SailPoint IdentityIQ

Version 8.1

System Configuration Guide

Table of Contents

IdentityIQ Introduction

SailPoint IdentityIQ is an identity and access management solution for enterprise customers that delivers a wide variety of IAM processes-including automated access certifications, policy management, access request and provisioning, password management, and identity intelligence. Furthermore, IdentityIQ has a flexible connectivity model that simplifies the management of applications running in the datacenter or the cloud.

Compliance Manager — IdentityIQ Compliance Manager automates access certifications, policy management, and audit reporting through a unified governance framework. This enables you to streamline compliance processes and improve the effectiveness of identity governance, all while lowering costs.

Lifecycle Manager — IdentityIQ Lifecycle Manager manages changes to access through user-friendly self-service request and password management interfaces and automated lifecycle events. It provides a flexible, scalable provisioning solution for addressing the constantly evolving access needs of your business in a way that's both efficient and compliant.

IdentityAI – Integrating IdentityAI within IdentityIQ enables the delivery of Predictive Identity. IdentityAI is a rule based machine learning engine using identity graph technology to provide recommendations for access review and access request decisions. With IdentityAI enabled, you can also review access history for identity cubes, create dashboards that can be customized from an administrative perspective, and view peer groups within the IdentityAI user interface.

Privileged Account Management Module — IdentityIQ Privileged Account Management module provides a standardized approach for extending critical identity governance processes and controls to highly privileged accounts, enabling IdentityIQ to be used as a central platform to govern standard and privileged accounts.

Connectors and Integration Modules — IdentityIQ offers Integration Modules that support the extended enterprise IT infrastructure. Third party provisioning and service desk integration enable multiple sources of fulfillment to access change. Service catalog integration supports a unified service request experience with integrated governance and fulfillment. Mobile device management integration mitigates risk posed by mobile devices through centralized visibility, control and automation. And IdentityIQ's IT security integration provides enhanced security with improved responsiveness and controls.

Open Identity Platform — SailPoint's Open Identity Platform lays the foundation for effective and scalable IAM within the enterprise. It establishes a common framework that centralizes identity data, captures business policy, models roles, and takes a risk-based, proactive approach to managing users and resources. The Open Identity Platform is fully extensible, providing robust analytics which transforms disparate and technical identity data into relevant business information, resource connectivity that allows organizations to directly connect IdentityIQ to applications running in the datacenter or in the cloud, and APIs and a plugin framework to allow customers and partners to extend IdentityIQ to meet a wide array of needs. An open platform allows organizations to build a single preventive and detective control model that supports all identity business processes, across all applications-in the datacenter and the cloud. SailPoint IdentityIQ applies consistent governance across compliance, provisioning and access management processes, maximizing investment and eliminating the need to buy and integrate multiple products.

Password Manager — IdentityIQ Password Manager delivers a simple-to-use solution for managing user passwords across cloud and on-premises applications policies from any desktop browser or mobile device. By providing intuitive self-service and delegated administration options to manage passwords while enforcing enterprise-grade password, IdentityIQ enables businesses to reduce operational costs and boost productivity.

Amazon Web Services (AWS) Governance Module — Enables organizations to extend existing identity lifecycle and compliance management capabilities within IdentityIQ to mission-critical AWS IaaS environments to provide a central point of visibility, administration, and governance across the entire enterprise. This includes policy

discovery and access history across all organization accounts, provisioning AWS entities and objects, access review and certification, and federated access support.

SAP Governance Module — Improves the user experience by introducing a new integrated visual interface for navigating and selecting SAP identities and roles as part of IdentityIQ lifecycle management and compliance solution. SAP data is presented in a familiar hierarchy format that closely represents deployed system resources and organizational structures. New filtering capabilities enable more efficient browsing and selection of SAP data so tasks can be performed faster. Improved granular support for separation of duty (SOD) violation policies provides flexibility for customers to craft more detailed identity governance policies that include SAP role details such as T-Codes and Authorization Objects.

Chapter 1: System Setup

Use System Setup to configure the different options for IdentityIQ. To access System setup options, click the gear icon on the Navigation menu bar and select from the list of options that can include:

- “IdentityIQ Global Settings” on page 3
- “Lifecycle Manager Setup” on page 59
- “Compliance Manager” on page 51

Note: Do not open multiple tabs or browsers. Opening multiple tabs might overwrite changes made in the other.

Note: Because configuration options are based on your deployment, your available options can be different.

IdentityIQ Global Settings

From the Navigation bar, click the gear icon and then select **Global Settings**. Use the Global Setting index page to select the items you want to configure. The following table displays the available options.

Note: You must be a System Administrator to access this page.

Table 1—System Setup Globals Settings Page Descriptions

Page	Description
IdentityIQ	
IdentityIQ Configuration	Set default values for use with notifications, work item policy, object expiration, user interface preferences, and identity history. See "IdentityIQ Configuration" on page 4.
Login Configuration	Set an application other than IdentityIQ for authentication verification and select the automatic identity creation rule. See "Login Configuration" on page 17.
Identity Mappings	Specify the applications, and application attributes, from which the identity data, is derived. See "Identity Mappings" on page 26.
Account Mappings	Specify the account attributes to be used in filters and searches throughout the application. See "Account Mappings" on page 30.
Application Attributes	Define application attributes in addition to those provided by the connectors. See "Application Attributes" on page 35.
Entitlement Catalog Attributes	Define custom extended entitlement attributes and role types. See "Entitlement Catalog Attributes" on page 37.
Quicklink Populations	Configure the quicklink populations for IdentityIQ. See “Quicklink Populations” on page 38.
Forms	Configure forms for workflows, role provisioning policies, and application provisioning policies in IdentityIQ. See "Forms" on page 40.

Table 1—System Setup Globals Settings Page Descriptions

Page	Description
Role Configuration	Define custom extended role attributes and role type. See "Role Configuration" on page 41.
Scopes	Define scopes for use throughout your enterprise. See "Scopes" on page 45.
Time Periods	Define the time periods for use in activity searches. See "Time Periods" on page 47.
Audit Configuration	Specify the actions that are audited and stored in the audit logs. See “Audit Configuration” on page 48.
Electronic Signatures	Configure electronic signatures and their displayed meanings. See "Electronic Signatures" on page 48.
API Authentication	Import files into IdentityIQ. See "API Authentication" on page 48.
IdentityAI Configuration	**Note: This link is present only if you have purchased the IdentityAI product.** Connect IdentityIQ to the IdentityAI product. See “IdentityAI Configuration” on page 49.
File Access Manager Configuration	**Note: This link is present only if you have imported the File Access Manager module.** Configure IdentityIQ to connect to File Access Manager. See “File Access Manager Configuration” on page 50.
Import from File	Import files into IdentityIQ. See "Import From File" on page 51.

IdentityIQ Configuration

Use this page to set default values for use with notifications, work item policy, object expiration, user interface preferences, and identity history. This page contains the following tabs:

- “Mail Settings” on page 4
- “Work Items” on page 6
- “Identities” on page 7
- “Roles” on page 8
- “Password” on page 9
- “Miscellaneous” on page 11

Mail Settings

Table 2— System Setup - IdentityIQ - IdentityIQ Configuration - Mail Settings

Field	Description
Email Settings:	
Email Notification Type	Specify whether to send email using SMTP or to use a redirection email address or file name.

Table 2— System Setup - IdentityIQ - IdentityIQ Configuration - Mail Settings

Field	Description
Redirection Email Address	Specify the email address to which email is redirect if Redirecting is selected as the **Email Notification Type**. **Note: This setting is ignored if a Redirection File Name is set. This setting is primarily a test setting.**
Redirection File Name	Specify the name of the file to which email is redirect if Redirecting is selected as the **Email Notification Type.** **Note: This setting overrides the Redirection Email Address. This setting is primarily a test setting.**
Encryption	Select NONE, SSL, or TLS from the drop-down list.
Default SMTP Host	Specify a default mail host.
Default SMTP Port	Specify a default SMTP port.
Default From Address	Specify the address to be used as the From address for all notices automatically generated by IdentityIQ.
Username and Password	Enter the username and password required to access the SMTP host.
Maximum Email Retries	Specify the maximum number of times to retry sending emails if the SMTP server returns a temporary error. Set this to 0 to disable retries.
Suppress Duplicate Emails	Prevent the sending of multiple emails of the same type to the same recipient at one time. For example, if five work item reminders are sent to the same person at one time, they only receive an email for the first one. This option is enabled by default.
Email Templates:	
Specify the email template that corresponds with each notification type. Email templates are highly configurable.	
Email Task Alerts:	
Specify the configuration parameters in order to receive the status of different tasks after completion. These setting would be applicable in case of the email notification configured at the task level is disabled.	
Email Notification	Select a frequency for email notification to be sent upon task completion. **Disable** — no email notification sent on task completion **Warning** — send an email notification if the task results in a warning **Failure** — send an email notification if the task fails **Always** — always send an email notification upon task completion
Email Notification Template	Select a notification email template (Task Status) from the drop-down list.This option is disabled if **Email Notification** field is disabled.
Email Recipients	The list of users to receive the task completion notification. Use the drop-down arrow to display all identities, or type the first few letters of a name. select names from the list.

Work Items

Table 3— System Setup - IdentityIQ - IdentityIQ Configuration - Work Items Settings

Field	Description
Certification Related Work Item Policy:	
Days before expiration	Specify the number of days after which a work item should expire.
Days before expiration to send first reminder	Specify the number of days, before a work item expires, that IdentityIQ should begin sending the owner of that work item reminder notices.
Days between expiration reminders	Specify the frequency with which reminder notices should be sent to the owners of certifications and work items.
Number of notices before escalation	Specify the number of reminder notices that should be sent before the first escalation notice is sent to the manager of the owner of the assess certification or work item.
Send notification email on work item assignment	Select this option to send an email notification when a work item is assigned.
Send notification email on work item assignment removal	Select this option to send an email notification when a work item assignment is removed.
Allow priority editing on work items	Select this option to give work item recipients the ability to adjust the priority level of work items.
Work Item Archives:	
Work item types to archive	Select one or more work item types to be archived. Press the **<Ctrl>** key to select multiple items.
Work Item Rules:	
Inactive user work item escalation rule	Select the rule from the drop-down list for determining a new owner for work items from an inactive user.
Global work item forwarding rule	Use the drop-down list to select the rule used to determine general work item forwarding.
Self-certification work item forwarding rule	Use the drop-down list to select the rule used to determine work item forwarding in special cases. Allows for the specification of a fallback forwarding user in the case that configured automatic forwarding would cause self-certification. This rule only applies if a user has configured a forwarding rule, the rule does not apply when a pre-delegation rule causes self certification.

Identities

Table 4— System Setup - IdentityIQ - IdentityIQ Configuration - Identities Settings

Field	Description
Identity Risk:	
Number of Bands	Specify the number of colored bands, from 2 to 6, to display on all score card charts, graphs, and tables. These bands are used to indicate various levels of risk associated with ranges of Identity risk scores. Specify a number that best meets the needs of your enterprise.
Label	Select the default labels or create your own text label associated with the colored risk bank.
Range	Input the numeric risk score range associated with each risk band. Risk scores are determined by multiple contributing factors defined on the Configure Risk Scoring page. Refer to the *SailPoint IdentityIQ System Administration Guide* for more information.
Indicator	The indication color associated with the risk level.
Identity Attributes:	
Number of searchable attributes	Specify the number of attributes that can be configured for use as searchable attributes on the Identity Attributes page. This can be any number between 1 and 20. The default is 10. **Note: This number should match the number configured during the installation and deployment process. If no customization was performed during the installation and deployment process, the maximum number you can enter is 10.**
Index History Granularity:	
Identity history Group history	The increments at which to store history. For example, if the Identity history is set to 'Week', snapshots are preserved on a weekly basis. This means that when a snapshot is taken it overwrites any snapshots taken within the previous week (7 days). Any snapshot that is older than 7 days is saved.
Identity Snapshots:	
Snapshot frequency in days(2 equals every second day)	Specify the frequency with which identity snapshot should be taken. Identity snapshots are used to build the risk score card history that can be used to track trends and patterns for individual users, groups, departments, and your entire organization.
Account Attributes:	
Number of searchable account attributes	Specify the number of attributes that can be configured for use as searchable attributes. This can be any number between 1 and 20. The default is 5. **Note: This number should match the number configured during the installation and deployment process. If no customization was performed during the installation and deployment process, the maximum number you can enter is 5.**
Business Processes:	

Table 4— System Setup - IdentityIQ - IdentityIQ Configuration - Identities Settings

Field	Description
Identity update	Select which business process is executed when an identity is edited in IdentityIQ. This can perform role assignment approvals and send provisioning requests.
Identity refresh	Select which business process is executed when an identity is refreshed in a background task. This might perform role assignment approvals and send provisioning requests.
Identity Correlation	Select which business process is executed when an manual correlation of accounts is done.

Roles

Table 5— System Setup - IdentityIQ - IdentityIQ Configuration - Roles Settings

Field	Description
Role Sunrise/Sunset Dates:	
Enable Sunrise/Sunset Dates on Role Assignment	Enable the ability to set activation and deactivation dates on roles when they are assigned. Activation and deactivation dates can be used to grant temporary access to sensitive roles.
Enable Sunrise/Sunset Dates on Role Activation	Enable the ability to insert activation and deactivation events into roles from the role modeler. Activation events are used to automatically activate or deactivate roles using business processes.
Days before Sunset expiration to send notification	Send a notification to both the requestor and the requestee of the role or entitlement, when access is about to expire. This value determines when the notification is sent. To disable notifications, enter 0. The email template to use for notifications is configured on the **Mail Settings** tab in the **For notice of deprovisioning of sunsetted roles and entitlements** field.
Business Process Editor:	
Role create, update, and delete	Select which business process is executed when roles are created, modified, or deleted in the role modeler.
Schedule role activation	Select which business process is executed when a scheduled role assignment becomes due. This assigns the role and can perform provisioning.
Schedule role/entitlement assignment	Select which business process is executed when a scheduled role assignment or de-assignment becomes due. This de-assigns the role and can perform provisioning.
Additional Role Options:	
Show option to allow multiple application accounts	Enables an option on the Role Management page that enables a role to specify its own target account, or create a new account, during a role request, even if it is required by another role and included in that roles required roles list. If this option is not enabled, required roles are assigned to the same account as the top-level role.

Table 5— System Setup - IdentityIQ - IdentityIQ Configuration - Roles Settings

Field	Description
Show option to allow multiple assignments	Enable an option on the Role Management page that enables a role to be assigned to the same identity multiple times. This option is only available on assignable role types.
Allow multiple assignment for all assignable roles	Make all assignable role types available for multiple assignment to the same identity. This setting supersedes the settings on the individual role definitions.
Allow propagation of role changes	Enables a role change to propagate to all identities that have the role assigned.
Retain assigned entitlements when detected roles are removed	Do not remove assigned entitlements from an identity when a detected role with which they are associated is removed from that identity.
Retain assigned entitlements when assigned roles are removed	Do not remove assigned entitlements from an identity when an assigned role with which they are associated is removed from that identity.

Password

Use this tab to define the password policy for IdentityIQ. All of the users must set up their passwords based on the policy created on this tab.

Use the Define Character Types dialog to define a custom set of character that are allowed in passwords. These can be used to match password requirements for specific application types. Click **Define Character Types** to open the dialog and enter character sets by category, such as **Digits**, **Uppercase Characters**, **Lowercase** or **Non-English Characters**, **Special Characters**. All characters are allowed if these fields are empty.

Refer to the *SailPoint IdentityIQ System Administration Guide* for more information.

Table 6— System Setup - IdentityIQ Configuration - Password Settings

Field	Description
Configuration:	
Enable one-way hashing of secret values	**Note: You must run the Encrypt Sensitive Data Task after selecting this option to convert any saved values from encrypted to hashed.** These values include passwords, password history, and authentication questions. When this option is enabled, specific password policy options are disabled. When this option is selected, all values are hashed instead of encrypted. For more information, see “Data Encryption” on page 125.
Number of hashing iterations	The number of iterations performed in the hashing algorithm.
Password Policy:	

Table 6— System Setup - IdentityIQ Configuration - Password Settings

Field	Description
Minimum number of characters	The minimum number of characters, letters or digits, required for a valid password.
Maximum number of characters	The maximum number of characters, letters or digits, allowed in 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 required for a valid password. Applicable character types are upper case, lower case, digits, and special characters. If no value is set, all of the character type constraints must be met.
Minimum number of digits	The minimum number of digits required for a valid password.
Minimum uppercase letters	The minimum number of uppercase letters required for a valid password.
Minimum lowercase letters	The minimum number of lowercase letters required for a valid password.
Minimum special characters	The minimum number of special characters required.
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” is valid, “clouddd”, “cloooud” and “cccloud” are invalid. For “cclooudd” invalid password, the following error message is displayed: Password should not contain more than 2 occurrence(s) of the repeated characters. For “Clouddd” invalid password, the following error message is displayed: Password should not contain more than 2 consecutive repeated characters
	The maximum number of occurrences of repeat characters allowed in a valid password. For example, if this option is set to 1, “happy123” is valid, however, “happy123dd” and “happpy123” are not.
Password history length	The number of previous passwords stored by IdentityIQ. This number includes the current password so if the length is two, the history is the current password and one other. If the length is set to zero there is no history.
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 for the new password. Can be used to ensure that not just the first or last character is changed. Select **Case sensitive check** to ensure that more than just the case is changing in the new password.

Table 6— System Setup - IdentityIQ Configuration - Password Settings

Field	Description
Days until expiration for manually set passwords	The number of days until a password set manually expires. If the days are zero passwords do not expire.
Days until expiration for generated passwords	The number of days until a password set by the identity create rule during aggregation expires. If zero the days are zero passwords do not expire.
Minimum Hours between password changes	The minimum number of hours that must past before a user's password can be changed again.
Validate passwords against the password dictionary	Ensures that the password to be created is unique.
Validate passwords against the identity's list of attributes	Check the new password for validity against the attributes assigned to the identity.
Require users to enter their current password when setting a new password	Require users to enter there current password before creating a new password.

Miscellaneous

Table 7— System Setup - IdentityIQ - IdentityIQ Configuration - Miscellaneous Settings

Field	Description
Other Object Expirations:	
Days before snapshot deletion	Specify the number of days to keep an identity snapshot in the system before it is deleted. Identity snapshots are used to build history.
Days before task result deletion	Specify the number of days to keep task results on the Task Results page before removing them from the system.
Days before certifications are archived	Specify the number of days after which to archive certifications. Leave the settings at zero (0) to never archive certifications. **Note: Certification archives are not visible from the IdentityIQ GUI. It is recommended that you do not change the default setting at this time.**

Table 7— System Setup - IdentityIQ - IdentityIQ Configuration - Miscellaneous Settings

Field	Description
Days before certification archive deletion	Specify the number of days to maintain the certification archive before deleting certifications records. Leave the settings at zero (0) to never delete certifications archives. **Note: Certification archives are not visible from the IdentityIQ GUI. It is recommended that you do not change the default setting at this time.**
Minutes before object locks are released	Specify the number of minutes to elapse before releasing an object lock. Leave the settings at zero (0) to have no time delay when objects are released.
Days before provisioning request logs expire	Specify the number of days to maintain provisioning request logs before deleting them. Leave the settings at zero (0) to never delete provisioning request logs.
UI Preferences:	
Disable Role Modeler Tree View	Disable the tree view on the Role Manager page. Disabling the tree view might enhance performance on that page.
Maximum Roles Page Size	The maximum number of roles to display per page on the **Role Management** page.
Show unsupported browser message	Display a message when an unsupported browser is used.
Accessibility: Color Contrast	Enable color contrast throughout the entire IdentityIQ instance.
Syslog Settings:	
Enable syslog	Enable the syslog.
Level at which syslog events will be stored	Select the lowest level of even which is stored in the syslog. Choose from FATAL, ERROR, and WARN.
Days before syslog event deletion	Input the number of days an event in the syslog must remain before becoming eligible for purging.
Provisioning Transaction Log Settings	
Enable Provisioning Transaction Log	Enable the Provisioning Transaction table and begging logging all, or some, of the provisioning action within IdentityIQ.
Maximum Log Level	The level at which transactions are logged based on their completion status. Success — all transaction are logged Retry — transactions that did not succeed and are in either the retry or failed state Failure — only log transaction that have failed and are setup for retry

Table 7— System Setup - IdentityIQ - IdentityIQ Configuration - Miscellaneous Settings

Field	Description
Days before provisioning transaction event deletion	The number of days before a provisioning transaction is removed from the table.
File Preferences:	
Temporary Directory	Input the path to a default temporary director for use by IdentityIQ. This is the directory where IdentityIQ stores temporary files, such as log files, during processing.
System Help Settings:	
Help Contact Email Address	Input an email address of a user responsible for supporting IdentityIQ in your enterprise. The email account is accessible from an **Email Help** button displayed at the bottom of some pages.
Localized Object Attributes:	
Allow applications to be configured with multi-language descriptions	Allow applications to be configured with multi-language descriptions. See “Multi-language Description Files” on page 15.
Allow roles to be configured with multi-language descriptions	Allow roles to be configured with multi-language descriptions. See “Multi-language Description Files” on page 15.
Allow policies to be configured with multi-language descriptions	Allow policies to be configured with multi-language descriptions. See “Multi-language Description Files” on page 15.
Allow entitlements to be configured with multi-language descriptions	Allow entitlements to be configured with multi-language descriptions. See “Multi-language Description Files” on page 15.
Multi-Languages Descriptions: **Note: You must add all supported languages to the <locale-configure> section of the faces-config.xml file before the application properly recognizes the languages.**	
Default Language	Select the language to use as a default from the list of supported languages.
Supported Languages	Enter the languages that your instance of IdentityIQ supports.
Business Processes:	

Table 7— System Setup - IdentityIQ - IdentityIQ Configuration - Miscellaneous Settings

Field	Description
Entitlement Update	Select the business process to execute when a managed entitlement or group is created or edited.
Password Intercept	Select the business process to execute when a password change interception event is received.
Caches:	
Enable asynchronous policy and role cache refresh	Disable the immediate cache refresh with each Lifecycle Manger request. When you enable this option, IdentityIQ does not check for changes to policy and role objects. When a Lifecycle Manager request is submitted, the cache is refreshed immediately. Using this option can speed the request process. However, the effects of a recent policy or role change might not display for a few minutes.
Reports:	
CSV Delimiter	The character used as the CSV delimiter when exporting report results. Comma is used by default.
Plugin Settings:	
Prohibit scripts from accessing plugin-loaded classes	**Note: All beanshell executions are referred to as scripts.** Restrict the access to classes loaded by plugins. Without this restriction, all class are available in IdentityIQ.
Relax strict declaration enforcement	Enable IdentityIQ to work fully with plugins that were created without explicitly declaring classes for export. By default, for a fresh installation of IdentityIQ this option is not selected. For an upgraded installation of IdentityIQ, this option is selected if plugins exist.
Attachment Settings: **Note: IdentityIQ does not perform file content validation or verification on attachments. It is your responsibility to ensure that only files that do not violate security policies within your environment are included as attachments.** **Note: Attachments are only allowed on single-user requests.** **Note: Attachments are only available for manual access requests.** See “Access Request Attachments” on page 16 for more information.	
Enable Attachments	Enable the attachments feature. Allow users to add attachments to access requests.
Maximum file size (MB)	Maximum file size for any single attachment up to 20 MB. Maximum attachment size limits can be adjusted by a system administrator using the `attachmentsMaxFileSizeLimit` key in the system configuration file.

