be included in the provisioning policy to set additional values as needed.

Provisioning policies can be configured so that all values are provided automatically by IdentityIQ. Alternatively, attributes can be specified without a value but marked as required, so that IdentityIQ will prompt a user to provide the data manually.

In this example, the values for First, Last, and Login are auto-calculated based on identity attributes; values for Status and Locked are specified as static values.

Attribute	Value
First	return identity.firstname
Last	return identity.lastname
Login	return identity.name
Status	A
Locked	N

Based on this policy, IdentityIQ can automatically build a provisioning plan to pass the required values to the connector to create a new account on the target system.

For some connector types, the provisioning policy for creating accounts is pre-defined; however, predefined policies can be modified as needed, according to your business needs.

When a user submits a Lifecycle Manager request, such requesting a business role or unlocking an account on a target system, IdentityIQ pulls the list of required attributes from the provisioning policy, and includes those attributes and their values in the provisioning plan.

Using the Form Editor for Configuring Provisioning Policies

A provisioning policy is essentially a collection of fields. These fields are defined in a form editor UI. For each field, you can define a variety of settings that determine what type of information is required and how IdentityIQ provides it.

The provisioning policy's **Name** field should match the corresponding native attribute on the application.

If **Review Required** is selected for a field, the field is always presented on a form during provisioning-request processing, even if a default value is provided or calculated successfully.

For more information on working with form fields, see the *IdentityIQ Forms Guide*.

Application Dependencies

Your business may have dependency requirements, where a user's access to one application is dependent on access to another application. For direct-connection applications, you can define **Application Dependencies** on the **Provisioning** tab, to specify that a user must have an account on one system before they can have an account on another system. When an Application Dependency has been defined, IdentityIQ's provisioning process will check to see if there is an account on the application(s) listed. If not, when IdentityIQ processes a provisioning request for a new account, a request for the an account on specified system(s) is added.

When Application Dependencies have been defined, you can use values from these applications as part of your Provisioning plans. When you select Value Settings in the provisioning form, you will see dependent as an option. You can choose the application and the attribute to retrieve from it in your form.

Using Rules in Applications

Rules in applications allow the addition of custom business logic to extend the functionality of IdentityIQ in aggregating from or provisioning to the application.

Some rules are applicable to and can be used in all applications. Other connector types support rules that are specific to the connector type. For more information on connector-specific rules, see Rules Tab.

When implementing rules for applications, be aware of potential performance impacts. Complex rules that act on every record being processed by IdentityIQ can have a significant performance impact.

For general information about implementing rules, see the *IdentityIQ System Administration Guide*.

Native Change Detection Configuration

Native change detection is a mechanism in IdentityIQ's application connectors that can detect changes made in account information outside of IdentityIQ's control. The native change detection can detect newly created accounts, deleted accounts, or modified accounts. The changes are detected on aggregation by comparing stored information with newly read information. If any changes are detected, actions can be taken to respond to these changes. These actions include automatic recertification, approval, notifications or even automatically reverting the changes.

Native change detection is enabled **on each application separately**.

Once enabled, aggregation will start detecting changes (while filtering SailPoint requested items) and storing them with other Lifecycle Events on the identity. If you make native changes you will see them being stored on the Identity object.

> Note: For Native Change Detection to operate you must have both a lifecycle event defined and the application enabled.

To configure IdentityIQ to detect native changes during aggregation, do the following:

1. Run an aggregation to obtain the baseline information for the application.
2. Configure a Native Change life cycle event on the Lifecycle Events page.

There are two lifecycle change events included with IdentityIQ and you can configure your own as needed:

Lifecycle Event – Email manager for all native changes

Sends a formatted email to the manager describing the changes detected.

Lifecycle Event – Manager Approval for all native changes

Generates an approval work item for each change detected. Any items rejected are undone / reversed and provisioned. This business process also creates an access request within IdentityIQ so that once the changes are made they will be visible from the Access Request page.