Table 7— System Setup - IdentityIQ - IdentityIQ Configuration - Miscellaneous Settings

Field	Description
Supported file types	Comma separated list of file types. The dot prefix is not required.
Configuration Rules	**Note: Only the rules selected in this list are run during an access request.** This list contains all of the attachment configuration rules available in your installation of IdentityIQ. Use the Ctrl or Shift keys to select multiple rules.

Privileged Account Management

The SailPoint IdentityIQ Privileged Account Management Module (PAM) extends identity governance processes and controls to highly privileged access, enabling you to centrally manage access to privileged and non-privileged accounts.

Talk to your SailPoint representative or refer to the SailPoint IdentityIQ Privileged Account Management Module Guide for more information.

Multi-language Description Files

Some escaped HTML characters are not recognized and do not display in descriptions if they are formatted using those characters. You must ensure that all files are formatted correctly before importing them into IdentityIQ and referencing them from the product. Use the following examples to format the HTML correctly:

```
test (to appear in bold) - <b>test</b>
<test> - &lt;test&gt;
<test> (to appear in bold) - <b>&lt;test&gt;<b\>
<<test>> - &lt;&lt;test&gt;&lt;
“test” - &quot;test&quot;
‘test’ - ‘test’
&test - &test
```

Rule Editor

The Rule Editor page enables you to edit any existing rule to your specifications. Click the “**...**” icon next to a rule drop-down list to access the rule editor throughout IdentityIQ. Choose to either create a new rule, or edit an existing rule structure.

The Rule Editor panel includes the following items:

Table 8—Rule Editor Panel Field Descriptions

Option	Description
Copy from an existing rule	Select an existing rule from the drop-down list. This option is available if you did not select a rule from the drop-down list on the previous page.
Code input area	Field where code is input. IdentityIQ recognizes BeanShell programing language. You can edit code from an existing rule or create a new one from scratch.
Description	Enter the description of your new rule.
Rule Name	Enter the name of your rule.
Rule Type	Non-editable field which displays the type of rule (for example, Violation).

Table 8—Rule Editor Panel Field Descriptions

Option	Description
Return Type.	Non-editable field which displays the type of return (for example, PolicyViolation).
Arguments.	Non-editable field which displays the arguments used in the rule (for example, log, context, state, etc.).
Returns.	Non-editable field which displays the type of return the rule executes (for example, Violation).

When you have completed your rule edits, click **Save** to return to the previous page. The new rule is now available from the drop-down list.

Access Request Attachments

Note: IdentityIQ does not perform file content validation or verification on attachments. It is your responsibility to ensure that only files that do not violate security policies within your environment are included as attachments.

Note: Attachments are only allowed on single-user requests.

Note: Attachments are only available for manual access requests.

The attachments feature enables users to add attachments to single user access requests. For example you could attach training certificates or a notarized document of authorization.

By enabling attachments on the Configure IdentityIQ Settings -> Miscellaneous tab, you are enabling, but not requiring, any user to add an attachment to any single user access request. When the feature is enabled, requests display the attachment icon, paper clip, on each item in a request, but the icon is only active if an attachment is allowed for that item. When you click the icon, the attachment overlay is displayed and you can add an attachment by dragging and dropping or uploading a file.

Attachments are controlled through AttachmentConfig rules. If there are no AttachmentConfig rules for an item, or they all have null or empty prompts, the attachment overlay contains no additional information.

Attachment Configuration

Attachments are controlled through the AttachmentConfig rules. Each of these rules is run with every request made. Use the AttachmentConfig rules to require attachments for specific access request scenarios and customize the prompts displayed on the attachment overlay. When an attachment is required, the word required is displayed with the attachment icon and an error is displayed if a request is submitted without an attachment.

Activate the attachment configuration rules to run with access requests by selecting them from the **Configuration Rules** list on the Global Settings -> IdentityIQ Configuration -> Miscellaneous tab under the gear icon.

Import attachment configuration rules using the System Settings -> Import from File page under the gear icon.

To remove an attachment configuration rule from IdentityIQ, first de-select that rule from the **Configuration Rules** list and then delete the rule object.

These rules can be as simple or complex as the needs of your organization require.

These rules contain the following inputs:

- requestor — the user making the request
- requestee — the user for whom the request is being made
- requestItem— the item being requested
- action — the request action (add or remove)

Each attachment configuration rule is run once for each item being requested and returns a list of configuration objects.

The fields of an attachment configuration object are:

- required — boolean (true, false) where true means an attachment is required
- prompt — string – the prompt that is displayed in the attachment overlay when attaching files to this request item

Multiple attachment configuration objects can be associated with a single request item. In this case, the prompt strings are concatenated on the attachment overlay.

A file containing an example of attachment configuration rules is included in the IdentityIQ installation package. The `examplerules.xml` file is located in the `IdentityIQ_HOME/WEB-INF/config` directory.

Prune Unassociated Attachments

In rare cases attachments that are not associated with an access request might end up getting loaded into the database. IdentityIQ provides two system maintenance task to prune those attachments and clean them out of your database. Refer to the *SailPoint IdentityIQ System Administration Guide* for more information on using the System Maintenance and System Maintenance Object Prune tasks.

Login Configuration

Use the Login Configuration page to set an application for authentication verification. For example, if all of the users in your organization are set up with roles and authorization in an LDAP server, use that server to verify users logging into IdentityIQ. Login Configuration has the following tabs:

- “Login Settings” on page 17
- “User Reset” on page 19
- “Multi-Factor Authentication” on page 21
- “SSO Configuration” on page 24

Authentication Method Processing Order

IdentityIQ attempts to authenticate users by all enabled methods before reporting login failure. The methods are executed in this order, skipping any disabled methods:

Note: If configured, Multi-Factor Authentication follows the initial user authentication through any of these means.

1. Single Sign On (Rule-based or SAML)
2. Pass-Through Authentication
3. Internal IdentityIQ Authentication

Login Settings

Use the Login Settings tab to configure general settings for login criteria.

Note: Any user discovered by an aggregation task displays in the identities lists and can be assigned work items. Before a user can access IdentityIQ and the work item, they must be validated by an authentication verification server.

Use Auto create user rules when adding users to the application. The first time a user logs into the application, and is verified by the pass-through server, the **Auto create user rules** creates an IdentityIQ user based on specifications defined in this rule. Those rules are applied each time the user accesses product.

The following table describes the login settings.

Table 9— System Setup - IdentityIQ - Login Configuration - Login Settings

Field	Description
Pass through application	Specify an application to use as the authentication verification server for all users logging into IdentityIQ.
Auto create user rule	Specify an auto create user rule to use when creating IdentityIQ identities based on account attributes discovered during aggregations. Note: Click the "..." icon to launch the Rule Editor to make changes to your rules if needed. See "Rule Editor" on page 15
Login error style	**Note: If you select Simple and are using the Lockout feature, users that are locked out do not receive a message providing that information.** Select a login error message style. **Simple** — shows an error with no information about what is incorrect. **Detailed** — provides information about the incorrect part of the login. For example, Invalid password for user admin.
Login after timeout returns to	Specify how navigation is handled after a session times out and you log back in to that session. If checked, the Home page is displayed. If not, the session returns to the page that was viewed at the time of the timeout.
Enable Authorization Lockout	Enable a lockout period for users who enter the wrong authorization information. Use the options that display to set the lockout parameters. **Note: This option is only associated with the IdentityIQ password. It does not apply to the pass through authentication application. For example, if a user is locked out of directly logging into IdentityIQ, but they enter the correct information on the pass through authentication server, they are allowed into the application.**
Number of Unsuccessful Login Attempts before lockout	Specify the number of login attempt failures allowed before the user is locked out of IdentityIQ.
Number of minutes a user will be locked out due to unsuccessful login	Specify the number of minutes a user is locked out of IdentityIQ before they can attempt to login again.

Table 9— System Setup - IdentityIQ - Login Configuration - Login Settings

Field	Description
Enable Protected User Lockout	Select this if you want users marked as "Protected" (such as the default **spadmin** user) to be treated the same as other users in authorization lockout. Leave it unchecked if you do not want protected users to be subject to lockout. By default, only the **spadmin** user is marked as protected; if there are other users you want to protect from lockout, you can make them protected by adding a `protected="true"` flag to the user's Identity object in the Debug Pages.

User Reset

User Reset has the following options:

- Enable Authentication Questions — displays a **Forgot Password** link on the Login page and uses answers to pre-defined questions to authenticate a user's identity
- Enable SMS Reset — displays a **Forgot Password** link on the Login page and uses Short Message Service (SMS) to send the user a text message with a verification code

Authentication Questions

Note: The Authentication questions and settings are associated with the password set on the password application. These are not associated with a direct logon to IdentityIQ.

Authentication questions confirm a user's identity if they have forgotten their IdentityIQ password and the environment is configured to enable the question authentication feature. Question authentication is enabled using the **Enable Forgot Password** select box on the Login Settings tab.

These questions display when you click the **Forgot Password** link off of the Login page during the authentication process.

The Questions list can contain tags from the properties file configured when your IdentityIQ instance was deployed, text entered directly on this tab, or a combination of both. Mapping tags from a properties file is generally used for internationalization purposes.

Click the plus (**+**) icon to add a new question and the minus (**-**) icon to remove a question. You can enter as many questions as you deem necessary. A user who forgets their password must answer the designated number of the questions in the list. The number of questions a user must answer for authentication is defined in the Settings section below.

When a user clicks the **Forgot Password** link and then selects and answers the authentication questions, **by default the user's answers are shown in plain text** as they are typed in the UI. If you want to obscure the users answers with asterisks as they are typed, add this entry key to IdentityIQ's **SystemConfiguration** object. (This is done in the Debug Pages.)

<entry key="obscureAuthAnswers" value="true"/>

Use the **Settings** section to configure behaviors for password attempts.

SMS Reset

Before you set up SMS Reset, you need the following items from twilio.com:

- an active Twilio account
- Twilio ID
- Twilio credentials (authentication token)
- From phone number configured on account

The following table describes the password reset settings.

Table 10— System Setup - IdentityIQ - Login Configuration - Password Reset Settings

Field	Description
Authentication Question Configuration:	
Number of questions asked to authenticate an identity	Specify the number of questions that must be answered correctly in order to reset the password.
Number of authentication answers a user must have defined in IdentityIQ	Specify the number of authentications that must provide to set up password reset.
Prompt users for answers to unanswered authentication questions upon successful login	Adds an extra layer of security to logon screen. Select to have users prompted for answers until they define the required number, as defined in Edit Preferences page or if questions are added or changed. When enabled, users are automatically redirected to the Answer Authentication Questions page upon successfully entering user name and password.
Maximum number of unsuccessful authentication attempts before IdentityIQ lockout	Specify the number of failed authentication answer attempts before the user is locked out of IdentityIQ. **Note: After the maximum number of unsuccessful attempts, the SMS token is no longer accepted and the user must request a new code.**
Number of minutes a user will remain locked out due to unsuccessful authentication	Specify how long a user is locked out after the specified number of failed authentication question answer attempts is exceeded. A user with the proper capability can overwrite the lockout period.
SMS Reset Configuration:	
Twilio Account ID	Enter the account ID you receive from Twilio when you set up your company Twilio account.
Twilio Authentication Token	Enter the authentication token you receive from Twilio when you set up your company Twilio account.

Table 10— System Setup - IdentityIQ - Login Configuration - Password Reset Settings

Field	Description
'From' Phone Number	Specify the phone number to use for sending the SMS message. **Note: This phone number must be configured as the from number on your Twilio account.**
Phone Number Attribute on Identity	Select the identity attribute that represents the mobile phone number. To define a new identity attribute, see "How to Add or Edit Account Attributes" on page 32. **Note: For a user to reset their password using the SMS Reset feature, the field associated with their mobile phone number must contain a complete number including the area code. Using E.164 number formatting for all phone numbers in the "To" and "From" fields is strongly encouraged.** For more information, see "SSO Configuration" on page 24.
Verification Token Timeout (minutes)	Specify how long the user's reset token is valid (in minutes).
Throttle requests at a rate of 1 per N minute(s)	Specify the limit of request that can be made in a certain amount of time. For example, limit the requests to 1 every N minutes.
Maximum Failed Attempts	After reaching the maximum failed attempts, a user cannot verify a reset token until that token expires and a new token is requested.

Multi-Factor Authentication

Multi Factor Authentication (MFA) adds an additional layer of security by requiring users to use multiple methods to authenticate their identity before they can log in to IdentityIQ. IdentityIQ supports the following MFA options:

- RSA Workflow
- Duo Workflow

To access MFA Login Configuration settings in IdentityIQ, click the **gear** icon in the menu bar and select **Global Settings > Login Configuration > Multi Factor Authentication** tab

This section includes the following topics:

MFA Prerequisites

Note: To use Duo, you must follow the Duo Auth API instruction to enable the AuthAPI in the Duo Admin Panel. To locate the Duo API documentation, go to https://duo.com, search for Auth API documentation and then follow the steps for adding Duo two-factor authentication.

MFA User Process Flow Overview

The basic process flow for using MFA to log in to IdentityIQ includes the following steps:

1. The user enters a valid username and password at the IdentityIQ login screen.
2. The IdentityIQ MFA workflow begins and displays the MFA provider's login page or process for login. If a user is assigned to multiple providers, the user must select a provider from the provider list before proceeding to the provider's login page.
3. The user completes the authentication process for their MFA provider.
4. The user is logged in to IdentityIQ and the Home page displays.

MFA Configuration Process Flow Overview

The basic process flow for configuring MFA for IdentityIQ includes the following steps:

1. Use a pre-defined MFA workflow or choose to create a custom workflow.
2. Install the workflow.
   - Import the workflow.
   - Configure the workflow as a business process.
   - Enable the populations to use with MFA.
3. Save your MFA configuration.

For more information, see "How to Install a Multi-Factor Authentication Workflow - DUO Example" on page 22

Multi-Factor Authentication Workflows

Each MFA provider has its own flow and process. MFA Providers contain the populations and providers are configured from an existing list of DynamicScopes/Populations. Workflows of type **MultiFactorAuthentication** can enable Multi-Factor Authentication for a particular provider.

Pre-defined workflows are provided. These workflows use existing pre-configured applications to perform Multi-Factor Authentication

You can choose to create a custom workflow. See "Custom Multi-Factor Authentication Workflows" on page 24.

How to Install a Multi-Factor Authentication Workflow - DUO Example

The following workflows are provided, however they are not installed by default. These workflows use existing pre-configured applications to perform Multi-Factor Authentication. The provided workflows are located:

- `WEB-INF/config/workflow_MultiFactor_DUO.xml`
- `WEB-INF/config/workflow_MultiFactor_RSA.xml`

Note: The following instruction are specific to using DUO as your MFA application. You can use these instructions to install RSA by changing the DUO-specific items to RSA. As noted in the instructions, you do not need to add API authentication credentials for RSA.

1. Review any prerequisites. See "MFA Prerequisites" on page 21.
2. To import the workflow, you can use the **Import From File** function in the **Global Settings** menu or use the IdentityIQ console. To use the IdentityIQ console, open the console and use the following command:

   ```
   import workflow_MultiFactor_DUO.xml
   ```

3. Configure the workflow as a business process:

a. Login to IdentityIQ using an administrator account and navigate to **Setup > Business Processes**.
b. Click the workflow named **MFA DUO**
c. Click **Process Variables**
d. Select a pre-configured application, of type **Duo**, for the field **Duo Application Name**. The workflow reads new properties added to the application used to authenticate with the Duo cloud Authentication service.
e. Click **Save**.

4. Use the following steps to add Duo authentication API credentials:

   IMPORTANT! These steps are not necessary for the MFA RSA workflow because RSA uses existing API credential information already configured in the RSA Application.

   a. Navigate to **Applications > Application Definition**.
   b. Select the Application of type Duo you configured in the previous step.
   c. Click **Configuration**.
   d. Complete the **Admin API Credentials** section using the credentials you obtained from the Duo Admin Panel.

      The first time you set up a Duo application, you must enter the Admin API information received from Duo after you completed Step . in “How to Install a Multi-Factor Authentication Workflow - DUO Example” on page 22. If you are modifying a previously configured Duo application, the Admin API credentials should already be configured.
   e. Click **Save**.

5. Next, enable a population of users that must use Multi-Factor Authentication to authenticate using the following steps:
   a. Click the **gear** icon.
   b. Navigate to **Global Settings > Quicklink Populations**.
   c. Verify you have an existing population of users you want to authenticate using Multi-Factor Authentication.

6. The population you enabled can allow a user in the population to request access for other users. If you do not want a user have that capability, you can create a new QuickLink population. You must select **No one** in the section **who can members request for?** when you create the new QuickLink population. This configuration separates Request Access type Quicklink Populations from Multi-Factor Authentication Populations.

7. Next, associate the population to the Multi-Factor Authentication workflow using the following steps:
   a. Click the **gear** icon and navigate to **Global Settings > Login Configuration > MFA tab**.
   b. Check the box for the MFA Workflow you want to enable.
   c. Add any populations to the multi-select list you want to enable for this MFA workflow.
   d. Click **Save**.

Custom Multi-Factor Authentication Workflows

Implementors can create custom Multi-Factor authentication workflows. Any workflow of type **MultiFactorAuthentication** displays in the MFA Configuration page. If you choose to create a custom workflow, review the following information:

- Adding an error message to the workflow case using:

  ```
  wfcase.addMessage(new Message(Type.Error, "An error has occurred that prevents
  Multi-Factor Authentication"))
  ```

  This adds an error to the workflow case and signals to the Multi-Factor framework the user should not be logged in.
- A workflow that was not marked **complete** will signal Multi-Factor authentication has failed. During normal workflow execution, if a workflow has not produced an error, the workflow is automatically marked complete.

SSO Configuration

IdentityIQ supports two different options for single sign-on (SSO) configuration, rule-based and SAML. SSO streamlines the login process for users even further than pass-through authentication by enabling the user to bypass signing in to each system, once they have completed the initial sign-on to the authenticating application.

SSO Configuration has the following options:

- Enable Rule-Based Single Sign-On (SSO) — uses rules for Single Sign-On and Validation
- Enable SAML Based Single Sign-On (SSO) — uses Security Assertion Markup Language (SAML) as an authentication protocol

  Note: To access the IdentityIQ Login page directly when Single Sign-On is configured, use a supported browser and enter http://<iiq server>/iiq/login.jsf?prompt=true.