1. Go to **Applications > Application Definition** and select an application.
2. Select **Native Change Detection** on the Details panel of the application configuration UI.
3. Define the operations to include when detecting native changes – **Create**, **Modify**, **Delete**.
4. Define the attributes to compare when detecting native changes:
   Entitlements: All entitlement attributes
   User Defined: Manually enter the names of the attributes to compare, one per line.

5. Run or schedule aggregations to detect and store any changes.
6. Run an Identity Refresh task with the Process Events option enabled to trigger the lifecycle events for any changes detected since the last time the events were processed.

Editing an Application's Configuration in the Application XML

The majority of the options you will use to create and configure an application are available in IdentityIQ's UI. Each configured application is stored in IdentityIQ as an application object. These objects can be viewed in the Debug pages, and they can be exported for modification and testing, then imported to implement these changes in your production environment. Only users with system administrator or similar elevated rights in IdentityIQ have access to the Debug pages.

In some limited cases, specific behaviors for certain connectors can only be configured by editing the application's XML. Where these occur, they are described in the specific connector guides. For more information about specific connectors, refer to the IdentityIQ Connectors documentation on SailPoint's documentation portal.

> **Important**:
> Use caution when editing application objects in the Debug pages, as **there is no mechanism in the Debug pages for rolling back changes once they have been saved**. It is a best practice to export the application object, edit the XML outside of IdentityIQ, and then import the updated object to a test environment for validation before importing into your production instance. This also allows you to manage your application changes in a source control system, to facilitate rollback.
>
> For modifying a single application, the iiq console **checkout** command is preferable to the **export** command, because **checkout** lets you write a copy of a single object, whereas **export** will write all objects of the class you specify. For more information, see the *IdentityIQ Console Guide*.

Only users with **system administrator** or similar elevated rights in IdentityIQ have access to the Debug pages.

To access an application object in the Debug pages:

1. In the IdentityIQ menu, select the **wrench icon > Object**.
2. In the Object Browser, select the **Application** object.
3. The configured applications are listed alphabetically. Scroll or use the search box to locate the application object.
4. Double-click the object to open the application XML.

Application Maintenance Windows

Applications may need to be taken offline for maintenance at various points and for various reasons. You can set a maintenance window in IdentityIQ for any application; maintenance windows are timeframes you specify when an application offline. During a maintenance window, aggregation tasks and provisioning operations are suspended for the application until the maintenance window ends.

Defining a Maintenance Window

To set a maintenance window for an application:

1. Navigate to **Applications > Application Definition** and open the application you want to edit.
2. On the **Details** tab, check **Maintenance Enabled**. To set the timeframe for the maintenance window, you can either:
   - Set a specific date and time when the maintenance window should end and the application should return to online status.
   - Leave the expiration field blank to keep the application in maintenance mode until you are ready to bring it back online. In this case, you can return to the application definition to manually clear the **Maintenance Enabled** checkbox when the maintenance window is over.
3. **Save** your changes.

Maintenance Windows and Aggregation

During a maintenance window, IdentityIQ automatically skips the application when running aggregations. A warning message is recorded in the TaskResult for skipped applications.

Maintenance Windows and Provisioning

When an application is in a maintenance window, any provisioning operations (which might be launched by Identity Refresh tasks, certifications, LCM business processes, or custom business processes) are prevented from running for the application. A status of "retry" is automatically returned for any provisioning requests to the applications.

In processes that are launched by a **business process**, the provisioning will use the standard "retry" behavior configured for the workflow, to attempt to provision again later.

In **task-based** provisioning processes (such as an Identity Refresh task), retries are handled through Request objects that are created each time the process runs. These are automatically re-processed through the built-in Request Processor retry logic. For example, if you have an Identity Refresh task scheduled to run twice per day, and an application is in a maintenance window for 5 days, IdentityIQ will create 10 Request objects for that action.

Certification remediations will also generate retry Request objects when the target application is in a maintenance window.

If you want to change the retry behavior, you can do so in these ways:

- It is possible, but not recommended, to modify business process retry behavior through the Provision with retries subprocess. Refer to the LCM Subprocess Workflows white paper on Compass for more information.
- In the Identity Refresh task, you can select the *"Do not schedule retry requests during application maintenance windows"* option to prevent retries.
- For certifications, there is not an option to turn retries off; certification remediations need to be set for retry, because certification cannot be simply "rerun" in the way that tasks and business processes can be.

Reconfiguring an Application

The application reconfiguration option enables you to change the application type without losing history associated with the application or having to create a new application. For example, if you first deployed your instance of IdentityIQ using a flat file connection, but now want to use some of the more advance features, such as provisioning. The type defines the way in which IdentityIQ connects to the application.

Application types that have the same value format for identity and group attributes in the original and reconfigure target are best suited for reconfiguration.

The following application types can be reconfigured:

> Note: Even in the following scenarios, there might be some connectors that do not re-configure correctly. See Application Reconfiguration Considerations.

- Delimited file to the corresponding direct connector (Delimited File to Active Directory – Direct)
- JDBC connector to corresponding direct connector (JDBC to Oracle Applications – Direct)
- Agent based connector to direct connector (Active Directory – Full to Active Directory – Direct)
- To a rewritten connector for better performance or more functional support (Google Apps – Direct)

Application Reconfiguration Considerations

Take the following points into consideration before deciding to reconfigure an application.

Do the identity attribute of account and group in the original application match the identity attribute of account and group in the reconfigured application?

For example:

- Two application types (Delimited File and Active Directory – Direct) use distinguishedName as the identity attribute of account, and use the same identity attribute for group. Since both of these applications refer to the same identity attribute of both account and group, they would be good candidates for reconfiguration.
- Two application types (Oracle Application – FULL and Oracle Application – Direct) use different identity attributes for account, USER_ID in one and USER_NAME in the other and, USER_ID and USER_NAME differ in format. These are not good candidates for reconfiguration.

> Note: If there are special attributes (native identity, managed attribute, entitlement) that split into multiple attributes in the new application type, reconfiguration is not recommended.

- Profiles in SAP–Full refer to both profiles and groups in the managed system, whereas in SAP–Direct, profile refers to profiles and group refers to the groups in the managed system. These are not good candidates for reconfiguration.

Before Application Reconfiguration

Perform the following actions before you begin the reconfiguration process:

- Backup the application xml and application type specific customizations such as rules and business processes.
- Plan the attribute mapping of the original and new applications for accounts and group schema. If there are attributes in the original application type that are not in the reconfigured application type, you might lose some configuration and historical data.
- Check the provisioning policies of the target application and decide which policy to use, the policy from the original application type or the policy from the reconfigured application type.

How to Reconfigure an Application

> Note: While reconfiguring an application the target application must have a static schema and not a dynamic schema like JDBC or DelimitedFile connectors. There is the button named Discover Schemas to generate the schema.

1. Go to **Applications > Application Definition** and select an application.
2. On the Application Configuration page, click the **Reconfigure** button to display the Select New Application Type dialog.
3. Select an application type from the **New Application Type** dropdown list and click **Save**.
4. Confirm your selection to go to the Application Configuration page in edit mode.
5. The tabs that contain information requiring attention are marked with an red asterisk.
6. Go to the Attributes tab and enter the valid configuration attribute settings and test the connection.
7. Go to the Schema Mapping tab and map the Previous Schema Attributes to the New Application Type Schema Attributes for the Account and Group.

   Use the **Add Missing Attributes** and **Keep Extra Attributes** options to select to add missing attributes from the old (original) application type to the new application type, and to keep attributes that are on the new application type but were not on the original application type.

8. Go to the Provisioning Policy tab and select the provisioning policy to use for the reconfigured application.

   It is recommended that you use the policy that corresponds to the application type of the newly reconfigured application. You can use a different policy, but you must manually edit that policy to match the changes made during the reconfiguration process.