IdentityIQ supports specifying both types of SSO in the same installation's login configuration. The order in which they are consulted during user authentication will be determined as follows:

- If an ssoAuthenticators attribute is specified in the SystemConfiguration object, it will specify the configured SSO options in a CSV list, and the options will be checked in the order they are specified
- If that attribute is not present, SAML SSO will be used first and then rule-based SSO

Rules-Based SSO

In rule-based Single Sign-On (SSO) configurations, when the user accesses the IdentityIQ web application, the authentication source recognizes it as a secure resource, requires the user to authenticate to it (if the user has not already done so), and passes a “token”, containing contextual information, in the HTTP header to IdentityIQ. The SSOAuthenticationRule validates that information and maps the user to the appropriate IdentityIQ Identity.

SAML-Based SSO

In SAML SSO, the authorization request can be initiated with the Service Provider (the application itself - IdentityIQ) or with the SSO authentication application (known as the Identity Provider). In either case, the Identity Provider handles authentication of the user and provides a signed XML <Response>, or Assertion. This response contains information that IdentityIQ can match to an identity to determine the user's proper authorization to IdentityIQ functionality.

IdentityIQ has the following SSO configuration areas:

- Identity Provider (IDP)
- Service Provider (SP)
  - Identity ID / Issuer — string that represents how each side (IDP or SP) refers to itself
  - Login URL (also known as the SSO Service or SSO Login URL) — the URL on the IDP which understands SAML.
  - Public X.509 Certificate — the X509 certificate public key of the EDP.
- Assertion Consumer Service (ACS) — URL on the SP which understands SAML

Note: You can have an IDP EntityI D / Issuer and a Service Provider (SP) Entity ID / Issuer. Both IDs are very important for SAML 2.0 flows.

The following table describes the SSO settings.

Table 11— System Setup - IdentityIQ - Login Configuration - SSO Settings

Field	Description
Rule Based SSO:	
Single Sign-On Rule	Specify the rule to use when authorizing users through and single sign-on system, such as SiteMinder. **Note: Click the "..." icon to launch the Rule Editor to make changes to your rules if needed.** See "Rule Editor" on page 15.
Single Sign-On Validation Rule	Specify the rule to use to verify a single sign-on session to make sure a stale session is not actually a different user. The rule type (SSOValidation) runs on every request. If the request is valid, it returns null. If it returns a string, that string is an indication of an error and is used in the error that is displayed in the logs. If the session is invalidated, the request is redirect to logoutUrl configured in web.xml. This is designed to be used with the Single Sign-On Rule.
SAML Based SSO: Identity Provider Settings	
Entity ID / Issuer	Unique identifier defining the organization (IdentityIQ) to the IdP. This ID is usually the URL or domain name for the organization. For example, `https://identityiq-server.your-domain.com:your-port/identityiq` **Note: If you use the standard https port for communication, the :your-port is not necessary.**
SSO Login URL	IdP SAML SSO URL. Specify the url of the IdP SSO service provider. You can configure IdentityIQ to interact with other Identity Providers (IdP)s. You can obtain this address from your IdP. **Note: If the Identity Provider Issuer is not set, the configuration default to Identity Provider Single Sign-On service URL.**

Table 11— System Setup - IdentityIQ - Login Configuration - SSO Settings

Field	Description
Public X.509 Certificate	Select the NameId Format specified by the IdP.
Identity Provider Issuer URL	Specify the Unique Identifier that defines the IdP to IdentityIQ. This identifier is often in the form of a url but does not have to be. **Note: The Identity Provider Issuer URL field is only necessary if the SAML response does not contain an Issuer value that does not match the leading characters of the Identities Provider SSO Server URL field. For example,** Identity Provider SSO Server URL = `https://idp.your-domain.com/SSOApp/SSOLogin` SAML Response Issuer field = `https://idp.your-domain.com/SSOApp`
SAML Based SSO: Service Provider Settings	
SAML Response URL (Assertion Consumer Service)	Specify the IdentityIQ url where the SAML is to be accepted. For example, `https://identityiqserver.your-domain.com:your-port/identityiq/home.jsf`
Binding Method	Select **HTTP POST** or **HTTP Redirect** for the communications scheme.
NameId Format	Select the name format from the list. The IdP provides the formats listed in drop-down box.
SAML Correlation Rule	Select a rule to use to match a SailPoint identity with IdP results.

Identity Mappings

Use the Identity Attributes page to view and edit the identity attributes information for your configuration. These attributes are used throughout the product for certifications, searches, and to collect and correlate identity data from applications.

IdentityIQ also supports the use of Robotic Process Automation (RPA) or bot, an application that can perform automated tasks, especially simple, repetitive tasks such as requesting access and managing identities.

Bots require effective governance just as traditional identities do:

- The need to manage bots in your organization or under your control. You need to be able to see all the bots, along with their access and have the ability to add, remove access to bots.
- Your organization might have bots that do password resets for certain populations in the organization. You need to make sure that the bots have the right access and are the right version to do their job.
- The need to show auditors that your organization has owners who are accountable for managing bots, as part of certification.
- The need for an ability to define policies to ensure that bots do not get too much access.
- The need for an ability to define lifecycle events on bots, so that you can enforce controls on when bot access changes or when bots are retired.

IdentityIQ's governance capabilities for bots includes the abilities to:

- Manage bots and their attributes
- Request access for bots
- Certify bots

The Identity Attributes page contains the following information:

Table 12— System Setup - IdentityIQ - Identity Mapping - Identity Attributes Descriptions

Column	Description
Attribute	The display name of an identity attribute derived from the attribute and its associated application in the Primary Source Mapping column. The following attributes are required by IdentityIQ to perform correctly: ID manager email firstname lastname **Manager** and **role** are system attributes that are configured for grouping. However, you can use any identity attribute or grouping by defining it as a group factory in the Advanced Options.
Primary Source Mapping	The first of the list of application/attribute pairs from which employee attributes are derived. If the required data is unavailable on this primary source, the collection process continues down the list of configured sources until the information is found. Set up the list of sources on the Edit Identity Attributes page. **Note: Setting the same application and attribute as the source and target for an identity attribute creates circular references.** **Identity attributes with circular references between sources and targets can cause values to be continually changed on every attribute synchronization. This can be problematic when a transformation rule modifies a value without first checking the identity attribute value has already been transformed.**
Advanced Options	The advanced options that are enabled for this attribute. Editable — the attribute can be edited. **Group Factory** — the attribute can be used to create groups that are used for analytical purpose throughout IdentityIQ. **Searchable** — the attributes that are available for filtering in identity searches.

To delete identity attributes, right-click the attribute and select **Delete**.

Note: Deleting an identity attribute also deletes any group factories that reference it. Review the group factory information in the Confirm Deletion of Attribute dialog before clicking Yes.

Edit Identity Attributes Page

Use the Edit Identity Attribute page to create and edit identity attributes including the display name, advanced options and source mapping.

The maximum number of searchable attributes you can create is defined during the application installation and configuration process and controlled from the System Setup pages. The default number is ten (10). See "Create Icons to Represent Specialized Account Attributes" on page 34.

To support the governance of bots, IdentityIQ has three new standard attributes in the identity object that enable you to do things like run a focused certification on just bots.

The attributes are:

- **Type**: an attribute to define the type of identity. The standard values for this attribute are:
  - Employee
  - Contractor
  - External / Partner
  - RPA / Bots
  - Service Account

  However, you can define your own types in addition to these 5, via editing XML in debug
- **Version:** an attribute to indicate what version of software the bot is using. This attribute is intended to be used only for bots.
- **Administrator:** the owner, certifier, of the bot. This is used instead of manager for bots throughout IdentityIQ.

The Edit Identity Attribute page contains the following information:

Table 13— Edit Identity Attributes Page Field Descriptions

Field	Description
Identity Attribute:	
Attribute Name	The name of the attribute as it is used throughout IdentityIQ. For example, this the name used to identify this attribute in rules.
Display Name	The IdentityIQ user assigned name.
Advanced Options:	
Attribute Type	Select from the following attribute types: **String** — creates a text-editable field. **Identity** — creates a drop-down list from which you choose an existing identity.

Table 13— Edit Identity Attributes Page Field Descriptions

Field	Description
Edit Mode	Enable editing of this attribute from the Identity pages. **Read Only** — this attribute cannot be edited from the Identities pages. **Permanent** — changes made on the identities pages are not overwritten by refresh tasks. **Temporary** — changes made on the edit identities pages are overwritten when an aggregation task brings over a new (changed) value for the attribute.
Searchable	Enable this attribute for use in searches and filtering through IdentityIQ.
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.
Group Factory	Enable this attribute for use in creating groups used for analytical purpose throughout IdentityIQ.
Value Change Rule	Specify a rule to run every time a change is detected on this attribute during the aggregation process. For example, a rule can be written to send change notifications, request change approval or launch a certification. Click the "..." icon to launch the Rule Editor to make changes to your rules if needed. See "Rule Editor" on page 15
Value Change Workflow	Specify a business process to run every time a change is detected on this attribute during the aggregation process. For example, a business process can be written to send change notifications, request change approval or launch a certification.
Note: If you set the source and target mapping to the same application/attribute pair, it creates circular references and where the values continuously change with every attribute synchronization.	
Source Mappings: The list of application/attribute pairs from which employee attributes are derived. If the required data is unavailable on the primary source, the collection process continues down the list of configured sources until the information is found.	
Target Mapping (Only available for Identity attribute types): When creating or editing an Identity attribute, use the Target Attribute options to define targets that the basis for attribute synchronization. Click **Add Target** to display the Add a target to the AttributeName attribute dialog, and complete all of the information.	

How to Add or Edit Identity Attributes

1. Click **Add New Attribute** or click an existing attribute to display the Edit Identity Attribute page.
2. Enter or change the attribute name and an intuitive display name.

   Note: You cannot define an extended attribute with the same name as any existing identity attribute.

   Note: Changing an attribute name might cause attributes that were previously aggregated to no longer be recognized.
3. **Optional:** Enable or change the Advance Options.
4. Click **Add Source** to display the Add a source dialog.

5. Specify a source for the new attribute.

   Map directly to an attribute on an application.

   For Application Attributes you have the option to also make this source a target for attribute synchronization. If there are multiple source applications on which a user might have accounts, you would likely want to push the most authoritative value to the rest of the accounts.

   a. Select **Application Attribute**.
   b. Select an application from the **Application** drop-down list.
   c. Select an attribute from the **Attribute** drop-down list.

   Map to an application rule. This rule only applies to the application specified.

   a. Select **Application Rule**.
   b. Select an application from the **Application** drop-down list.
   c. Select a rule from the **Rule** drop-down list.

   Map to a global rule. This rule applies to all applications that contain this attribute.

   a. Select **Global rule (all apps)**.
   b. Select a rule from the **Rule** drop-down list.

6. Click **Add** to add the new source.
7. Use the arrows to the right of the sources list to rearrange the search order for the attribute sources. When aggregation tasks are run they search the source at the top of the list, or the primary source, first and then work down the list.
8. For Identity attribute types only, add targets for attribute synchronization:
   a. Select **Add Target** to display the Add a target to the attribute dialog.
   b. Select the application to receive the value.
   c. Select the attribute to receive the value.
   d. **Optional:** Select a transformation rule to transform the value before it is set on the destination.
   e. **Optional:** Select Provision All Accounts to provision all of the identities accounts on the targeted application. If you disable this option you are asked to select the accounts to provision manually.
9. Click **Save** to create the new attribute and return to the Identity Attribute page.

Account Mappings

Note: Extended attribute names must be unique. Extended attributes cannot share a name with any other attribute in any other application schema.

Use the Account Mapping page to setup and map specialized accounts. Specialized accounts can be any accounts that justify special handling throughout your enterprise. For example privileged accounts such as Root, Administrator, or Super User, and service accounts that access a specific service or function on an application. Any attribute extended on this page is available for searching on the Identity Search page.

You can assign icons to extended attributes to highlight these accounts in certifications and the detailed identity pages. See "Create Icons to Represent Specialized Account Attributes" on page 34.

Specialized account attributes can be modeled to handle any concept using simple one-to-one mapping and rules. This section describes two of the most common scenarios.

Use the Account Attributes page to view the extended account attribute information for your configuration. Use this page to set up specialized account attributes such as Privileged and Service, and any other extended attributes for use in certifications and searches.

The Account Attributes page contains the following information:

Table 14— System Setup- IdentityIQ - Account Mapping - Account Attributes Descriptions

Column	Description
Attribute	The display name of an account attribute derived from the attribute and its associated application in the Primary Source Mapping column.
Primary Source Mapping	The first of the list of attribute/application pairs or rules from which account attributes are derived. If the required data is unavailable on this primary source, the collection process continues down the list of configured sources until the information is found. Set up the list of sources on the Edit Account Attribute page.

To work with the attributes and sources, see "How to Add or Edit Account Attributes" on page 32.

To delete account attributes, right-click the attribute and select **Delete**.

To edit account attributes, right-click the attribute and select **Edit**.

Edit Account Attributes Page

Use the Edit Account Attribute page to create and edit account attributes including the display name, attribute type and source mapping. You can also use this page to create specialized account attributes. See "Create Icons to Represent Specialized Account Attributes" on page 34.

The maximum number of searchable attributes that can be created is defined during the installation and configuration process. By default you can set five searchable account attributes.See "System Setup" on page 3.

The Edit Account Attribute page contains the following information:

Table 15— Edit Account Attributes Page Field Descriptions

Field	Description
Account Attribute:	
Attribute Name	The name of the attribute as it appears in the application. **Note: Changing an attribute name might cause attributes that were previously aggregated to no longer be recognized.**
Display Name	The IdentityIQ user assigned name for use throughout IdentityIQ.
Advanced Options:	
Edit Mode	Enable editing of this attribute. **Read Only** — this attribute cannot be edited. **Permanent** — changes made to this attribute manually are not overwritten by refresh tasks. **Temporary** — changes made to this attribute manually are overwritten by the first refresh task that detects a value different than the original value. For example, if the original value is A and it is manually changed to B, the value is not overwritten by a refresh task until the newly aggregated value is not A. When an aggregation detects a value that is not A, it refreshes the manually-changed value and the value is updated with each subsequent refresh.
Attribute Type	The attribute type to be linked, for example string, boolean or date.

Table 15— Edit Account Attributes Page Field Descriptions

Field	Description
Searchable	Account attributes are existing link values and are always searchable. This field is displayed as selected and read only so that identity and account attribute configuration pages are consistent in appearance.
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.
Source Mappings: The list of attribute/application pairs or rules from which account attributes are derived. If the required data is unavailable on this primary source, the collection process continues down the list of configured sources until the information is found. This feature is unlikely to be used for Account Attribute mapping.	

How to Add or Edit Account Attributes

1. Click **Add New Attribute** or click an existing attribute to display the Edit Account Attribute page.
2. Enter or change the attribute name and an intuitive display name.

   Note: You cannot define an extended attribute with the same name as any application attribute that is provided by a connector.
3. Edit the Advance Options as required.
4. Click **Add Source** to display the Add a source dialog and specify a source for the new attribute.

   Map directly to an attribute on an application.
   a. Select **Application Attribute**.
   b. Select an application from the **Application** drop-down list.
   c. Select an attribute from the **Attribute** drop-down list.

   Map to an application rule. This rule only applies to the application specified.
   a. Select **Application Rule**.
   b. Select an application from the **Application** drop-down list.
   c. Select a rule from the **Rule** drop-down list.

   Map to a global rule. This rule applies to all applications that contain this attribute.
   a. Select **Global rule (all applications)**.
   b. Select a rule from the **Rule** drop-down list.
5. Click **Add** to add the new source.
6. Use the arrows to the right of the sources list to rearrange the search order for the attribute sources. When aggregation tasks are run they search the source at the top of the list, or the primary source, first and then work down the list.
7. Click **Save** to create the new attribute and return to the Account Attribute page.

Account Attributes

Create a Service Account Using Simple Mapping

In this example, if IdentityIQ finds an attribute named Service that has a value of true on the application DB Application it is marked as a service account. For this case the database connector has already provided an attribute value to reflect the service state, so a simple mapping is all that is required.

Note: After configuring these attributes you must re-aggregate or refresh the identity cubes to set the values.

To configure the mapping:

1. Access the Account Attributes page.
   Select the System Setup tab and select **Account Mappings** from the table.
2. Click **Add New Attribute** to display the Edit Account Attribute page.
3. Specify the following values:
   - **Attribute Name** — service
   - **Display Name** — Service Account
   - **Edit Mode** — Read Only
   - **Attribute Type** — boolean
   - **Searchable** — Read Only
   - **Multi-Valued** — this is not a multi-valued attribute so do not select this field.
4. Click **Add Source Mapping** to display the Add a source to the attribute dialog.
5. Map the attribute:
   a. Select **Application Attribute**.
   b. Select **DB Application** from the Application drop-down list.
   c. Select **Service** from the Attribute drop-down list.
6. Click **Add**.

Create a Privileged Account Using a Rule

In this example, if IdentityIQ finds an account that is a member of the group **Domain Admins** on any AD application, that account should be marked as a privileged account.

1. Write the rule to define the logic.
   This rule checks each account on every AD application and looks for the Domain Admins group. If the Domain Admins group is found, the rule returns true, and the account is considered privileged.
   Example rule:

```
<Rule language="beanshell" name="Example privileged promotion rule"
type="LinkAttribute">
<Source>
<![CDATA[
Boolean privileged = null;
If ( link.getApplication().getName().contains(“AD”) ) {
privileged = new Boolean(false);
List groups = (List)link.getAttribute("memberOf");
if ( groups != null ) {
```

```
for ( String group : groups ) {
if ( ( group != null ) &&
( group.startsWith("cn=Domain Admins") ) ) {
privileged = new Boolean(true);
}
}
}
)
return privileged;
]]>
</Source>
</Rule>
```

2. Access the Account Attributes page.
   Go to the Global Settings and select **Account Mappings**.
3. Click **Add New Attribute** to display the Edit Account Attribute page.
4. Specify the following values:
   - **Attribute Name** — service
   - **Display Name** — Service Account
   - **Edit Mode** — Read Only
   - **Attribute Type** — boolean
   - **Searchable** — Read Only
   - **Multi-Valued** — this is not a multi-valued attribute so do not select this field.
5. Click **Add Source** to display the Add a source to the attribute dialog.
6. Map the attribute:
   Select **Global Rule (all applications)**.
   Select **Example privileged promotion rule** from the Application drop-down list.
7. Click **Save**.

Create Icons to Represent Specialized Account Attributes

Assign icons to extended attributes to highlight these accounts in certifications and the detailed identity pages. To assign icons you must modify the UIConfig file and add AccountIconConfig entries for any value that should be recognized.

The following example references the attributes defined in this section.

```
<ImportAction name='merge'>
<UIConfig name='UIConfig'>
<Attributes>
<Map>
<entry key='accountIconConfig'>
<value>
<List>
<!—This indicates that when we are displaying accounts and we see
 the value “true” for the extended account attribute named
privileged we should display the icon listed in the “source”
attribute. The title will be used in hover-over help.
->
<AccountIconConfig attribute="privileged"
value="true"
```

```
source="/images/icons/privilege_16.png"
title="This is a privileged account"/>
<!—This indicates that when we are displaying accounts and we see
the value “true” for the extended account attribute named
service we should display the icon listed in the “source”
attribute. The title will be used in hover-over help.
->
<AccountIconConfig attribute="service"
value="true"
source="/images/icons/service.png"
title="This is a service account"/>
</List>
</value>
</entry>
</Map>
</Attributes>
</UIConfig>
</ImportAction>
```

Use the IdentityIQ console to import the modifications.

Application Attributes

Use the Edit Application Configuration page to define extended application attributes not provided by the application connectors during aggregation. These extended attributes are displayed on the Attributes tab of the Application Configuration page below the connection attributes provided by the connector. Because the additional attributes are stored with those provided by the connector, if you define an extended attribute with a name that matches any connector attribute, the values of the extended attribute overwrite the values of the connector attribute.

You can use these extended attributes inside rules and custom reports and queries.

The Edit Application Configuration page contains the following information:

Table 16— System Setup- IdentityIQ -Configure Account Mapping Descriptions

Column	Description
Name	The display name of the application attribute assigned when it was added.
Category	The category defined when the attribute was created. If no category was defined this column is blank.
Description	A short description of the extended application attribute.

Click **New Attribute** to add additional attributes to the applications.

To edit or delete an existing attribute from the list, right-click the attribute and select the corresponding option from the menu. If you are deleting an attribute you must confirm the deletion in the pop-up dialog.

Edit Application Attributes

Use the Edit Extended Attribute page to create and edit additional application attributes including the display name, attribute type and description.

The fields displayed on the Edit Extended Attribute page are dependent on the attribute type selected.