9. Save the reconfigured application.

After Application Reconfiguration

Check the reconfigured application for the following when the process is complete:

- Attributes that were not mapped might not work, and the values might not get populated.
  - Unmapped attributes affect configurations (for example, policy or business roles), and context based historical data (for example, viewing certification history), that is based on a population that relies on the attribute.
  - Related populations might not be populated with identities.
  - Pending provisioning operations that contain that attribute might fail.
  - Verify other places that use the attribute, such as identity mapping, account mapping, roles, policies, and policy violations.
- Verify the application definition for unwanted entries like build map rules or provisioning rules
- Run the account and account group aggregation task
- Run the refresh Identity Cube task
- Run a task to prune the Identity Cubes

File Access Manager Applications

File Access Manager enables enterprises to discover and govern access to sensitive data and better address the security threat to unstructured data. As a key component of SailPoint's Identity Governance strategy organizations can take a comprehensive approach to govern access across all users, applications and data with enhanced visibility while reducing risk.

Use File Access Manager to:

- Identify and govern access to exposed sensitive data found within cloud and on-premises file stores.
- Enable business users to manage access to data they know best, alleviating IT burden and over-permissioned access.
- Leverage a comprehensive identity governance solution that extends to unstructured data throughout the enterprise.

When you create a new application of type File Access Manager, this application by default will start with an alert schema, as well as unstructured and associations schemas. The unstructured and associations schemas are used to define the makeup the Target and TargetAssociation respectively.

For more information, you can also refer to the File Access Manager documentation.

File Access Manager Attributes/Configuration

Schemas are defined on the Configuration tab. If an alert schema is defined, this will include the configuration needed to set up the Alerts. If the Unstructured schema is defined, this will include the configuration needed to set up the Target / Target Permissions.

Connection Settings

- **Database URL** – the jdbc connection URL for the File Access Manager database.
- **Driver Class** – the SQLServer JDBC Driver class to use for the connection. This should be the appropriate jar for the version of SQLServer you are using with File Access Manager. It must be contained in the corresponding jdbc SQLServer jar file, and located in your IdentityIQ installation's `\lib` directory.
- **UserName** – the database user name
- **Password** – password for the configured user name
- **Schema** – schema used for the File Access Manager database

General Settings

Referenced Applications

This is a list of applications to which the given permission are correlated. The target permissions are correlated to either a Link or ManagedAttribute belonging to one of the applications in the list.

Aggregate Inherited

True to aggregate inherited permissions. If set to true, the dataset will be much larger. If false, only the top level permissions are aggregated, and inheritance is assumed as defined on the native source.

Target Hosts

The File Access Manager Business Application Monitors (BAMs) from which to aggregate permissions.

Target Host Paths

This is a CSV of paths from which to aggregate. This aggregation starts at the given root paths, and discovers all permissions under these paths. If not specified, all target / target permissions for the specified BAM are aggregated.

Rules

The rules tab within the Application Definition user interface enables the defining of rules for given object types. The Application level rules and schema level rules, for schemas that allow them, are shown with the ability to select / edit (based off of correct capabilities) the given rules. The unstructured schema support Correlation/Creation/Customization/Refresh rules on the schema level.

Creation rules for unstructured schema will be of **Rule Type** TargetCreation

Refresh rules for unstructured schema will be of **Rule Type** TargetRefresh

Correlation rules for unstructured schema will be of **Rule Type** TargetCorrelation

> Note: Customization rules might run multiple times, updating the same ResourceObject. For example, once for the provisioning result, once for the result in the provisioning plan, and once for the result in the account request.

Customization rules for unstructured schema will be of **Rule Type** ResourceObjectCustomization

The unstructured / associations schema AttributeDefinitions are used to define the columns to include in the query.

Activity Data Source Configuration

Use the Activity Data Source Configuration page to add or edit activity data sources. Activity collectors access activity data sources such as event or audit logs, collect the activity information that is to be monitored, and transform that data into a format that can be read by IdentityIQ. These Activity Data Sources are use for all activity aggregation and reporting.

Changes made on this page are not committed until a save is performed on the application with which they are associated. For example, if you add or delete a data source on this page and click **Save**, you do not see that change reflected on the application until you click **Save** on the application page and commit the change.