The Edit Extended Attribute page contains the following information:

Table 17— Edit Extended Attribute Page Field Descriptions

Field	Description
Attribute Name	The name of the attribute as it appears in the application. Changing an attribute name might cause attributes that were previously aggregated to no longer be recognized.
Display Name	The IdentityIQ user assigned name for use throughout IdentityIQ.
Type	The attribute type to be linked, for example string, boolean, date, rule, or identity.
Description	A brief description of the application attribute.
Category Name	An optional category used to separate the attributes into categories on the Application Configuration page. Enter a category name or select an existing one from the drop-down list.
Searchable	Enable this application attribute for use in searches throughout the product.
Editable	Enable editing of this attribute from other pages in the product.
Required	For String type attributes only. Required attributes must have a value before you can save an application.
Allowed Values	For String type attributes only. Enter the values that are allowed for this attribute. The values entered in this list are used to populate the drop-down value list on the Application Configuration page.
Default Value	Enter a default value for the attribute or select a value from the drop-down list, depending on the attribute type you are working with.

How to Add or Edit Extended Attributes

1. Click **New Attribute** or click an existing attribute to display the Edit Extended Attribute page.
2. Enter or change the attribute name and an intuitive display name.

   Note: You cannot define an extended attribute with the same name as any application attribute that is provided by a connector.

   Note: If you define an extended attribute with the same name as an application attribute, the value of the extended attribute overwrites the value of the connector attribute.
3. Select the attribute type from the drop-down list, String, Integer, Boolean, Date, Rule, or Identity.
4. **Optional:** Enter a description of the additional attribute.
5. **Optional:** Select a category for the attribute.
6. **Optional:** Activate the Searchable option to enable this attribute for searching throughout the product.
7. **Optional:** Activate the Editable option to enable this attribute for editing from other pages within the product.
8. **Optional:** Mark the attribute as required. For string type attributes only.
9. **Optional:** Enter allowed values for the attribute. For string type attributes only.
10. **Optional:** Specify a default value.
11. Click **Save** to save your changes and return to the Edit Application Configuration page.

Entitlement Catalog Attributes

Use the Edit Entitlement Catalog Configuration page to define custom extended entitlement attributes. The extended attributes are displayed with the rest of the entitlement information throughout the product. An example of a extended entitlement attribute might be Time Zone.

The Edit Entitlement Catalog Configuration page contains the following information:

Table 18—System Setup- IdentityIQ - Entitlement Catalog Attributes - Column Descriptions

Column	Description
Name	The display name of the extended entitlement attribute assigned when it was added.
Category Name	The category defined when the attribute was created. If no category was defined this column is blank.
Description	A short description of the extended entitlement attribute.

Click **New Attribute** to add additional extended entitlement attributes.

To edit or delete an existing attribute or type from the list, right-click the item and select the corresponding option from the menu. If you are deleting, you must confirm the deletion in the pop-up dialog.

Edit Extended Entitlement Attributes

Use the Edit Extended Attribute page to create and edit additional role attributes including the display name, attribute type and description.

The Edit Extended Attribute page contains the following information:

Table 19— Edit Extended Entitlement Attribute Page Field Descriptions

Field	Description
Attribute Name	The name of the attribute as it appears in the application. Changing an attribute name might cause attributes that were previously aggregated to no longer be recognized.
Display Name	The name for use throughout the product.
Type	The attribute type to be linked, for example string, boolean, date, rule, or identity.
Description	A brief description of the entitlement attribute.
Category Name	An optional category used to separate the attributes into categories on the Application Configuration page. Enter a category name or select an existing one from the drop-down list.
Searchable	Enable this entitlement attribute for use in queries.
Editable	Enable editing of this attribute from other pages in the product.
Required	For String type attributes only. Required attributes must have a value before you can save an entitlement.
Allowed Values	For String type attributes only. Enter the values that are allowed for this attribute. The values entered in this list are used to populate the drop-down value list on the Roles page.

Table 19— Edit Extended Entitlement Attribute Page Field Descriptions

Field	Description
Default Value	Enter a default value for the attribute or select a value from the drop-down list, depending on the attribute type you are working with.

How to Add or Edit Extended Entitlement Attributes

1. Click **New Attribute** or click an existing attribute to display the Edit Extended Attribute page.
2. Enter or change the attribute name and an intuitive display name.

   Note: You cannot define an extended attribute with the same name as any application attribute that is provided by a connector.
3. Select the attribute type from the drop-down list, String, Integer, Boolean, Date, Rule, or Identity.
4. **Optional:** Enter a description of the additional attribute.
5. **Optional:** Select a category for the attribute.
6. **Optional:** Activate the Searchable option to enable this attribute for searching throughout the product.
7. **Optional:** Activate the Editable option to enable this attribute for editing from other pages within the product.
8. **Optional:** Mark the attribute as required. For string type attributes only.
9. **Optional:** Enter allowed values for the attribute. For string type attributes only.
10. **Optional:** Specify a default value.
11. Click **Save** to save your changes and return to the Edit Entitlement Catalog Configuration page.

Quicklink Populations

Quicklinks are tasked-based links to frequently-used areas of IdentityIQ. Quicklinks are displayed as cards on the IdentityIQ Home page and as links in the Quicklink Menu, which is available throughout the product.

Use the Quicklinks Populations page to associate quicklinks, that are created and imported into IdentityIQ by your administrators, with quicklink populations, sometimes referred to as dynamic scopes.

Quicklink populations grant access to specific areas of IdentityIQ to predetermined populations of users. These populations can be defined based on capabilities, identity attributes, work groups, or by selecting individual identities.

One predefined population, Everyone, is in the list by default. If you have purchased IdentityIQ Lifecycle Manager you also see Help Desk, Manager, and Self Service in the Populations list.

Select a population from the list or click **New** to open the Configuration and Quicklinks tabs.

The Configuration tab contains the following:

Table 20—Quicklink Populations page Configuration tab field descriptions

Field	Description
Details	
Name	Name of the population.
Description	Description of the population.

Table 20—Quicklink Populations page Configuration tab field descriptions

Field	Description
Membership	
Membership Rule	Select a membership rule to define the population. **None** — only the identities specified in the **Included Identities** list are in the population. **All** — include all identities in the population. **Match List** — only identities whose criteria match that specified in the list. Add identity attributes, application attributes and application permissions. Customize further by creating attribute groups to which this assignment rule applies. 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** — a custom database query. **Script** — a custom script. **Rule** — select an existing rule from the drop-down list. Click **Edit Rule** to launch the Rule Editor. See “Rule Editor” on page 15. **Population** — select an existing population.
Included Identities	Manually select identities to include in the population.
Excluded Identities	Manually select identities that should not be included in the population. For example, Administrator.
Who can members request for? Identities for whom the members of this population can make access requests.	
Everyone	Can create access request for anyone.
Specific Users	Can only create access requests for identities based on the selected criteria. Use the drop down list to specify if they must match all of the criteria or just any of the criteria.
Share attributes with the requester	Can make requests for identities that share the attributes specified.
Report to the requester	Enable managers to make requests for their subordinates. Specify if this applies to direct reports or all subordinates. If all subordinates, specify a **Maximum Hierarchical Depth**.
Match custom criteria	The filter is the context of the identity object and is parsed as a Velocity template with a parameter called requester.spa For example for an identity whose manager’s name is the same as the manager’s name for the requester: `manger.name == “$requester.manager.name”`
Ignore scoping	Disregard IdentityIQ scopes when determining for whom request can be made.
What can members request? **Note: Click Edit Rule to launch the Rule Editor for any of the following. See “Rule Editor” on page 15.**	
Roles	Select a rule that defines the set of roles that this population can request.
Applications	Select a rule that defines the set of applications from which this population can request entitlements.
Entitlements	Select a rule that defines the set of entitlements that this population can request.

Table 20—Quicklink Populations page Configuration tab field descriptions

Field	Description
What can members remove? **Note: Click Edit Rule to launch the Rule Editor for any of the following. See "Rule Editor" on page 15.**	
Roles	Select a rule that defines the set of roles that this population can remove.
Applications	Select a rule that defines the set of applications from which this population can remove entitlements.
Entitlements	Select a rule that defines the set of entitlements that this population can remove.
Sync with Request	The selections from **What members can request** is copied to **What members can remove**.

Note: The Quicklinks tab contains all of the quicklinks that are available in your environment. You cannot add quicklinks within IdentityIQ. The New button opens the Configuration tab to create a new population.

Table 21—Quicklink Populations page Quicklinks tab description

Column	Description
Enabled	Specify which quicklinks to associate with this population.
Name	Name of the quicklink as it appears on cards on the IdentityIQ Home page and as links in the Quicklink Menu.
Description	Description of the quicklink.
Category	The category in which this quicklink displays in the Quicklink Menu.
Options	When available, use **Configure** to specify quicklink settings.

Forms

Form Editors are used for configuring forms for Workflows, Role Provisioning Policies, and Application Provisioning Policies in IdentityIQ. Forms are referred in two ways, Centralized Forms and Reference Forms.

Centralized Forms

Centralized Form location is a single location within the system, where an Administrator would be able to view all the forms. All of the forms can be created, edited, managed, and maintained as one object.

The Forms page displays all the standalone forms. The form grid displays Workflow, Role and Application types of forms.

Create and Edit Forms

To create a new form, click the **Create Form** button. On the **Create New Form** window, select one of the following type of the form to be created:

- Application Provisioning Policy Form
- Role Provisioning Policy Form
- Workflow Form

On saving the newly created forms, the form grid would be updated.

To edit an existing form, click the **Edit** button provided next to each form in the grid to display the Form Editor page.

Search Forms

Forms are searched by their names. To search the required form, enter any name word of the Form Name. The search results is displayed by refreshing the list to display the searched form.

Reference Forms

Reference Forms provide a means to create a single form that can be reused and referenced as standalone forms for Application Provisioning Policy Form, Role Provisioning Policy Form, and Workflow Form.

Create Forms

To create a new policy form, click **Add Policy** under the Provisioning Policies tab to display the Forms window. Click **Create Policy Form** to display the Provisioning Policy Editor page.

On saving the newly created policy, the provisioning policies are updated.

Referencing Forms

Note: The Existing Forms lists only contain the reference forms of the type associated with the policy type you are viewing, application, role, or workflow. If you are not seeing forms in the list, ensure that the forms you are looking for have the correct type set on the Global Settings -> Forms page.

Form referencing is done by using the form name. To reference a form, click **Add Policy** under the Account Provisioning Policies tab to display the Forms window. Click **Reference Policy Form** to display the list of application provisioning policy forms from the central location on the Existing Forms page.

Edit Reference Forms

Click the policy name to display an option to **Create a Form** or **Change Reference Policy Form** button to edit an existing reference form.

- Click **Create Policy Form** to display the Provisioning Policy Editor page. On saving the newly created policy, the reference provisioning policies are removed and policy is updated with newly created form.
- Click **Change Reference Policy Form** to display a list of application provisioning forms.

Role Configuration

Use the Edit Role Configuration page to define custom extended role attributes and role types. The extended attributes are displayed with the rest of the role information throughout the product. An example of a extended role attribute might be role status. Role type is used to configure roles to perform different functions within your business model. For example, type might be used to control inheritance or automatic assignment of roles.

The Edit Role Configuration page contains the following information:

Table 22—System Setup- IdentityIQ -Configure Role Attribute Column Descriptions

Column	Description
Role Attributes:	
Name	The display name of the role attribute assigned when it was added.
Category	The category defined when the attribute was created. If no category was defined this column is blank.
Description	A short description of the role attribute.
Role Types:	
Name	The display name of the role type.
Description	A short description of the role type.

Click **New Attribute** to add additional role attributes. See "Edit Extended Role Attributes" on page 42.

Click **New Type** to add or edit a role type. See "Edit Role Types" on page 43.

To edit or delete an existing attribute or type from the list, right-click the item and select the corresponding option from the menu. If you are deleting, you must confirm the deletion in the pop-up dialog.

Edit Extended Role Attributes

Use the Edit Extended Attribute page to create and edit additional role attributes including the display name, attribute type and description.

The Edit Extended Attribute page contains the following information:

Table 23— Edit Extended Role Attribute Page Field Descriptions

Field	Description
Attribute Name	The name of the attribute as it appears in the application. **Note: Changing an attribute name might cause attributes that were previously aggregated to no longer be recognized.**
Display Name	The name for use throughout the product.
Type	The attribute type to be linked, for example string, boolean, date, rule, or identity.
Description	A brief description of the role attribute.
Category Name	An optional category used to separate the attributes into categories on the Application Configuration page. Enter a category name or select an existing one from the drop-down list.
Searchable	Enable this role attribute for use in queries.
Editable	Enable editing of this attribute from other pages in the product.
Required	For String type attributes only. Required attributes must have a value before you can save a role.

Table 23— Edit Extended Role Attribute Page Field Descriptions

Field	Description
Allowed Values	For String type attributes only. Enter the values that are allowed for this attribute. The values entered in this list are used to populate the drop-down value list on the Roles page.
Default Value	Enter a default value for the attribute or select a value from the drop-down list, depending on the attribute type you are working with.

How to Add or Edit Extended Attributes

1. Click **New Attribute** or click an existing attribute to display the Edit Extended Attribute page.
2. Enter or change the attribute name and an intuitive display name.

   Note: You cannot define an extended attribute with the same name as any application attribute that is provided by a connector.
3. Select the attribute type from the drop-down list, String, Integer, Boolean, Date, Rule, or Identity.
4. **Optional:** Enter a description of the additional attribute.
5. **Optional:** Select a category for the attribute.
6. **Optional:** Activate the Searchable option to enable this attribute for searching throughout the product.
7. **Optional:** Activate the Editable option to enable this attribute for editing from other pages within the product.
8. **Optional:** Mark the attribute as required. For string type attributes only.
9. **Optional:** Enter allowed values for the attribute. For string type attributes only.
10. **Optional:** Specify a default value.
11. Click **Save** to save your changes and return to the Edit Role Configuration page.

Edit Role Types

Use the Edit Role Type Definition page to create and edit types to use with roles. Role type is used to configure roles to perform different functions within your business model. For example, type might be used to control inheritance or automatic assignment of roles.

Role modeling also uses the concept of permission to enable you to grant users permission to specific roles without assigning them the role or incorporating it in their role hierarchy. For example, while a non-IT user with a business-type role might need access to the entitlements contained within an IT-type role, they probably do not need to have that role assigned to them or included as part of their hierarchal role structure.

The Edit Role Type Definition page contains the following information:

Table 24— Edit Role Type Definition Page Field Descriptions

Field	Description
Type Name	The name of the role type.
Display Name	The display name of the role type used throughout the product.
Description	A brief description of the role type.

Table 24— Edit Role Type Definition Page Field Descriptions

Field	Description
Icon Path	The path to the iconic representation of this role type. See "Edit Role Types" on page 43
Disallow inheritance of other roles	Do not allow roles of this type to inherit other defined roles.
Disallow other roles from inheriting this role	Do no allow roles of this type to be inherited.
No automatic detection with profiles	Do not automatically detect and assign this role to identities during aggregation and correlation.
No automatic detection with profiles unless assigned	Do not automatically detect and assign a role during aggregation and correlation unless it is required or permitted by an identity's assigned roles.
No entitlement profiles	Do not enable the direct assignment of profiles to this role type. For example, a roles used to create hierarchy in your business model might only gain access to entitlement profiles through permitted IT roles.
No automatic assignment with rule	Do not allow a rule to automatically assign roles of this type to identities.
No assignment rule	Do not display the Assignment Rule panel in the Role Modeler for rules of this type.
No manual assignment	Do not allow roles of this type to be assigned manually from the Identities User Rights tab.
No permitted roles list	Do not display the Permitted Roles panel in the Role Modeler for rules of this type.
Disallow this role from being on a permitted roles list	Do not display roles of this type on the select list of the Permitted Roles panel of any other role.
No required roles list	Do not display the Required Roles panel in the Role Modeler for rules of this type.
Disallow this role from being on a required roles list	Do not display roles of this type on the select list of the Required Roles panel of any other role.
Disallow Granting of IdentityIQ User Rights	Do not allow the granting of IdentityIQ capabilities or scopes based on role assignment. If this option is selected, the Granted IdentityIQ User Rights table is not displayed on the Role Editor page.

How to Add or Edit Role Types

1. Click **New Type** or click an existing type to display the Edit Role Type Definition page.

2. Enter or change the name and display name.
3. Enter an icon path to link to the iconic image associated with roles of this type in the Role Modeler.
   To assign an icon to a role type, do the following:
   a. Add two icon images to iiq_home/images/icons folder of your IdentityIQ installation, one for the role and one for the role as it is undergoing analysis or approval. For example,
   ```
   .itIcon {
   background-image: url("../images/icons/modeler_application_16.png")
    !important;
   background-repeat: no-repeat;

   .itIconPendingbusiness process {
   background-image: url("../images/icons/
   modeler_application_approval_16.png") !important;
   background-repeat: no-repeat;
   ```
   b. Reference the images from the iiq-custom.css file in the iiq_home/css directory.
4. **Optional:** Select configuration options for the role type.
5. Click **Save** to save your changes and return to the Edit Role Configuration page.

Scopes

Scope is used to determine the objects to which a user has access. If scoping is active, identities can only see objects that they created or that are within the scopes they control. IdentityIQ capabilities control the components within the product to which a user has access. Scope controls access to the individual objects within those components. For example, a user might be able to access the Identity Search page, however, the Application and Role drop-down lists only display application and roles that are contained within a scope they control.

Scope is referred to in two ways, Controlled Scope and Assigned Scope. Assigned scope is the scope assigned to an identity or object manually, automatically, or through aggregation and correlation. Controlled scopes refer to the scopes to which an identity has access. You can only see objects that are within your controlled scopes, that you created, or possibly that have no scope assigned. Controlled scope is hierarchical. If you control a parent scope, you control any child scopes contained within.

Use the Configure Scoping page to create new scopes, edit existing scopes, and configure scoping for your enterprise.

Note: If you manually create scopes they should be associated with existing identity attributes or be defined in a scope correlation rule.

Create and Edit Scopes

To create a new scope, right-click **Scopes** and select **New** to display the Create Scope page. Enter the scope name and click **Create** to return to the Scope page. Use the Scope Correlation Rule to correlate identities with the correct scopes.

To edit an existing scope, right-click the scope and select **Edit** to display the Edit Scope page. You can only edit the display name.

Drag and drop existing scopes to create a scope hierarchy.

Delete Scopes

To delete a scope, right-click the scope and select **Delete** to display the Delete Scope page. The Delete Scope page contains the following:

Table 25—Scope Configuration - Delete Scope Page Field Descriptions

Field	Description
Assigned Scope Replacement	Reassign objects to a different scope upon deletion.
Authorized Scope Replacement	Assign an authorized scope to replace the one to be deleted.
Delete Child Scopes	Delete all child scopes in the scope hierarchy.

Configure Scoping

Use the Configure Scoping page to configure scope assignment and correlation. The Configure Scoping page contains the following:

Note: You must run an identity refresh task with the refresh scope option enabled before scope configuration changes are visible.

Table 26—Scope Configuration - Configure Scoping Page Field Descriptions

Field	Description
Enable Scoping	When checked, scoping mechanisms are enabled. Scopes do not take effect until this is enabled, even if the scopes are already defined and assigned. Note: De-selecting this option is useful in troubleshooting performance issues.
Scope Identity Attribute	Select an identity attribute from the drop-down list to use for scoping. A scope is created for each value of the selected attribute aggregated during the identity refresh task. This attribute is used to correlate identities to assigned scope.
Scope Correlation Rule	Select a rule to use to correlated scopes and identities during aggregation and refresh task. If a scope is not found that correlates to the value returned by an attribute, one is created. Scope correlation rules enable more flexibility in scope assignment than specifying a single identity attribute. Note: Click the "..." icon to launch the Rule Editor to make changes to your rules if needed. ... See "Rule Editor" on page 15

Table 26—Scope Configuration - Configure Scoping Page Field Descriptions

Field	Description
Scope Selection Rule	Select a selection rule to use if the identity attribute or scope correlation rule return more then one value for the assigned scope of an identity. For example, if department is specified as the scope identity attribute and the identity aggregation task returns more then on value for department for an identity, this rule determines which value to use as the assigned scope. Note: Click the "..." icon to launch the Rule Editor to make changes to your rules if needed. See "Rule Editor" on page 15
Unsloped Objects Globally Accessible	When selected, all objects that do not have an assigned scope are available to all users. When cleared, all objects that do not have an assigned scope are only available to system administrators.
Identity Controls Assigned Scope	When selected, identities automatically control the scope to which they are assigned.

Time Periods

Use the Configure Time Periods page to specify the time periods used for activity searching. Setting time periods for your enterprise enables you to track who is accessing your sensitive applications and when they are accessing it. Access at unusual times can indicate a security issue that requires investigation. Time periods include things such as office hours, holidays, and weekends. Because each time period is set individually, you can customize the setting to meet the needs of your enterprise.

The following are the available time periods:

- Date ranges — a range of specific dates that define things such as fiscal quarters.
- Time ranges — a range of hours, or times, that define office hours and non-office hours.
- Date lists — a list of dates that define enterprise holidays.
- Day lists — a list of days that define week days and weekends.

To edit a time period, click a time period in the Time Periods column to access the Configure Time Period page and make the required changes. See “Configure Time Period” on page 47.

Configure Time Period

The configuration options on the Configure Time Period page are based on the type of time period to be edited.

Date Ranges:

For date ranges, specify the Begins on and Ends on dates for the time period to be defined. For these date ranges you can enter dates manually of click the... icon and select a date from the calendar.

Time Ranges:

For time ranges, specify the starting at and ending at times. Time ranges are used to define working and non-working hours.

Date Lists:

Use the date list to specify a list of holidays, or regularly schedule dates on which your enterprise business would not normally conduct business. This list does not include weekends. Weekends are defined separately from a day list. You can enter dates manually of click the "..." icon and select a date from the calendar. Use the Add and Delete buttons to add or remove dates from the list.

Day Lists:

Use the day lists to specify week days from weekend days. Select the correct days using the selection boxes on the right of the table and deselect, or DE-activate, the days that should not be included.

Audit Configuration

Use the Audit Configuration page to specify the actions that are collected for audit logs. Since collecting event information and storing it in the audit logs affects performance, a system administrator must specify the actions that are audited. Before any data is collected by the audit logs for use in an audit search, IdentityIQ must be configured for auditing.

The Audit Configuration page contains the following types of actions:

- General Actions — typical action performed while using IdentityIQ. For example, running a task and signing off on a certification are general actions.
- Link Attribute Changes — changes made to any assigned link attributes.
- Identity Attribute Changes — changes to assigned roles, capabilities, authorized scopes, and controlled scopes, and changes to the password. This list might also include extended identity attributes.
- Class Actions — action taken on the underlying classes used to configure the way in which IdentityIQ operates. For example, editing a role, creating a policy, specifying the default email template, and adding attachments are class actions.
- SCIM Resource Actions — action taken on any SCIM resource, for example, create, read, update, and delete.

Electronic Signatures

Electronic signatures provide proof of a decision. Use the Electronic Signatures page to set the meaning or text that is displayed to the end user when they perform an electronic signature.

Note: Electronic Signatures have legal implications. They need to explicitly state each action and consequence to ensure end users understand what they are signing.

Electronic Signature Meanings

The Electronic Signature Meanings page displays a table of configured meanings. Click **New Meaning** to open the New Electronic Signature Meaning window. Input the Name, Display Name, and Meaning then click **Save**.

API Authentication

IdentityIQ supports the use of OAuth 2.0 (client credentials) as a token-based protocol for API authentication. Use this feature to create and manage OAuth clients that you use with the IdentityIQ API.

Note: You set up a proxy user that connects on behalf of the user to avoid exposing sensitive user data. In order for the proxy user to have correct rights to make API calls, you must assign capabilities to that proxy user.

OAuth Client Management Page

The OAuth Client Management page has the following tabs and options:

- OAuth Client Management tab — displays a list of the current OAuth clients.
  - Create Button — creates an OAuth client that has a proxy user with an associated secret.
  - Secret Details icon — displays the secret for the an OAuth client.
  - Actions icons — Edit, Delete, Regenerate Secret
- General Settings tab
  - Access Token Expiration In Seconds

How To Create An OAuth Client

1. From the top menu, navigate to the **Gear icon > Global Settings > API Authentication**.
2. On the OAuth Client Management tab, click **Create**.
3. In the OAuth Client dialog enter a unique name for **Client Name** and then enter a user name or select a user from the drop-down list for the **Proxy User**.
4. Click **Save** to save your new OAuth client.

After your create an OAuth client, you can use it with the associated secret to log in and access the token for that proxy user.

IdentityAI Configuration

Note: This link is only present if you have purchased the IdentityAI product. Refer to the SailPoint *IdentityAI Implementation Guide*.

Use the IdentityAI Configuration page to connect IdentityIQ to the IdentityAI product.

Table 27— IdentityAI Configuration Page Field Descriptions

Field	Description
Connection Information for IdentityAI:	
IdentityAI Hostname	The host name of the IdentityAI recommendation API
Client ID	OAuth client ID for the IdentityAI recommendation API
Client Secret	OAuth client secret for the IdentityAI recommendation API
Advanced:	
Read Timeout	The number of seconds IdentityIQ attempts to read recommendations from IdentityIQA before reporting a failure
Connect Timeout	The number of seconds IdentityIQ attempts to connect to IdentityIQA before reporting a failure

Use **Test Connection** to ensure the connection information is accurate and operating.

File Access Manager Configuration

IdentityIQ can integrate with File Access Manager to bring key data governance features to the IdentityIQ business user. A Data Governance menu in IdentityIQ provides direct access to the File Access Manager website, and dashboard widgets provide the context needed to make informed access decisions.

You can also use the File Access Manager Configuration settings to configure correlation logic for mapping File Access Manager objects to IdentityIQ identities.

To install and configure the File Access Manager integration:

1. Import the `init-fam.xml` file into IdentityIQ, using the iiq console or the **gear menu > Global Settings > Import From File** page.
2. Click the **gear menu > Global Settings > File Access Manager Configuration.**

Table 28— File Access Manager Configuration Page Field Descriptions

Field	Description
Connection Information for File Access Manager:	
File Access Manager Hostname	The host name of the File Access Manager server. This is the host where the File Access Manager website is installed.
Basic / OAuth	Choose your method of authenticating with the File Access Manager website. **Basic** uses a username and password. **OAuth** uses a client ID and client secret. Basic authentication can be used for identities that are configured in the File Access Manager Administrative Client as having the API User privilege. OAuth credentials can be retrieved from the File Manager website, through the S**ettings > General > API Authorization** menu.
Username	For Basic authentication, enter the username for an identity that has the API User privilege in File Access Manager.
Password	For Basic authentication, enter the password for an identity that has the API User privilege in File Access Manager.
Client ID	For OAuth authentication, enter the OAuth client ID for File Access Manager
Client Secret	For OAuth authentication, enter the OAuth client secret for File Access Manager
Correlation Information for File Access Manager:	
SCIM Correlation Rule	If the correlation logic in your configured applications does not meet your needs for correlating File Access Manager groups and accounts to IdentityIQ groups, you can use a custom rule to manage correlation. The rule must have a rule type of `Correlation` in order to appear in this drop-down.
SCIM Correlation Applications	Select the application(s) to use for correlating File Access Manager objects to IdentityIQ identities. Selecting an application here means that the correlation logic defined for the application will determine how File Access Manager objects are correlated to identities.

Use **Test Connection** to verify that the connection information is accurate and functional.

Data Governance Menu

Once the File Access Manager integration is configured, a **Data Governance** menu is available in IdentityIQ. The Data Governance menu provides direct access to features in the File Access Manager website.

The Data Governance menu is available only to users who have the IdentityIQ **FAM Administrator** capability, or any capability that includes the **ViewFAMNavigationMenu** SPright.

For more information about Data Governance in File Access Manager, refer to the *IdentityIQ File Access Manager Administrator Guide*.

Dashboard Widgets

Once the File Access Manager integration is configured, widgets that show data about **Sensitive Data Exposure** and **Sensitive Resources Missing Owners** are available on the IdentityIQ home page.

The widgets display **read-only** information about sensitive data that is monitored in File Access Manager. Each widget shows counts for resources, and an overall compliance score. The compliance score is color-coded to indicate risk (0-5 is considered high risk, 5.1-7.5 medium risk, and 7.6-10 low risk).

The widgets do not provide direct access to the File Access Manager website; in other words, users cannot click the widgets for more detailed information, or to access the File Access Manager website.

These widgets are available to users who have the IdentityIQ **FAM Administrator** capability, or any capability that includes the **ViewFAMAdminWidgets** SPright.

Users can click **Edit** on the home page to add, remove, or move these widgets.

For more information about Sensitive Data Exposure and Data Ownership in File Access Manager, refer to the *IdentityIQ File Access Manager Administrator Guide*.

Import From File

Use the Import From File page to import files into IdentityIQ. For example, use this page to import custom rules or scoring pages.

Note: Imported files might not be immediately available. Some file contents might require an application restart.

Note: Any include directives in the import file include files from the application server file system and not that of the client browser.

Use the Browse button to navigate to a file or enter the path manually and click **Import** to begin the download process.

Select **No role events generated for role propagation** to avoid event generation during role propagation.

When the import is complete the results are displayed on the Import from File Results page. Click **Import Another File** to go back to the Import from File page, or **Done** to return to the System Setup page.

Compliance Manager

From the Navigation bar, click the gear icon and then select **Compliance Manager**. Use the Compliance Manager page to configure control and default settings for certifications. The following table displays the available options.

Note: Most of the fields on this page enable you to configure default settings that a user can change on certifications they are reviewing or scheduling. Those fields that behave differently are described as such.

Table 29—Compliance Manager - Certification Configuration Descriptions

Field	Description
Presentation:	
Initial Access Review View	Select the view displayed when access review reports are initially accessed. **List** — open the grid view, either the worksheet or list view. **Detailed** — open the Access Review Decisions tab associated with the first item in the access review.
Default Access Review Grid View	Select the grid view to display for all identity-type access review report list pages. **Worksheet** — the individual line items that are assigned to the identities within identity-type access reviews. **Identity** — the top-level items that make up an access review, such as identities, groups, or roles.
Default Entitlement Display Mode	Select the default display mode for entitlements. Entitlement Value - display only the assigned value of the entitlement. Entitlement Description - display the longer description of the entitlement.
Subordinate Access Review Page Size	Specify the number of subordinate access reviews to show per page on the certification page before paging. If set to zero, paging is disabled.
Lifecycle:	
Notify Users of Revocations	Select to enable email notifications to users that have items revoked.
Certification Escalation Rule	Select a rule from the drop-down list as the default rule that the system uses when an access review is escalated.
When Exceptions Expire	Select the action performed on a mitigation when it expires
Active Period Duration	Input the number of units and unit type (hours, days, weeks or months) to use as the default active period duration.
Enable Challenge Period	Select to enable default challenge period and its default duration. The challenge period enables users to challenge requests from certifiers to remove access privileges.
Enable Revocation Period	**Note: If the revocation period is disabled, the certification is not scanned for completed revocations and revocation status might not be accurately reflected throughout the product.** Select to enable the default revocation period and its default duration. The revocation period places a limit on the amount of time a revoker has to act on a revocation request before that request work item is escalated.

Table 29—Compliance Manager - Certification Configuration Descriptions

Field	Description
Default Revoker	Select the user to whom all bulk remediation requests are to be sent. Bulk revocation requests are made during the certification process. You can select an item from the **Select Bulk Action** drop-down list on the Certification Report worksheet view or click **Revoke All** on the Certifications Decision tab. If this field is left blank, the remediator is specified as part of the request process.
Enable Automatic Closing	Specifies that the remediation period should be enabled, during which IdentityIQ periodically scans users to determine whether the requested remediations have been carried out. Use the following options to configure the details of this process. **Time After Certification Expiration** — Select the amount of time following this access review expiration date that IdentityIQ should wait before attempting to automatically close it. **Closing Rule** — Select the rule that IdentityIQ runs at the beginning of the automatic closing process. **Action Taken On Undecided Items** — The action that IdentityIQ assigns to any undecided items when automatically closing this access review. Choose from Approve, Revoke, or Allow Exception. **Comments** — Input the comments that IdentityIQ adds to any undecided items when automatically closing this access review. **Signer** — Select the identity who signs off on automatically closed access reviews. This setting is only configurable at the system setup level. Individuals who are scheduling certifications cannot define the signer.
Behavior:	
Require Bulk Certification Confirmation	Select to prompt user for confirmation on bulk access reviews.
Selection Count Requiring Bulk Revoke Confirmation	Input the number of selected items which require additional confirmation for bulk revocations.
Prompt for Sign Off	Select to display a pop-up window when an access review is complete and ready for sign off.
Require Electronic Signature	Select to require that, by default, all certifications require an electronic signature.
Require Subordinate Completion	Require that, by default, all subordinate access reviews be completed before the parent access review can be completed.
Auto Sign Off When Nothing to Certify	Automatically sign off the certification when assignee has nothing to certify.

Table 29—Compliance Manager - Certification Configuration Descriptions

Field	Description
Suppress Notification When Nothing to Certify	Suppress notification of certification when assignee has nothing to certify.
Require Reassignment Completion	Require that, by default, all reassigned access review items be completed before the parent access review can be completed.
Return Reassignments to Original Access Review	Specify that, by default, the content of reassigned access reviews be returned to the parent access review upon sign off. Use this option to ensure that the original content of an access review request is preserved for tracking and reporting purposes.
Automatically Sign Off When All Items Are Reassigned	Specify that an access review be automatically signed off on when all items in that access review are reassigned. **Note: This item is not available if the Required Reassignment Completion or the Return Reassignments to Original Access Review options are selected.**
Require Comments for Approval	Require that all certifiers enter comments for each item they approve in an access review request.
Require Comments When Allowing Exceptions	Require certifier to include comments when a certification decision is made.
Require a review on delegated certification items	Select to require that all access review approvers review the decision made on any user, role, entitlement, or policy violation that they delegated to another approver before they can complete the access review containing that delegation.
Require delegated certification items to be completed	Select to require that all items in a delegation work item have a decision associated with them before the work item can be marked as complete. This setting is only configurable at the system setup level. Individuals cannot change the value of this setting for a single certification.
Disable Delegation Forwarding	Select to disallow the forwarding of a work item that a different user delegated.
Allow Self Certification For	Choose which users may self-certify - that is, be the certifier for their own access, either by forwarding or reassigning an access review: All certifiers, Certification and System Administrators, System Administrators only.
Self Certification Violation Owner	For users that are not allowed to self-certify, this is the identity or workgroup that will receive any items that would require a self-certification - that is, when the reviewer and the user whose access is under review are the same person. If a Self Certification Violation Owner is not specified, any items that require self-certification will be read-only to the reviewer.
Limit Reassignments	The limit reassignment feature allows you to limit the number of times the users within the certification campaign can reassign a certificatation item.

Table 29—Compliance Manager - Certification Configuration Descriptions

Field	Description
Reassignment Limit	Set the number of reassignments allowed. **Note: Certification is not forwarded or reassigned when the reassignment limit is reached.**
Show Classifications	Set the global default to show classification data in certification access reviews. Classifications can be shown in Manager, Application Owner, Advanced, Role Membership, and Targeted certifications. This setting also determines whether classification information is shown in Separation of Duties (SOD) policy violations, in the dialog for correcting violations by revoking access.
Decisions:	
Enable Provisioning Missing Role Requirements	Enable the certifier to provision missing role requirements from within an access review.
Enable Line Item Delegation	Enables certifiers to delegate individual access review items, such as a single role or entitlement, rather than the entire identity to be reviewed. This option also enables the delegation of policy violations, either from inside an access certification or from the Manage -> Policy Violations page.
Enable Account Approval	Enable certifiers to bulk approve all accounts for a given entitlement.
Enable Account Revocation	Allow users to bulk revoke all entitlements for a given account.
Enable Account Reassignment	Enables a certifier to reassign an account and all of its associated entitlements.
Enable Overriding Violation Remediator	Enable certifiers to specify a remediator from the policy violation pop-up dialog even if there is a default remediator specified.
Enable Role Creation Requests from Certifications	Activate this field to enable certifiers to request that new roles be created from the certification pages.This setting is only configurable at the system setup level. Individuals cannot change the value of this setting for a single certification. Roles requested from the certification pages must be approved before they are available for use by the system. The assignment of role approval requests is controlled by a rule specified on the Configure System Settings Rules tab.
Enable Identity Delegation	Enable certifiers to delegate entire identities from a certification request.
Enable Allow Exceptions (applies only to non-policy violation items)	Enables certifiers to allow exceptions on access review items such as roles or entitlements, that are not policy violations. Allowing an exception means the user should not have access indefinitely, but can retain access for a specified period of time.

Table 29—Compliance Manager - Certification Configuration Descriptions

Field	Description
Deprovision Items When Exception Expires (applies only to non-policy violation items)	Enables automatic deprovisioning of access when the allowed exception period has expired. This setting applies only to items such as roles or entitlements, that are not policy violations.
Enable Allow Exception Popup	Enables certifiers to view the Allow Exception pop-up and manually set expiration dates.
Default Duration for Exceptions	Set the time period during which exceptions should be allowed. Input the number of units and unit type (hours, days, weeks or months) to use as the exception duration.
Default Operation for Remediation Modifiable Attributes	Set the default operation shown on the revocation dialog for remediation-modifiable attributes.
Show Recommendations	**Note: This option is only visible if you have purchased and activated the SailPoint IdentityAI product.** Enable recommendations from IdentityAI to display in access reviews.
Automatically Approve Recommended Items	**Note: This option is only visible if you have purchased and activated the SailPoint IdentityAI product.** Enable access review items to be automatically marked as approved by IdentityAI and move to the Access Certification Review tab for final approval.
Bulk Actions:	
Select the actions to enable from the Worksheet/Identity view and the Detail view. The actions include the following: Enable Bulk Approve Enable Bulk Revocation Enable Bulk Allow Exceptions Enable Bulk Reassignment Enable Bulk Account Revocation Enable Bulk Clear Decisions	
Certification Contents:	
Additional Entitlement Granularity	The default granularity at which additional entitlements are listed in the access review. For example, if you select **Attribute/Permission**, each permission associated with each attribute is listed, and must be acted upon, separately.
Exclude Logical Tier Entitlements	Exclude entitlements on tier application accounts from the access review. This only applies to logical applications. Tier applications are those application that make up a logical application.

Table 29—Compliance Manager - Certification Configuration Descriptions

Field	Description
Generate Certification(s)	Specify whether, by default, access review requests should generate an access review request for the specified managers, or for the specified managers and all employees below them in the reporting hierarchy. If you select **For the specified manager(s) only**, the **Flatten Hierarchy** option is displayed. Select the **Flatten Hierarchy** option to include all of the employees that report directory to the selected managers and the employees that report to their subordinate managers on the access review request.
Email Templates:	
Much of the communication performed during the access review process is done through email notifications sent automatically by IdentityIQ as an access review proceeds through its life cycle. Use this section to specify the template to use for each certification-related notice.	

Chapter 2: Lifecycle Manager Setup

The Lifecycle Manager portion of IdentityIQ Setup includes of the following:

Lifecycle Manager Configuration

Use Lifecycle Configuration to customize the availability of tools and functionality based on end user needs. Lifecycle Manager configuration is divided into the following sections:

Note: IdentityIQ System Administrators can make any request regardless of the Lifecycle Manager Configuration settings.

- See “Configure Tab” on page 59
- See “Business Processes Tab” on page 63
- See “Identity Provisioning Policies Tab” on page 63

Configure Tab

Use the Configure tab to customize your Lifecycle Manager configuration. The Configure tab includes the following.

Table 30—Lifecycle Manager Configuration — Configure Tab Options Descriptions

Field	Description
General Options	
Allow requesters to set request priorities	Use this option to enable requesters to set the priority level of their request. If this option is not selected, all requests have a default “Normal” priority level.
Enable Account Group Management	Use this option to enable provisioning of account groups through Lifecycle Manager requests.
Enable Full Text Search	Use this option to enable full text searching on the Lifecycle Manager request pages. Enabling full text searching might have some affect on the performance of those pages. For detailed information, see "Configuring Full Text Searching" on page 65. You must run the Full Text Index Refresh task before full-text search is available. Refer to the *SailPoint IdentityIQ System Administration Guide* for more information.
Base directory path used to store full text index files	The directory on the server in which full text index searches are stored.

Table 30—Lifecycle Manager Configuration — Configure Tab Options Descriptions

Field	Description
Enable automatic index refresh	Enables the automatic refreshing of the full text index at the interval specified.
Allow Searching by Population when requesting access	Enable the use of populations as a search filter.
Allow Searching by Identity when requesting access	Enable the use of identities as a search filter.
Allow opt-in to viewing request access search result details	Use this option to limit the amount of information displayed for each item on the Access Request, Review and Submit panel and add a **View Details** button on each item to show the complete information. This feature enables more items to display on each table.
Show external service request details	Use this option to display the information such as request numbers and ID from external ticketing systems throughout IdentityIQ.
Maximum number of results returned in a Request Access search	Limit the number of items returned by an access request. Large lists are hard to scan and the search should be narrowed or refined.
Maximum number of selectable users in Request Access	Limit the number of selectable users returned by an access request. Large lists are hard to scan and the search should be narrowed or refined.
Applications that support additional account requests	Use the drop-down list to specify the applications on which multiple accounts can exist or be created. Select **All Applications** to include all applications in your environment.
Request Role Options	
Request Role Options	Select the role types that are available for role requests. Any options not selected are unavailable to any user attempting to make that type of request.
When searching for roles based on population, only return roles contained by at least the following percentage of the population	Specify the minimum percentage of a population whose roles must match any given search criteria.
Request Entitlement Options	

Table 30—Lifecycle Manager Configuration — Configure Tab Options Descriptions

Field	Description
When searching for entitlements based on population, only return entitlements contained by at least the following percentage of the population	Specify the minimum percentage of a population whose entitlements must match any given search criteria.
Entitlement Search Results must return less than this number of identities when searching by identity	Indicate the maximum amount of identities an entitlement search result can yield.
Create Identity Options	
Require password on all identity creation requests.	Require a password on all identity creation requests.
Enable self-service registration	Enables new user self-registration and creates a link for registration on the IdentityIQ login page. Use the **Advanced View** option to view or configure all available variables. The securityOfficerName variable must be configured within the LCM Registration process variable before the self-service registration functionality is fully enabled. This is done using the "Compliance Manager" on page 51. The default securityOfficername is the IdentityIQ system administrator. Follow these steps to setup self-service registration: 1. From the navigation menu bar, go to **Setup** -> **Business Processes**. 2. In the Edit An Existing Process panel, select **LCM Registration**. 3. Click the **Process Variables** tab. 4. **Security Officer** is the default setting for the **Approvers** field. - To delete the **Security Officer** setting, click the **x** icon next to it. - To add another setting, click the down-arrow next to the **Approvers** field and select another entry. 5. The default entry for the **Fallback Approver** is the IdentityIQ system administrator. If desired, you can change the **Fallback Approver**. 6. When you are satisfied with all of the entries, click **Save** at the bottom of the screen.