For each activity data source enter or edit the following:

- The general data source information in Activity Data Source Configuration.
- Activity target information found on the Activity Target tab for each source type, see Activity Targets.
- The unique connection and query setting for each activity data source type.
  - JDBC Collector Settings
  - Windows Event Log Collector Settings
  - Log File Collector Settings
  - RACF Audit Log Collector
  - CEF Log File

Activity Data Source Fields

Name

A short, descriptive name for the activity data source.

Description

A brief description of the activity data source.

Transformation Rule

The transformation rule required to convert the data collected from the data source into a format that can be used by IdentityIQ.

> Note: Click the [...] icon to launch the Rule Editor to make changes to your rules if needed.

Correlation Rule

The correlation rule that should be used to correlate the activity data collected with identities.

> Note: Click the [...] icon to launch the Rule Editor to make changes to your rules if needed.

Activity Data Source Type

The type of data source from which the activity is being collected.
The **Activity Data Source Type** dropdown list contains the types of data source from which activity information can be collected. This list will grow and change to meet the needs of IdentityIQ users.

> Note: When **CEF Log File** is selected from the dropdown list, the Transformation Rule and Correlation Rule fields are displayed with the following respective values:
> - Transformation Rule: CEFTransformRule
> - Correlation Rule: CEFActivityCorrelation

Activity Targets

The Activity Targets tab is used to specify targets within this data source for use in activity searches. A target is a specific object within a data source that is acted upon. For example, a target might be a machine name for a login action, or a file name for a create action.

The targets specified here are used to populate lists on the Activity Search page. These targets can be grouped with targets specified on other applications to create categories of targets. For example, if you have inventory applications at three different locations and a procurement database on each, you can set each procurement database as a target, create a Procurement category, and then collect activity for all three procurement databases using a single activity search.

On the Activity Targets tab you can add activity targets for the data source with which you are working. Type the name of the activity target in the field at the bottom of the list and click **Add Activity Target**. To remove activity targets, use the selection boxes on the left of the table and click **Delete**.

JDBC Collector Settings

JDBC Connection Settings

IdentityIQ uses the connection settings to access the activity data source.

Connection User

A valid JDBC user with access to the data source being accessed by this collector.

Connection Password

The password associated with the Connection User if a password is required.

The password is encrypted and is not displayed with the activity data source information.

Database URL

The full URL to the activity data source. For example, `jdbc:mysql://localhost/db`

JDBC Driver

The driver class of the activity data source. For example, `com.mysql.jdbc.Driver`

Query Settings

The query settings are used to control the activity information that is collected when an Activity Aggregation task is run.

SQL Statement

The SQL statement used to query activity from the database.

Condition Builder

Transforms the data mapped in the rule selected as the Position Builder into a SQL statement used by subsequent queries to determine start position.

Position Builder

Rule that converts the last row in the result set returned by the query into a configuration map that is persisted into the IdentityIQ database.

The data that is mapped in this rule is used by the condition builder to create a SQL statement used in future queries to determine the start location. This enables IdentityIQ to perform scheduled activity aggregations without having to scan entire data sets with each subsequent aggregation.

Windows Event Log Collector Settings

> Note: Before you can use the Windows Event Log Collector, the IQService must be installed and registered. See the **Installation** documentation for information on installing and registering the IQService.

Event Log Settings

SailPoint IdentityIQ uses the connection settings to access the activity data source and the query settings to control the activity information that is collected when an Activity Aggregation task is run.

User

Valid Windows user name with access to the event log containing the activity data.

Password

The password associated with the user specified.

IQ Service Host

The host name where the IQ service is running.

IQ Service Port

The listening port of the IQ service.

Event Log Server

The server where the activity data source resides.

Query String

The MQL query use to specify the activity data to collect during the activity aggregation.

Block Size

The number of events to retrieve with each activity aggregation performed on this activity data source.

Log File Collector Settings

Transport Settings

The transportation settings are used to access the server where the log file containing the activity data resides.

Transport Type – depending on the transport type selected you will see the following:

local

If the log file containing the activity data is on the same server as IdentityIQ, no further connection-type information is required.

ftp

FTP User – a valid user name with authentication access to the FTP host.