Table 30—Lifecycle Manager Configuration — Configure Tab Options Descriptions

Field	Description
URL of action button after successful registration	Enter a URL to redirect the browser to the specified page after successful user registration. If this field is blank, the user is redirected to the login page.
Prevent pruning of new identities for this many days	Select the number of days that must pass after the creation of an identity before it can be pruned. Default is 30 days.
Manage Account Options	
Show Enable/Unlock decision buttons regardless of whether the account is disabled or unlocked.	Display the decision buttons on account management page for disabled or unlocked accounts.
Manage Account Actions	Choose which actions are enabled for Manage Accounts requests for yourself and subordinates. Options include the following: Delete Disable Enable Unlock Deselected options are unavailable to a user attempting to make that type of request. Select one or more applications from the Applications that support account only requests to specify which applications allow Account Only requests. Select All Applications to enable this feature for all applications.
Disable auto refresh account status	The status is automatically refreshed only for the accounts from applications that are not listed in the Disable auto refresh account status list AND accounts that support the Enable or Unlock feature AND accounts without the NO_RANDOM_ACESS feature. Deactivate auto refresh for account status. By default, accounts from all applications support this feature.
Applications that do not support auto refresh account status	Select one or more applications to deactivate auto refresh.
Applications that support account only requests	Select applications from the drop-down list that support request for accounts that are not associated with a role or entitlement. Select All Applications if un-associated accounts can be request for all applications.
Manage Password Options	

Table 30—Lifecycle Manager Configuration — Configure Tab Options Descriptions

Field	Description
Choose Enable password auto-generation when requesting for others to enable passwords to be auto-generated when requests are made on behalf of another user by an authorized user.	
Password Validation Rule	
Select a rule from the drop-down list to used when validating password creations.	
IdentityAI Approval Recommendations	
Enable generation of IdentityAI recommendations for approvals.	
Batch Request Approver	
Require an approval before granting batch requests.	
Manage Classification Options	
This option determines whether classification data is shown with access items (roles or entitlements) in access **requests**. This option is provided so that you can choose whether or not to alert requesters to the fact that certain roles or entitlements may allow access to sensitive or protected data. Classification data always appears in access **approvals**, regardless of this setting.	

Business Processes Tab

Use the Business Process tab on Lifecycle Manager Configuration to determine which business process is used when performing specified Lifecycle Manager actions.

Identity Provisioning Policies Tab

Identity Provisioning Policies are used to define identity attributes that must be set when creating an identity from a Lifecycle Manager request.

The following types of Identity Provisioning Policies are available:

- Create Identity
- Update identity
- Self-service Registration

Note: If an Update provisioning policy is defined, that policy overwrites the Create policy.

You must include the criteria required by the provisioning policy in the generated form before the request can be completed. Use the Provisioning Policy Editor to customize the look and function of the form fields generated from the provisioning policy.

Table 31—Provisioning Policy Editor: Identity Provisioning Policy Field Descriptions

Field Name	Description
Name	The name of your provisioning policy.
Description	A brief description of the provisioning policy.

Table 31—Provisioning Policy Editor: Identity Provisioning Policy Field Descriptions

Field Name	Description
Provisioning Policy Editor Use the Edit Provisioning Policy Fields panel to customize the look and function of the form fields generated from the provisioning policy.	
Attribute	Select the attribute field from the drop-down list to display on the form generated from the provisioning policy.
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 drop-down 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.
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 the provisioning policy. 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.
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 only for multi-valued attributes

Table 31—Provisioning Policy Editor: Identity Provisioning Policy Field Descriptions

Field Name	Description
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
Allowed Values	The value(s) which can be displayed in the field of the generated form. Choose from the following: None — the field is blank 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.

Configuring Full Text Searching

When full text searching is enable, users can use the following types of searches to find the correct access to request:

- Keyword search — Users can search based on keywords that relate to role, entitlements and descriptions.
- Affinity search — Users can search for access based on what other users who are similar to them currently have.

Feature / Enhancement	Description	Benefit
Keyword search	Search that finds results based on role and entitlement names, descriptions and extended attributes using relevance-based search and predictive analytics.	Provides a familiar shopping experience for end users. The keyword search makes it possible for end users and managers to find the right access to request.
Affinity Search	Guides users to the right access to request by enabling them to find roles or entitlements assigned to specific users or a population of users.	Enables users to locate roles and entitlements by reviewing access that others in the organization have. The affinity search provides a controlled, governance-based approach that enables you to compare similar access and view any areas of risk, such as high identity risk scores or open policy violations.

Enabling Full Text Searching

To enable the most basic full text searching:

1. From the navigation menu bar, go to **gear icon->Lifecycle Manager Configuration** page. Select the **Enable Full Text Search** option on the **Addition Options** Tab.
2. Select the **Enable Full Text Search**.
3. Run the Full Text Index Refresh task. Refer to the *SailPoint IdentityIQ System Administration Guide* for more information.

   Note: The Full Text Index Refresh must run every time you make a change to roles, managed attributes, or the FullTextIndex objects in your enterprise. The index files are only updated when this task is run. If you do not select this option, you will have to schedule the Full Text Index Refresh to run periodically or you will have to remember to run it manually.

When you run the **Full Text Index Refresh** task the first time, files for each FullTextIndex object in your IdentityIQ configuration are created.

Setting the Location of Index Files

To set the location of the index files, edit the FullTextIndex objects and add an indexPath key.

For example, `<entry key=”indexPath” value=”indexFileLocation”>` where **IndexFileLocation** is a fully qualified path name. By default the index files for roles, BundleIndex, managed attributes, ManagedAttributesIndex, and unstructured targets TargetAssociation are added to the `WEB-INF` folder of the directory where you installed IdentityIQ.

By default, after completing both steps above, you can do full text searches on the following fields:

- **Managed Attributes:** displayableName, description, and application.name
- **Roles:** name, displayableName, and description
- **Targets:** name and description

Adding Additional Fields

To add additional fields, edit the FullTextIndex objects and add a field with analyzed=”true” set: <FullTextField analyzed="true" name="*myAttribute*"/>.

The following example illustrates how to add a new full text searchable field (division) and indicate a location for the index files (/tmp/indexlocation). This example is for the roles index file.

Note: Roles are also referred to as bundles in the product code.

Field options:

- **Analyzed** – used to index the field and for full text searching. Add analyze fields to include custom attributes in full text search.
- **Indexed** – enables the field to be used in the advanced filters on the access request pages.
- **Stored** – enables the field to return in the search results and display on the access request pages, if the user interface is designed to support this use.
- **Ignored** – sets the field to not be used in full text searching nor filtering. This field does appear in the filter passed down from the user interface.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE FullTextIndex PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<FullTextIndex created="1346076712810" id="4028818239686c4f0139686c9f6900e7"
name="Bundle">
 <Attributes>
    <Map>
      <entry key="fields">
```

```
        <value>
          <List>
            <FullTextField analyzed="true" indexed="true" name="name"/>
            <FullTextField analyzed="true" indexed="true" name="displayableName"/>
            <FullTextField analyzed="true" name="description"/>
            <FullTextField indexed="true" name="assignedScope.path"/>
            <FullTextField indexed="true" name="type"/>
            <FullTextField name="defaultDescription" stored="true"/>
            <FullTextField ignored="true" name="disabled"/>
            <FullTextField name="riskScoreWeight" stored="true"/>
            <FullTextField name="owner.id"/>
            <FullTextField name="owner.name"/>
            <FullTextField name="owner.displayName" stored="true"/>
            <FullTextField name=“division” analyzed=“true” indexed=“true”>
          </List>
        </value>
      </entry>
      <entry key="indexPath" value="/tmp/indexlocation"/>
    </Map>
 </Attributes>
</FullTextIndex>
```

Special Considerations

When FullTextSearch is enabled, Bundle / Role references within filter objects in Request Object Authority rules should include only the following indexed attributes:

- name
- displayableName
- id
- description
- owner.name
- owner.id
- assignedScopePath (id of the associated scope).

Note: The only attributes that are indexed in the FullTextSearch index are listed above. If you use attributes that are not in this list, extra Bundles are returned during search, which can result in errors in the log.

Creating Direct Links to IdentityIQ

Lifecycle Manager enables you to create direct links into IdentityIQ pages from outside of the product from places such as emails, forms, or portal. These direct links can either use your single-sign on solution or require users to login to IdentityIQ as an intermediate step. Direct links can also use a number of filtering options enabling users to go directly to specific pages using specific filtering criteria.

IdentityIQ supports the following types of direct links:

Desktop Direct Links

Direct links provide a method to link directly to IdentityIQ Desktop pages. For Example, use the following direct links to go to the Manage Accounts, or Manage Passwords, or Manage Identity pages for a user that is logged in

to IdentityIQ, where <*hostName*> is the name of the host on which IdentityIQ is installed. The following direct links can be used:

- Manage Accounts
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=quickLinks/Manage+Account`
- Manage Specific Account
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=identities/<identityId>/accounts`
- Manage Password
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=quickLinks/Manage%20Passwords/identities`
- Manage Specific Password
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=identities/<identityId>/password`
- Create Identity
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=quickLinks/Create+Identity/createIdentity`
- Edit Identity
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=quickLinks/Edit+Identity`
- Edit Specific Identity
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=identities/<identityId>/edit`
- View Identity
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=quickLinks/View%20Identity/identities`
- View Specific Identity
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identities/identities.jsf&rp2=identities/<identityId>/attributes`
- Access Request Details (previously named Track My Requests)
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/identityRequest/identityRequest.jsf&rp2=requests`
- Track My Requests
  `https://<hostname>/identityiq/identityRequest/identityRequest.jsf`
- Manage Certifications
  `https://<hostname>/identityiq/certification/certifications.jsf#/certifications`
- Policy Violation List Page
  `https://<hostname>/identityiq/policyViolation/policyViolation.jsf#/policyViolations`

Mobile Interface Direct Links

Direct links provide a method to link directly to IdentityIQ Mobile pages. The following direct links can be used:

- “Direct Link to Passwords (Mobile)” on page 70
- “Direct Link to Manage Accounts (Mobile)” on page 70.
- “Direct Link to Manage Certifications (Mobile)” on page 70.
- “Direct Link to Policy Violations (Mobile)” on page 70.

- "Direct Link to Access Management Page (Mobile)" on page 70.
- "Direct Link to IdentityIQ Manage Access Review Page (Mobile)" on page 72.
- "Direct Link to Pending Work Items (Mobile)" on page 74.

Direct Link to Passwords (Mobile)

- Manage Password
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=quickLinks/Manage%20Passwords/identities`
- Manage Specific Password
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=identities/<identityId>/passwords`

Direct Link to Manage Accounts (Mobile)

- Manage Accounts
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=quickLinks/Manage%20Accounts/identities`
- Manage Specific Account
  `https://<hostname>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=identities/<identityId>/accounts`

Direct Link to Manage Certifications (Mobile)

- Manage Certifications
  `https://<hostname>/identityiq/ui/index.jsf#/certifications`

Direct Link to Policy Violations (Mobile)

- Policy Violations List Page
  `https://<hostname>/identityiq/ui/index.jsf#/listViolations`

Direct Link to Access Management Page (Mobile)

Specific access request pages can be accessed through direct links using parameters. Query parameters can be appended to the Access Review Management tab URL:

Note: Your browser may require Special characters in the parameter values to be URL encoded. For example, spaces must be replaced with %20, & must be replaced with %26, and ? must be replaced with %3F.

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/accessRequest/accessRequest.jsf&rp2=accessRequest/manageAccess/add?identityName=<identity1>&filterRoleType=<roleType1>&filterRoleStringAttr=<roleAttrib1>
```

The following parameters allow you to create direct links to the page with a variety of filters already selected:

Table 32—Access Request Management Deep Link Parameters

Type	Parameters
Identity	**identityName** (required) - name of identity the deep link is targeting.

Table 32—Access Request Management Deep Link Parameters

Type	Parameters
Role Filters	**filterRoleType** **filterRole<attribute>** **Note: Only role type and extended attributes are supported. Attributes from the bundle object are not supported.**
Entitlement Filters	**filterEntitlementApplication (multi)** **filterEntitlementAttribute (multi)** **filterEntitlementEntitlement (multi)** **filterEntitlementOwner** **filterEntitlement<attribute>** The (multi) params can be specified multiple times in a single URL. However, filterEntitlementOwner is NOT multi. If an entitlement application has only one attribute defined, the direct link can omit the entitlement attribute on the URL and the defined attribute is used by default. **Note: With the exception of Application, Attribute, and Value, only extended attributes are supported.**
Keyword Filters	**filterKeyword** **Note: If full text search indexing is enable, description is also searched for the keyword.**

Access Request for Single User Pre-Selected

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<identity1> is the name of the identity

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/accessRequest/accessRequest.jsf
&rp2=accessRequest/manageAccess/add?identityName=<identity1>
```

Access Request for Single User Pre-Selected — Filtering on Role Type

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<identity1> is the name of the user
`<roleType1>` is the requested role

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/accessRequest/accessRequest.jsf
&rp2=accessRequest/manageAccess/add?identityName=<identity1>&filterRoleType=<roleTy
pe1>
```

Access Request for Single User Pre-Selected — Filtering on Role Type and Role Extended Attribute

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<identity1> is the name of the user
<roleType1> is the type of role
<roleAttrib1> is the role attribute

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/accessRequest/accessRequest.jsf
&rp2=accessRequest/manageAccess/add?identityName=<identity1>&filterRoleType=<roleTy
pe1>&filterRoleStringAttr=<roleAttrib1>
```

Access Request for Single User Pre-Selected — Filtering on a Single Entitlement Application/Attribute/Value

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<identity1> is the name of the user
<entApp1> is the entitlement application
<entAttrib1> is the entitlement attribute (such as memberOf or groupmbr)
<entValue1> is the entitlement value

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/accessRequest/accessRequest.jsf
&rp2=accessRequest/manageAccess/add?identityName=<identity1>&filterEntitlementAppli
cation=<entApp1>&filterEntitlementAttribute=<entAttrib1>&filterEntitlementEntitleme
nt=<entValue1>
```

Access Request Logged In User Selected with Filtering on Multiple Applications

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<identity1> is the name of the user
<entApp1> and <entApp2> are the entitlement applications
<entAttrib1> and <entAttrib2> are the entitlement attributes (such as memberOf or groupmbr)
<entValue1> and <entValue2> are the entitlement values

Note: In the following example, two entitlements are requested.

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/accessRequest/accessRequest.jsf
&rp2=accessRequest/manageAccess/add?FidentityName=<identity1>
&filterEntitlementApplication=<entApp1>&filterEntitlementAttribute=<entAttrib1>
&filterEntitlementEntitlement=<entValue1>&filterEntitlementApplication=<entApp2>
&filterEntitlementAttribute=<entAttrib2>&filterEntitlementEntitlement=<entValue2>
```

Access Request Logged In User Selected with Filtering on a Keyword Search

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<keyword1> is the specific keyword you want to find

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/accessRequest/accessRequest.jsf
&rp2=accessRequest/manageAccess/add?filterKeyword=<keyword1>
```

Direct Link to IdentityIQ Manage Access Review Page (Mobile)

Specific access request review pages can be accessed through direct links using parameters. Query parameters can be appended to the Access Request Review tab URL:

Note: Your browser may require Special characters in the parameter values to be URl encoded. For example, spaces must be replaced with %20, & must be replaced with %26, and ? must be replaced with %3F.

```
https://<hostname>:<port>/ui/rest/redirect?rp1=/ui/index.jsf&rp2=certification/<id>
```

The following parameters allow you to create direct links to the page with a variety of filters already selected:

Table 33—Access Request Review Deep Link Parameters

Type	Parameters
Identity	**filterKeyword** — search term If no identityName parameter is specified, the loggedInUser is used.
Role	To specify a role or entitlement using name or id: **role (multi) — name of id of role** **entitlement (multi) — entitlement id** The (multi) params can be specified multiple times in a single URL.
Entitlements	To specify an entitlement without an id, use a combo: entitlementApplication<X> entitlementAttribute<X> entitlementValue<X> <X> corresponds to a matching integer, such as entitlementApplication1, entitilementAttribute1, entitlementValue1.

Access Request for Logged In User for a Single Role

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<role1> is the name of the role

```
https://<hostName>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=accessReque
st/review?role=<role1>
```

Access Request for a Specified User for Multiple Roles

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<identity1> is the name of the user
`<role1> and <role2> are requested roles`

```
https://<hostName>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=accessRequest/
review?identityName=<identity1>&role=<role1>&role=<role2>
```

Access Request for Logged In User for Single Entitlement Using Entitlement ID

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<entitlementId> is the identifying number for the entitlement

```
https://<hostName>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=accessRequest/
review?entitlement=<entitlementId>
```

Multiple Entitlements for Specified User Using Entitlement Application/Attribute/Value

Note: If you define only one attribute defined for an application, the entitlementAttribute can be omitted and it will be filled in automatically. In all other cases, the attribute is required. In all cases, entitlementApplication and entitlementValue are required for each entitlement combination.

In the following example,
<hostName> is the name of the host on which IdentityIQ is installed
<identity1> is the name of the user
<entApp1> and <entApp2> are the entitlement applications
<entAttrib1> and <entAttrib2> are the entitlement attributes (such as memberOf or groupmbr)
<entValue1> and <entValue2> are the entitlement values

Note: In the following example, two entitlements are requested.

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=accessRequest/
manageAccess/add&identityName=<identity1>&filterEntitlementApplication=<entApp1>&fi
lterEntitlementAttribute=<entAttrib1>&filterEntitlementEntitlement=<entValue1>&filt
erEntitlementApplication=<entApp2>&filterEntitlementAttribute=<entAttrib2>&filterEn
titlementEntitlement=<entValue2>
```

Direct Link to Pending Work Items (Mobile)

IdentityIQ supports the following mobile work items:

- Forms
- Approvals
- Request Violations

For all other types of work items, go to the desktop version of IdentityIQ and access the page associated with the work item.

You can link directly to any open work item such as a form or a violations. To access a direct link, a user must be logged in, have visibility to the work item and have authorization to access the item.

Note: Some work items, such as manager access reviews, are not supported as direct links. If a direct link contains a work item id that is not supported, a warning message displays that indicates the work item does not exist.

In the following example,
<hostname> is the name of the host on which IdentityIQ is installed
<workItemid> is the identifying number for the work item

```
https://<hostname>/identityiq/ui/rest/redirect?rp1=/ui/index.jsf&rp2=commonWorkItem
/<workItemid>
```

Using Direct Work Item Links in Email Templates

When you send an email with a direct link to a pending work item to a user, the email system variable must be configured to match server name and path of the currently deployed instance of IdentityIQ. Click the **Gear** icon in the navigation menu bar and go to **Global Settings -> Mail tab -> Email Templates -> Server Root Path**. For example, the default is set to `https://localhost:8080/IdentityIQ`. However, if you deploy from `/iiq` on port 80, you should change the setting to `https://localhost/iiq`.

Note: The $spTools.formatURL() is a velocity template function that formats the url correctly in the actual email sent to the user.