FTP Password – the password associated with the FTP user.

FTP Host – the host where the log file resides.

scp

SCP User – a valid user name with authentication access to the SCP host.

SCP Password – the password associated with the SCP user.

SCP Host – the host where the log file resides.

SCP Private Key – the private key that is used to encrypt the collected data.

Log File Settings

The log file settings are used to define the query used to collect the activity data on.

File Name

The name of the log file containing the activity data.

Lines to Skip

The number of lines to skip before starting the scan for activity information.

Filter Nulls

Skip lines that don't conform to the defined format.

Multi-lined Data

A single record in this file spans multiple rows.

Regular Expression

A regular expression groups that can be used to tokenize each record in the file.

Log Fields

The log field settings are used to create the log fields based on the column headings in the log file.

Name

The name of the log field to create based on a column name from the log file.

Trim Value

Remove white space around the column name before creating the log field.

Drop Nulls

If the column by this name is null, ignore this record. For example, if the user field is null, then the record cannot be correlated to a IdentityIQ identity and, therefore, cannot be used by IdentityIQ.

RACF Audit Log Collector

Transport Settings

The transportation settings are used to access the server where the log file containing the activity data resides.

Transport Type – depending on the transport type selected you will see the following:

local

If the log file containing the activity data is on the same server as IdentityIQ, no further connection-type information is required.

ftp

FTP User – a valid user name with authentication access to the FTP host.
FTP Password – the password associated with the FTP user.
FTP Host – the host where the log file resides.

scp

SCP User – a valid user name with authentication access to the SCP host.

SCP Password – the password associated with the SCP user.

SCP Host – the host where the log file resides.

SCP Private Key – the private key that is used to encrypt the collected data.

Log File Settings

The log file settings are used to define the query used to collect the activity data.

File Name

The name of the log file containing the activity data.

Lines to Skip

The number of lines to skip before starting the scan for activity information.

Filter Nulls

Skip lines that don't conform to the defined format.

CEF Log File

CEF Log File Transport Settings

The transportation settings are used to access the server where the log file containing the activity data resides.

Transport Type – depending on the transport type selected you will see the following:

local

If the CEF log file containing the activity data is on the same server as IdentityIQ, no further connection-type information is required.

ftp

FTP User – a valid user name with authentication access to the FTP host.
FTP Password – the password associated with the FTP user.
FTP Host – the host where the log file resides.

scp

SCP User – a valid user name with authentication access to the SCP host.
SCP Password – the password associated with the SCP user.
SCP Host – the host where the log file resides.
SCP Private Key – the private key that is used to encrypt the collected data.

CEF Log File Settings

The log file settings are used to define the query used to collect the activity data.

File Name

The name of the CEF log file containing the activity data.

Lines to Skip

The number of lines to skip before starting the scan for activity information.

Filter Nulls

Skip lines that do not conform to the defined format.

Multi-lined Data

A single record in this file spans multiple rows.

Regular Expression

A regular expression groups that can be used to tokenize each record in the file. The format of CEF Log File. For example, (\w\w\w\s\d\d\s\d\d:\d\d:\d\d)\s(.*)CEF:(.*)\|(.*)\|(.*)\|(.*)\|(.*)\|(.*)\|(.*)\|(.*)(.*)

CEF Log Fields

The log field settings are used to create the log fields based on the column headings in the log file.

Name

The name of the CEF log field to create based on a column name from the CEF log file.

Trim Value

Remove white space around the column name before creating the CEF log field.

Drop Nulls

If the column by this name is null, ignore this record. For example, if the user field is null, then the record cannot be correlated to a IdentityIQ identity and, therefore, cannot be used by IdentityIQ.

IdentityIQ uses connectors to extract data and transform it into a format it can read. A connector is a Java class that extends the IdentityIQ `AbstractConnector` class and implements the IdentityIQ Connector interface. Connectors provide the means by which IdentityIQ communicates with targeted platforms, applications and systems. Each application type requires different information to create and maintain a connection. For detailed connector information refer to the connector documentation delivered with IdentityIQ.