```
$spTools.formatURL('/ui/index.jsf#/commonWorkItem')/$item.id
```

Chapter 3: Lifecycle Events

Use the Lifecycle Events page to create new events or to configure existing events in your enterprise to trigger business process. When changes are detected during an identity refresh, IdentityIQ can be set up to launch event-based business processes.

Note: You must have IdentityIQ administrative capabilities to setup this function. For information about setting up administrative capabilities, contact your IdentityIQ administrator.

To access the Lifecycle Events page, navigate to **Setup -> Lifecycle Events**.

Lifecycle Events Page

The Lifecycle Events page displays the following information about existing lifecycle events:

Table 1—Lifecycle Events Page Column Descriptions

Column	Description
Name	The name assigned when the certification event was created. **Note: This name is used to identify the certification event. This name is not displayed in the certifications that are created when this event is triggered.**
Type	The event type associated with this certification event.
Attribute Name	The specified attribute when the **Event type** is set as **Attribute Change**.
Owner	The user who created the event certification.
Disabled	The Enabled/Disabled status of the event.

Use the Lifecycle Events page to edit or create a lifecycle event and the associated event behavior.

How To Create Lifecycle Events

Lifecycle events can be configured to run based on events that occur in IdentityIQ. For example, when a manager change is detected for an identity, an event-based business process can be configured to run and to send any requests to the newly-assigned manager.

Use the following parameters to set up lifecycle events:

Note: The options displayed are dependent on the event type selected.

Table 2—Lifecycle Event Options

Field Name	Description
Name	Assign an intuitive name for the event. This name is used to identify the event. This name is not displayed in the requests that are created when an event is triggered.
Description	Assign a brief description of the event.
Event Type	**Note: The fields displayed above Disabled are dependent on the Event Type specified here.** Specify an event-type. **Create** - launch a certification when a new identity is discovered. **Manager Transfer** - launch a business process when the manager changes for an identity. **Attribute Change** - launch a business process when a change is detected for the specified attribute. **Rule** - use a rule to determine when to launch a business process. To make changes to your rules, click the "..." icon to launch the Rule Editor. **Native Change** - launch a business process when a change is detected on a native application that was configured to pass this information to IdentityIQ.
Attribute	Select the identity attribute from the list to associate with this event. The attribute drop-down list contains all of the standard and extended identity attributes configured in your deployment of IdentityIQ.
Previous Manager Filter	For Manager Transfer event types only: IdentityIQ launches business processes only when identities are transferred from the specified manager. If no manager is specified, all managers are included.
New Manager Filter	For Manager Transfer event types only: IdentityIQ launches business processes only when identities are transferred from the specified manager. If no manager is specified, all managers are included.
Previous Value Filter	For Attribute Change event types only: IdentityIQ launches business processes only when the attribute value specified has changed. If no value is specified, all values are included.
New Value Filter	For Attribute Change event types only: IdentityIQ launches business processes only when the attribute value specified is newly assigned. If no value is specified, all values are included.
Disabled	Enabled / Disables status of the event.
Rule	For Rule event types only: Select the event rule used to launch business processes. Rules are created as part of the configuration process of IdentityIQ.

Table 2—Lifecycle Event Options

Field Name	Description
Include Identities	Select a rule to define the population. **None** — only the identities specified in the **Included Identities** list are in the population. **All** — include all identities in the population. **Match List** — only identities whose criteria match that specified in the list. Add identity attributes, application attributes and application permissions. Customize further by creating attribute groups to which this assignment rule applies. If the “Is Null” check box 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** — a custom database query. **Script** — a custom script. **Rule** — select an existing rule from the drop-down list. Click **Edit Rule** to launch the Rule Editor. **Population** — select an existing population.
Business Process	Select the business process triggered by this event. The business process drop-down list contains all of the standard and extended business processes configured in your IdentityIQ deployment.

Chapter 4: Working with Plugins

The SailPoint Plugin Framework is an extension framework model for IdentityIQ. It enables third parties to develop rich application and service-level enhancements to the core SailPoint platform. It enables plugins to extend the standard user interface, deliver custom REST endpoints, and to deliver custom background services.

A plugin can be a simple REST service or a full page application on top of IdentityIQ. A plugin can consist of one or all of the following components.

- A client side front end
- REST web services
- ServiceDefinition, PolicyDefinitions, and TaskDefinition implementations
- Java classes available for scripting
- Custom plugin configuration
- Database tables

During your initial installation, IdentityIQ is set up to work with plugins. A separate plugin table, identityiqPlugin, is created as part of the database schema creation scripts and the `plugins.runSqlScripts`, `plugins.importObjects`, and `plugins.enabled` properties are set to `true` in the iiq.properties file.

To disable plugins completely in IdentityIQ, set the plugin property values to `false`.

Plugin Framework

The plugin framework manages the installation and loading of plugins. It provides:

- Class path isolation on the server side
  - Implementers are free to use any 3rd party libraries or technology they choose. As long as it can be served from a REST end point, a background service, or a Java class called from scripts.
- JavaScript isolation on the client side
  - Implementers are free to use any 3rd party client side libraries.
- Core code protection
  - The framework insures and certifies no plugin overrides or changes backend product code behavior. Essential for security and upgrading.
- Web service extensions
  - Implementers can define custom REST end points to push and pull data between their plugin and the SailPoint data model.
- Plugin installation and removal
  - Plugins can be dynamically loaded to provide drag and drop installation and removal, or you can choose to require installation prior to application startup.

A plugin's user interface can be as simple as a piece of JavaScript or text injected on an existing page or a full page plugin. The behavior is defined by the `manifest.xml` in the plugin's root directory.

Working with Plugins in IdentityIQ

Note: The plugin feature must be enabled in IdentityIQ and you must have the proper access, for example System Administrator or Plugin Administrator, before this page can be displayed.

The Installed Plugins page displays and enables you to manage your plugins from within IdentityIQ. Open the page by selecting Plugins from the list under the gear icon.

From the Installed Plugins page you can install, uninstall, enable, disable, and configure your plugins.

Install — click New and either drag and drop a zip file onto the page, or navigate to the directory containing the plugins.

Enable/Disable — click the power button icon to enable or disable plugins. You will be asked to confirm your decision.

Uninstall — click the x icon to uninstall a plugin. See “Plugin Installation and Removal” on page 106 for additional information on removing a plugin.

Configure Plugins Page

The Configuration page enables you to view detailed information about the plugin, including its version, installation date, and certification level.

This page also enables you to change the values of the Settings objects for the plugin, based on the object type. If no Setting objects were defined when the plugin was created, none will display on the Configuration page.

Working with Plugins from the IdentityIQ Console

The IdentityIQ console has a number of commands that enable you to manage plugins from there, as well as perform scripted installation of multiple plugins.

The `iiq console` contains the following commands:

- **plugin install** — Installs a single plugin or multiple plugins in a directory, see “Install” on page 80
- **plugin upgrade** — Upgrades a plugin, “Upgrade” on page 81
- **plugin uninstall** — Uninstalls a plugin, “Uninstall” on page 81
- **plugin enable** — Enables a plugin, “Enable” on page 81
- **plugin disable** — Disables a plugin, “Disable” on page 82
- **plugin export** — Exports a plugin and all of its current configuration to a zip file in a specified directory, “Export” on page 82
- **plugin status** — View the enabled status of a plugin, all plugins or whether or not plugins are enabled globally as defined in the iiq.properties file, “Status” on page 82
- **plugin list** — View a list of all installed plugins, “List” on page 82
- **plugin classes** — “Classes” on page 82

Install

Install a single plugin or multiple plugins. Either a path to the zip file of the plugin or a directory containing multiple plugin zip files can be specified. If a directory is specified, any zip file in that directory is installed.

Flags:

- file — path to a plugin file
- dir — the directory containing the plugin zip file to install
- no-cache — the plugin should not be cached after install

Upgrade

Note: Refer to the “Plugin Versioning Requirements” on page 95

Upgrade to a newer version of a plugin.

Upgrading a plugin to the same version or a previous version is not supported. While developing a plugin, this behavior can be disabled for easier testing. To do so, include a "-dev" suffix on the version, for example, 2.0-dev.

The version of a plugin can either be official or development.

Development versions end with the suffix '-dev,' for example, 2.0-dev, and bypass most version checks so that the plugin can be recompiled, upgraded and tested easily.

Official versions drop the '-dev' suffix and can only be installed over a development version or an earlier official version. The minimum upgradeable version must also be valid.

Valid upgrade paths:

- 1.0 -> 2.0-dev
- 2.0-dev -> 2.0-dev
- 2.0-dev -> 2.0
- 1.0 -> 2.0

Invalid upgrade paths:

- 2.0 -> 2.0
- 2.0 -> 1.0

Flags:

- file — path to a plugin file
- no-cache — the plugin should not be cached after the upgrade

Uninstall

Uninstall a plugin.

Flags:

- id — plugin id
- name — plugin name

Enable

Enable the plugin.

Flags:

- id — plugin id
- name — plugin name
- no-cache — the plugin should not be cached after being enabled

Disable

Disable a plugin.

Flags:

- id — plugin id
- name — plugin name

Export

Export a single or all installed plugins to their respective zip files and, optionally, a specified directory.

Flags:

- id — plugin id
- name — plugin name
- * — export all installed plugins
- dir — the directory in which to save the zip files. If the directory does not exist, the command will attempt to create one. If none is specified, the files are save to the current working directory.

Status

The enabled status of a single plugin, all installed plugins or the system-wide enabled status of plugins.

Flags:

- id — plugin id
- name — plugin name
- * — view the status of all plugins
- no flag — system-wide status of plugins as defined in iiq`.properties`

List

The list of all install plugins.

Classes

The list of the classes available (from a plugin or all plugins), and the intended use for each class.

Flags:

- id — plugin id
- name — plugin name

* — the list of all available classes from all plugins

Developing Plugins

IdentityIQ stores the .zip archive file of the Plugin in the IdentityIQ database in a data LONGBLOB in the spt_file_bucket table. The data in the spt_file_bucket table is referenced ID to an entry in the spt_persisted_file table.

Plugins are loaded from this .zip file after installation or after an application server restart. The .zip file is extracted, and all important files are cached for later use. There are several accessor methods to reference the cached files, but they can also be referenced by the url prefix `/identityiq/plugin/pluginName` followed by the path found in the build structure. Compiled java classes are loaded and cached from the .zip archive using the PluginClassLoader class.

Plugin Versioning Requirements

Note: The single exception to these requirements are version numbers with -dev appended to the end. This suffix causes version number validation to be bypassed.

To provide better support for upgrading plugins, we have set new requirements for plugin version number formats. Plugin version numbers must be numeric, contain no alphabetic or other characters, and separate the elements of the version number with decimal points. Within each segment of the version number, the values between the decimal points, the values are cast as integers, and leading zeroes are trimmed.

For example:

- `04` and `00004` are both interpreted as `4`
- A segment containing any non-numeric values is interpreted as `0`
- `1.004.alpha` is parsed as `1.4.0`
- `2.3.4a` will be parsed as `2.3.0`

Plugin Object Model

A plugin is defined in IdentityIQ by the Plugin XML object. This object defines the parameters of the plugin, for example items such as REST Resources, Snippets, Widgets, and Settings. This Plugin object is defined in the `manifest.xml` file. The Plugin Object is an XML object that defines the features of the plugin. This object tells IdentityIQ what features are in your plugin by defining them as attributes of a Plugin Object. In the Plugin Object you also define items such as the name of the plugin, the rights required for using the plugin, version, snippets, and REST resources.

The following attributes are included in the plugin model:

Table 3—Plugin Model Attributes

Attribute Name	Description
name	Unique Name of the Plugin
installDate	Date that plugin is installed
displayName	Display Name for the plugin
disabled	Status of the plugin
rightRequired	What SPRIGHT is required for this plugin
version	The version of the plugin
minSystemVersion	The minimum version of IdentityIQ that the plugin will run on

Table 3—Plugin Model Attributes

Attribute Name	Description
maxSystemVersion	The maximum version of IdentityIQ that the plugin will run on
attributes	List of configurable attributes
file	Reference to the persisted file in the database

Plugin Structure

A plugin will consist of the following components:

- Manifest file
- Build file(s)
- Database Scripts
- UI Elements
- XML Artifacts
- Java Classes
- Java JAR libraries

Not all of these components are required for a plugin - it can be as basic as the manifest, and some javascript/xhtml pages. In order to understand how a plugin operates, and how best to create one, it is important to understand what each of these components does, and how they interact.

Plugin Manifest File

A plugin is defined in IdentityIQ by the Plugin XML object that defines the parameters of the plugin. For example, features such as REST resources, Snippets, Settings. The Plugin object is defined in the `manifest.xml` file. This is a required artifact.

For more complex plugins that require support for other field types, and more dynamic behavior, such as drop down lists or password fields, use the advanced plugin settings to define a form or reference a custom plugin configuration file.

Dynamic behavior might include showing or hiding additional fields depending on previous selections. For example, if a user chooses basic authentication, a username and password field would appear, but, if oauth authentication is chosen, it might be more appropriate to show an access token field.

Plugin Settings

Note: If your plugin requires more than simple input fields, string, boolean or int values, you must use the plugin advanced settings.

Plugin settings are attributes that are available for modification as part of the installation. Click **Configure** to display the configuration settings page. Settings are displayed as a form. If the plugin does not use the plugin advanced settings, the form is created automatically.

Settings from the manifest file are listed, in order, on the plugin settings page.

The Plugin setting object can be used to represent a single setting on the configuration settings page for a Plugin. Each object is used to represent a single configurable setting on the settings page.

Table 4—Plugin Settings

Attribute Name	Description
allowedValues	List of allowed values for population of a dropdown
dataType	The type of the setting, for example, "string" or "int" or "boolean"
defaultValue	The default value for the setting
helpText	Associated help text for the setting
label	Label to be displayed for the setting
name	Name of the current setting
value	Value for the setting

Plugin Advanced Settings

Plugin advanced settings are used to define forms or reference a custom plugin configuration file to define more complex plugins.

- settingsForm — define a plugin using a form
- settingsPage — reference a custom HTML/JS file

settingsForm

To use a form for plugin configuration, you can build a form using the form builder then copy and paste that form into the manifest file, or you can build the form directly into the manifest.

Values entered into a form can be accessed using the FormData.values. FormService has functions to assist with validating required fields and displaying errors. Additional validations are built into the HTML and AngularJS code based on the form design, which means Angular will set a field to undefined if it is not valid. These validations can be used to prevent a form from being submitted and show error messages if necessary.

settingsPage

Develop your own configuration settings page by providing the required HTML and javascript. You can use whatever frameworks you prefer for your settings, but they need to fit in with whatever IdentityIQ has loaded. For example, angular is not required, but you can use it.

To use angular frameworks, see “SailPoint Angular Components” on page 93.

Use the settingsPage setting to specify the name of your custom configuration settings page, for example, `config.xhtml`.

Snippets

Snippets are small, configurable snippets of code that can be injected into the rendering of normal IdentityIQ user interface pages. For example, you can insert a menu option, a button, or even a larger set of interface components into an IdentityIQ page.

Snippets must be specified in the plugin’s manifest file. They use a regular expression pattern to identify the IdentityIQ pages where the snippet should execute, and therefore appear.

You can have multiple snippet components in the same plugin, some that apply globally, and some that apply to specific targeted pages. You need to define a separate `js` file for each location a snippet applies, and then specify a separate snippet in the manifest file with the right regexPattern to run it on the appropriate pages.

The details for the user interface component's contents, and its placement within the page, are specified in the JavaScript file.

A snippet contains four equally important components:

Table 5— Snippet Components

Component Name	Description
regexPattern	This is a regular expression pattern that is run against the current URL in the browser - if the URL matches the pattern, the Snippet will attempt to displayed
rightRequired	This determines the scope of users allowed to view the Snippet element - should reference an IdentityIQ SPRight object
scripts	This is a list of the scripts to run when a particular URL matches the regexPattern. Normally this will consist of injecting an element into the DOM of the page. The example `hearder.js` file uses JQuery
styleSheets	List of any css files that are required by Snippet Scripts

Widgets

A Widget is a targeted snippet – one that inserts a block of user interface components into a fixed area of the Home page that can be added selectively for different users, as a unit.

Widgets can be configured to appear on the Home page for any or all IdentityIQ users.

The first thing you need, to implement a plugin Widget, is the Widget object itself. When you import that object into IdentityIQ during plugin installation, it defines the existence of the Widget making it available for any user.

Widget objects are simple, the only details about the user interface component that get defined in the object are its name and title.

Widgets require a snippet definition in the manifest file for this plugin. This snippet defines the home page hook for the widget. The regular expression pattern for the widget snippet must specify the IdentityIQ home page.

The rest of the snippet definition has IdentityIQ execute the contents of the specified JavaScript file when it loads pages that meet the regex pattern.

The contents of the JavaScript file then define both the user interface layout, in the form of a directive, and the controller for the Widget. Because the home page is an Angular page, this JavaScript must specify an Angular controller for the widget.

The name given to this directive must follow a fixed naming convention. It must be specified in relation to the name given to the widget object. Specifically, the widget object name must be prefixed with sp and suffixed with Widget. So the Search widget object requires a directive called sp*WidgetName*Widget.

The directive references the controller for the widget. That controller is also defined within the widget's JavaScript file and defines the variables that serve as the model for the view elements and performs the required operations to set their data values.

Plugin Build File

Note: Complications can arise when the plugin is built using a different version of java than the version deployed on the application servers hosting IdentityIQ. Parametrize the javac argument in the build.xml file with the most compatible Java version available. To do this, add the property target to the javac directive, and set equal to whatever version is being targeted.

For example:
<javac srcdir="${pluginSrc}" destdir="${pluginClasses}"
includeantruntime="false" target="1.7">

Apache Ant is a readily available tool that can be used to package plugins prior to deployment and distribution. To provide build specific values, the standard is to also include a `build.properties` file with a simple key-value pair for all build specific tokens.

The following example illustrates how a properties file can be leveraged to enable multiple developers to use the same build process, despite having dissimilar build environments. The actual `build.xml` file is responsible for creating the build directory, compiling any java classes, packaging those compiled classes into a .jar archive, and archiving in .zip format the complete plugin.

```
jdk.home.1.7=/Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk
iiq.home=/usr/local/apache-tomcat-8.0.30/webapps/identityiq/
pluginName=TodoPlugin
version=2.0.0
```

Plugin Database Scripts

Plugins that require persistence of data outside of that allowed by the IdentityIQ object model require at minimum the creation, updating, and deletion of unique tablespace. The plugin framework creates a database named identityiqPlugin. The creation of this database is handled by the installation scripts packaged with every download of IdentityIQ, in the `WEB-INF/database` folder. Additionally, a default user identityiqPlugin is created to perform operations, installation and deletion of plugins, on this new database. Similar to the base IdentityIQ username and password, these can be modified and updated in the `IdentityIQ iiq.properties` file located in `WEB-INF/classes/iiq.properties`.

When creating a plugin, you must create a folder named db in your project directory. This folder should be further subdivided into three operation specific folders: install, uninstall, and upgrade.

The scripts placed in these folders are automatically run when a plugin is installed or deleted. It is recommended that you include scripts for the four major database types supported by IdentityIQ, MySQL, SQLServer, DB2, and Oracle, or note in your documentation which databases are supported. Database specific scripts must include the database type as the file extension, for example `.mysql`. The `upgrade` folder should contain any deltas in table definitions from prior versions of the plugin.

Plugin User Interface Elements

Most plugins have some additional user interface component that appears in IdentityIQ. Images, CSS files, HTML templates, and JavaScript can all be used to provide the interactions and views required by the plugin. Plugins that use a fullPage element look for a file called `page.xhtml` in the build.

To extend the classes loaded with your plugin to the rest of IdentityIQ, you must specifically declare those classes in the manifest file.

Plugin Authorization

To prevent unauthorized access to your new endpoints, each should be guarded with an authorization mechanism. You can constrain which users can see and access the user interface components, and you can secure the REST endpoints you build into your plugin.

When you define snippets, including widget plugin components, in the manifest file for a plugin, you can specify a rightRequired attribute to constrain access. This attribute names a SailPoint SPRight which users must be assigned for the component to appear in their IdentityIQ instance.

You can also specify a rightRequired at the Plugin object level, in the manifest file, which will specify the required SPRight for a user to be able to access the full-page component of the plugin.

If you leave these rightRequired attributes off, all IdentityIQ users will be able to access those plugin components.

SPRights are the most granular permission object in IdentityIQ. In most cases, users are assigned SPRights in IdentityIQ by attaching those rights to one or more Capability objects and then granting the Capability to the appropriate users.

If the plugin contains a full-page component that users can access through a quicklink, the Quicklink access will be governed by Quicklink Populations, like any other IdentityIQ Quicklink.

User must also be authorized to the full page itself, with the rightRequired specified in the plugin manifest, to be able to view the page.

Other authorization points of note. First, whether or not you explicitly authorize system administrators to these components, they will have full visibility and access to them. Second, when you include a widget in your plugin, the widget will appear in the list of available widgets for all users when they are editing their home page and deciding which content to include there. However, if they are not authorized to the widget's snippet, and they add that widget to their Home page, IdentityIQ will add an empty widget and they will neither be able to see nor interact with any of its functional elements.

To secure the endpoints the plugin framework use Annotations. In Java, an annotation is a syntactic metadata that is added, often before a method signature, that describes the parameters used in that method.

An annotation should have at least three parts

- The HTTP method (GET, POST, PUT, DELETE, etc)
- The path or endpoint - this can be parameterized which is useful for pulling back a single record. The above example uses parameterization by adding the variable within {} tags to the end of the URL, and also declaring the @PathParam appName in the input arguments of the method signature
- The authorization of the method - the allowed values are:
  - **@AllowAll** - this allows anyone to interrogate the endpoint
  - **@RequiredRight("<SPRight>")** - allows users who possess the named SPRight to access the endpoint
  - **@SystemAdmin** - system administrator access only
  - **@Deferred** - Authorization is deferred to the method. When this option is selected, you must also create an Authorizer class that implements the sailpoint.authorization.Authorizer interface. The Authorizer class should overwrite the authorize(UserContext) method of the base Authorizer interface. Inside of the REST resource method, the author would then call authorize().

Plugin XML Artifacts

Any IdentityIQ objects that are required as part of a plugin need to be represented in XML artifacts. This could be something as small as a single new SPRight object or a complex workflow or rule. The mechanism that is used for importing these artifacts during installation is the same as any IdentityIQ object import, so the normal import actions are also available, merge, include, execute, logConfig.

Development of these XML artifacts can be done directly in the build folder, or in the IdentityIQ user interface and either exported using the console or copy and pasted from debug into the build.

When developing in the user interface and then migrating to your build folder using cut and paste, you must remove the id attribute assigned by Hibernate and any other hibernate ID value references. For this reason, it is preferable to export the artifacts using the IdentityIQ console command **./iiq export -clean**.

Everything in the `import` folder is imported The objects can be separated into individual files, or combined into a single file. When a plugin is uninstalled, the XML artifacts that were imported remain in the IdentityIQ database, but the `.zip` archive from which the plugin files was loaded, is removed from the spt_file_bucket and spt_persisted_file tables.

Plugin Java Classes

Plugins are a powerful productivity-enabler, that give users the ability to extend both the IdentityIQ user interface and server in a well-defined manner.

Plugin Java Classes - REST Classes

The plugin framework relies heavily on REST web services integration for the majority of CRUD (create, read, update, and delete) operations. To create a custom REST Resource:

- Extend the BasePluginResource class
- Secure the new endpoints

Extend BasePluginResource

The BasePluginResource class should be used as the base class for all resources. It provides access to utility methods for accessing plugin settings, getting database connections, and more.

- **getConnection** - gets connection to the datasource specified in the `iiq.properties` file for the plugins
- **getPluginName** - this method should be overwritten to return the correct name of the plugin
- **getSettingBool** - gets value of boolean plugin setting for plugin name returned by getPluginName()
- **getSettingInt** - gets value of int plugin setting for plugin name returned by getPluginName()
- **getSettingString** - gets value of String plugin setting for plugin name returned by getPluginName()
- **prepareStatement** - convenient security method for getting Java PreparedStatement object for any database queries that are required
  - signature is prepareStatement(Connection, String, Object...) where the String would be the SQL statement you want to execute
  - Object... would be a list of the parameters values, if any, to be used
- **authorize** - should be overwritten by implementers, but by default only ensures that SystemAdministrator can see everything

Introduce additional methods to handle the various endpoints required by the plugin.

Plugin Java Classes - Plugin Executors

The plugin framework enables you to include custom task implementations or services with your plugin. These items rely on executor classes that contain the business logic for these services. The following executors are currently available:

- Service Executors
- Task Executors
- Policy Executors

You must specifically declare the classes to be exported for each of the executors. Only classes specifically declared are accessible from the rest of IdentityIQ. If a class is not declared, it will fail to instantiate when you open the plugin. Classes are declared in the `manifest.xml` file.

Use the following attributes to declare classes:

- serviceExecutor
- taskExecutor
- policyExecutor

For example:

```
<entry key="taskExecutors">
  <value>
    <List>
      <String>com.acme.TaskExecutor1</String>
      <String>com.acme.TaskExecutor2</String>
    </List>
  </value>
</entry>
```

Plugin Object Properties

When defining your plugin object you must provide the list of service executors that are included. The list lives inside an attributes map under the key serviceExecutors.

- Plugin Helper methods
- All inherited Service methods
- BasePluginTaskExecutor
- Plugin Helper methods
- All inherited TaskExecutor methods
- BasePluginPolicyExecutor
- Plugin Helper methods
- All inherited PolicyExecutor methods.

Plugin Helper Methods

The list of methods that are included with the BasePlugin classes are as follows:

- getPluginName() - returns a string value of the name of the plugin
- getConnection() - returns a Connection object used to query the database
- getSettingString(String settingName) - returns a String setting value from the Plugin Settings
- getSettingBool(String settingName) - returns a boolean value from the Plugin Settings
- getSettingInt(String settingName) - returns a integer value from the Plugin Settings

You can think of the BasePlugin classes as foundation for creating your specific objects. By using them you gain access to the Plugin Helper Methods, but you are not required to use the BasePlugin classes. You can extend directly from the parent class object.

Implementing a Plugin Service Definition

To implement a Plugin Service there are two parts. The service class, containing the business logic that you want the service to actually do, and the service definition xml, that is loaded into IdentityIQ.

- BasePluginService Class — an abstract class that extends the Service class, as well as implements the PluginContext interface. You can use this class as the foundation for your custom Plugin Service
- Service Definition — specify a pluginName attribute. This tells IdentityIQ to use the plugin class loader for this executor. If this attribute is not specified the executor class will not be found

Implementing a Plugin Task Executor

To implement a Plugin Task Executor there are have two parts. The Task Executor class, which handles the business logic for your task, and the TaskDefinition xml object, which gets loaded into IdentityIQ.

- BasePluginTaskExecutor Class — an abstract class that extends the AbstractTaskExecutor class, as well as implements the PluginContext interface. You can use this class as the foundation for your custom Plugin Executor task
- TaskDefinition — include the pluginName attribute, as this attribute tells IdentityIQ to use the plugin class loader instead of default class loader. If the attribute is not specified the executor class will not be found

Implementing a Policy Executor

To implement a Policy Executor there are have two parts. The Policy Executor class, which handles the business logic for your policy, and the Policy xml object, which gets loaded into IdentityIQ.

- BasePluginPolicyExecutor Class — an abstract class that extends the AbstractPolicyExecutor class, as well as implements the PluginContext interface. You can use this class as the foundation for your custom Plugin Policy Executor
- Policy XML — include the pluginName attribute, as this attribute tells IdentityIQ to use the plugin class loader instead of default class loader. If the attribute is not specified the executor class will not be found.

Plugin Java Classes - Script Classes

Note: Beanshell executions are referred to as scripting in this document.

The classes installed for plugins can be made available to all beanshell executions. Beanshell (rules, scriptlets, workflows) scripting invocations are able to use all Java classes, from all plugins, that are declared as exported for scripting. The scripts should fail to load classes which are in plugins, but which are not explicitly exported.

Beanshell executions include:

- rules
- workflow steps (rules and scripts)
- scriptlets

Use the scriptPackages attribute to declare the Java packages exported for use by scripts as follows:

```
<entry key="scriptPackages">
  <value>
    <List>
      <String>com.acme.classy.util</String>
    </List>
  </value>
</entry>
```

Example: Using Plugin Classes From a Rule

This is a simple example of how to call the plugin classes from a rule.

The example Java class named `com.acme.classy.util.ClassyUtil` can be used from a rule if declared properly in the `manifest.xml` using the `sriptPackages` entry shown above.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Rule PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<Rule language="beanshell" name="ClassyRule" >
  <Description>Returns the current timestamp by calling class from
                      ClassyPlugin</Description>
```

```
  <Signature returnType="string"/>
  <Source>
import com.acme.classy.util.ClassyUtil;
long now = ClassyUtil.now();
return "The current timestamp is: " + now;
 </Source>
</Rule>
```

SailPoint Angular Components

To implement the SailPoint-styled angular components, your project needs to include the SailPointBundleLibrary JavaScript file. There are specific directive dependencies on this library when you use the SailPoint-styled components.

Widgets and snippets donot require this library in the plugin project, because for those, the plugin architecture automatically references this library from your IdentityIQ instance. But if you are going to use these in a full page plugin implementation, your plugin needs to include a copy of this JavaScript library, copied from your IdentityIQ version. Plus, your `page.xhtml` component needs to explicitly reference the JavaScript file in a script element so IdentityIQ can resolve the SailPoint Angular directives you are be using. This library needs to match the version of IdentityIQ where the plugin will be deployed.

Your angular module definition needs to specify any other modules that your module has dependencies on. This list will vary, depending on the elements you include. For example, if you are using modal boxes, you need to include `sailpoint.modal`, tree components rely on `sailpoint.tree`, and so on. If your user interface only contains some of the more basic of the field types, you need fewer modules in your dependency list.

An example plugin that demonstrates the different components that are part of the SailPoint Angular bundle, called the Kitchen Sink plugin is installed with IdentityIQ. The plugin is available for download from Compass. Once the plugin is downloaded, you can install and view the plugin through IdentityIQ to see how the different field types behave. Then you can examine the HTML and JavaScript files to see the directives involved and how they are populated.

The library contains a wide array of field types, just like you might see throughout core product pages. For example, selection lists, plain text entry fields or fields where you can enter multiple separate text values as a list, dropdown list boxes where you can select from a set of choices, and a special directive for when the choices should be Boolean true-false values, date pickers, checkboxes, and radio buttons.

The library offers SailPoint styled buttons, and the demo plugin shows how you can use them to attach logic to do things like open modal boxes of various types, post an alert notification to the page or navigate the user to another page in IdentityIQ while preserving navigation history.

Beyond the simpler field types and buttons, there are a few components that create more complex elements. For example, a tree directive for showing nested relationships in data, a directive for displaying a list of cards that is configured for paging when result sets are large, and a directive for a data table that matches the tables in the Angular pages of IdentityIQ.

Internationalization

Message catalog files are specified per language with the two character abbreviation for the corresponding language, or the four-character options when you have locale-specific message catalogs, for example, `TrainingPlugin_fr_ca.properties` for Canadian French. These files should be recorded in the messages folder of your plugin project.

To guard against collisions with IdentityIQ base product message key names, or message keys from other plugins, the best practice is to name your plugin's message keys with a prefix that makes them unique to your plugin. For example, consider using your plugin's name as a prefix.

Both the user interface and server side code can access the provided catalogs.

In the user interface, full page plugins and widget plugins use two different mechanisms, because of differences in library support that are included in the pages.

In a Full page plugin component, the HTML uses the `msgs` function to translate the text.

In Widgets or other snippets that display text, the `spTranslate` function in the SailPoint Bundle Library does the translation, by piping the message key through it.

It is possible to use this syntax in a full page plugin, but only if you have included the SailPoint Angular Bundle Library and have declared a dependency for your angular module on the `sailpoint.i18n module`. The `spTranslate` function is part of that module.

In server-side code, localization is done with the Message object's localize method, passing it the message catalog key to look up and translate.

Plugin Installation and Removal

To install a plugin, click the gear icon and select **Plugins** to navigate to the Installed Plugins page. Click **New** and drag and drop or upload the plugin `.zip` file.

If you downloaded a plugin, the `.zip` file should be included with the download. If you developed the plugin yourself, the `.zip` file is in your project directory under `build/`*`your plugin name`*`/dist`.

When a plugin is installed, the database scripts from the `db/install` folder run, which creates any tables necessary for the plugin, the XML configuration files are imported into the IdentityIQ database from the `import/install` folder, any compiled classes are loaded into the unique plugin classloader, and the manifest file is imported creating the Plugin object.

Remove a plugin by clicking **X** on the appropriate Plugin card on the Installed Plugins page. Database scripts in charge of cleaning up data run from the `db/uninstall` folder and the manifest file (the Plugin object) is removed.

Addition steps might need to be taken to edit the System Configuration file to remove objects created by the plugin, such as quicklinks, or to remove tables created in the plugin database.

Developing Plugins

IdentityIQ stores the .zip archive file of the Plugin in the IdentityIQ database in a data LONGBLOB in the spt_file_bucket table. The data in the spt_file_bucket table is referenced ID to an entry in the spt_persisted_file table.

Plugins are loaded from this .zip file after installation or after an application server restart. The .zip file is extracted, and all important files are cached for later use. There are several accessor methods to reference the cached files, but they can also be referenced by the url prefix `/identityiq/plugin/`*`pluginName`* followed by the path found in the build structure. Compiled java classes are loaded and cached from the .zip archive using the PluginClassLoader class.

Plugin Versioning Requirements

Note: The single exception to these requirements are version numbers with -dev appended to the end. This suffix causes version number validation to be bypassed.

To provide better support for upgrading plugins, we have set new requirements for plugin version number formats. Plugin version numbers must be numeric, contain no alphabetic or other characters, and separate the elements of the version number with decimal points. Within each segment of the version number, the values between the decimal points, the values are cast as integers, and leading zeroes are trimmed.

For example:

- `04` and `00004` are both interpreted as `4`
- A segment containing any non-numeric values is interpreted as `0`
- `1.004.alpha` is parsed as `1.4.0`
- `2.3.4a` will be parsed as `2.3.0`

Plugin Object Model

A plugin is defined in IdentityIQ by the Plugin XML object. This object defines the parameters of the plugin, for example items such as REST Resources, Snippets, Widgets, and Settings. This Plugin object is defined in the `manifest.xml` file. The Plugin Object is an XML object that defines the features of the plugin. This object tells IdentityIQ what features are in your plugin by defining them as attributes of a Plugin Object. In the Plugin Object you also define items such as the name of the plugin, the rights required for using the plugin, version, snippets, and REST resources.

The following attributes are included in the plugin model:

Table 6—Plugin Model Attributes

Attribute Name	Description
name	Unique Name of the Plugin
installDate	Date that plugin is installed
displayName	Display Name for the plugin
disabled	Status of the plugin
rightRequired	What SPRIGHT is required for this plugin
version	The version of the plugin
minSystemVersion	The minimum version of IdentityIQ that the plugin will run on

Table 6—Plugin Model Attributes

Attribute Name	Description
maxSystemVersion	The maximum version of IdentityIQ that the plugin will run on
attributes	List of configurable attributes
file	Reference to the persisted file in the database

Plugin Structure

A plugin will consist of the following components:

- Manifest file
- Build file(s)
- Database Scripts
- UI Elements
- XML Artifacts
- Java Classes
- Java JAR libraries

Not all of these components are required for a plugin - it can be as basic as the manifest, and some javascript/xhtml pages. In order to understand how a plugin operates, and how best to create one, it is important to understand what each of these components does, and how they interact.

Plugin Manifest File

A plugin is defined in IdentityIQ by the Plugin XML object that defines the parameters of the plugin. For example, features such as REST resources, Snippets, Settings. The Plugin object is defined in the `manifest.xml` file. This is a required artifact.

For more complex plugins that require support for other field types, and more dynamic behavior, such as drop down lists or password fields, use the advanced plugin settings to define a form or reference a custom plugin configuration file.

Dynamic behavior might include showing or hiding additional fields depending on previous selections. For example, if a user chooses basic authentication, a username and password field would appear, but, if oauth authentication is chosen, it might be more appropriate to show an access token field.

Plugin Settings

Note: If your plugin requires more than simple input fields, string, boolean or int values, you must use the plugin advanced settings.

Plugin settings are attributes that are available for modification as part of the installation. Click **Configure** to display the configuration settings page. Settings are displayed as a form. If the plugin does not use the plugin advanced settings, the form is created automatically.

Settings from the manifest file are listed, in order, on the plugin settings page.

The Plugin setting object can be used to represent a single setting on the configuration settings page for a Plugin. Each object is used to represent a single configurable setting on the settings page.

Table 7—Plugin Settings

Attribute Name	Description
allowedValues	List of allowed values for population of a dropdown
dataType	The type of the setting, for example, "string" or "int" or "boolean"
defaultValue	The default value for the setting
helpText	Associated help text for the setting
label	Label to be displayed for the setting
name	Name of the current setting
value	Value for the setting

Plugin Advanced Settings

Plugin advanced settings are used to define forms or reference a custom plugin configuration file to define more complex plugins.

- settingsForm — define a plugin using a form
- settingsPage — reference a custom HTML/JS file

settingsForm

To use a form for plugin configuration, you can build a form using the form builder then copy and paste that form into the manifest file, or you can build the form directly into the manifest.

Values entered into a form can be accessed using the FormData.values. FormService has functions to assist with validating required fields and displaying errors. Additional validations are built into the HTML and AngularJS code based on the form design, which means Angular will set a field to undefined if it is not valid. These validations can be used to prevent a form from being submitted and show error messages if necessary.

settingsPage

Develop your own configuration settings page by providing the required HTML and javascript. You can use whatever frameworks you prefer for your settings, but they need to fit in with whatever IdentityIQ has loaded. For example, angular is not required, but you can use it.

To use angular frameworks, see “SailPoint Angular Components” on page 105.

Use the settingsPage setting to specify the name of your custom configuration settings page, for example, `config.xhtml`.

Snippets

Snippets are small, configurable snippets of code that can be injected into the rendering of normal IdentityIQ user interface pages. For example, you can insert a menu option, a button, or even a larger set of interface components into an IdentityIQ page.

Snippets must be specified in the plugin’s manifest file. They use a regular expression pattern to identify the IdentityIQ pages where the snippet should execute, and therefore appear.

You can have multiple snippet components in the same plugin, some that apply globally, and some that apply to specific targeted pages. You need to define a separate `js` file for each location a snippet applies, and then specify a separate snippet in the manifest file with the right regexPattern to run it on the appropriate pages.

The details for the user interface component's contents, and its placement within the page, are specified in the JavaScript file.

A snippet contains four equally important components:

Table 8— Snippet Components

Component Name	Description
regexPattern	This is a regular expression pattern that is run against the current URL in the browser - if the URL matches the pattern, the Snippet will attempt to displayed
rightRequired	This determines the scope of users allowed to view the Snippet element - should reference an IdentityIQ SPRight object
scripts	This is a list of the scripts to run when a particular URL matches the regexPattern. Normally this will consist of injecting an element into the DOM of the page. The example `hearder.js` file uses JQuery
styleSheets	List of any css files that are required by Snippet Scripts

Widgets

A Widget is a targeted snippet – one that inserts a block of user interface components into a fixed area of the Home page that can be added selectively for different users, as a unit.

Widgets can be configured to appear on the Home page for any or all IdentityIQ users.

The first thing you need, to implement a plugin Widget, is the Widget object itself. When you import that object into IdentityIQ during plugin installation, it defines the existence of the Widget making it available for any user.

Widget objects are simple, the only details about the user interface component that get defined in the object are its name and title.

Widgets require a snippet definition in the manifest file for this plugin. This snippet defines the home page hook for the widget. The regular expression pattern for the widget snippet must specify the IdentityIQ home page.

The rest of the snippet definition has IdentityIQ execute the contents of the specified JavaScript file when it loads pages that meet the regex pattern.

The contents of the JavaScript file then define both the user interface layout, in the form of a directive, and the controller for the Widget. Because the home page is an Angular page, this JavaScript must specify an Angular controller for the widget.

The name given to this directive must follow a fixed naming convention. It must be specified in relation to the name given to the widget object. Specifically, the widget object name must be prefixed with sp and suffixed with Widget. So the Search widget object requires a directive called sp*WidgetName*Widget.

The directive references the controller for the widget. That controller is also defined within the widget's JavaScript file and defines the variables that serve as the model for the view elements and performs the required operations to set their data values.

Plugin Build File

Note: Complications can arise when the plugin is built using a different version of java than the version deployed on the application servers hosting IdentityIQ. Parametrize the javac argument in the build.xml file with the most compatible Java version available. To do this, add the property target to the javac directive, and set equal to whatever version is being targeted.

For example:
<javac srcdir="${pluginSrc}" destdir="${pluginClasses}"
includeantruntime="false" target="1.7">

Apache Ant is a readily available tool that can be used to package plugins prior to deployment and distribution. To provide build specific values, the standard is to also include a `build.properties` file with a simple key-value pair for all build specific tokens.

The following example illustrates how a properties file can be leveraged to enable multiple developers to use the same build process, despite having dissimilar build environments. The actual `build.xml` file is responsible for creating the build directory, compiling any java classes, packaging those compiled classes into a .jar archive, and archiving in .zip format the complete plugin.

```
jdk.home.1.7=/Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk
iiq.home=/usr/local/apache-tomcat-8.0.30/webapps/identityiq/
pluginName=TodoPlugin
version=2.0.0
```

Plugin Database Scripts

Plugins that require persistence of data outside of that allowed by the IdentityIQ object model require at minimum the creation, updating, and deletion of unique tablespace. The plugin framework creates a database named identityiqPlugin. The creation of this database is handled by the installation scripts packaged with every download of IdentityIQ, in the `WEB-INF/database` folder. Additionally, a default user identityiqPlugin is created to perform operations, installation and deletion of plugins, on this new database. Similar to the base IdentityIQ username and password, these can be modified and updated in the `IdentityIQ iiq.properties` file located in `WEB-INF/classes/iiq.properties`.

When creating a plugin, you must create a folder named db in your project directory. This folder should be further subdivided into three operation specific folders: install, uninstall, and upgrade.

The scripts placed in these folders are automatically run when a plugin is installed or deleted. It is recommended that you include scripts for the four major database types supported by IdentityIQ, MySQL, SQLServer, DB2, and Oracle, or note in your documentation which databases are supported. Database specific scripts must include the database type as the file extension, for example `.mysql`. The `upgrade` folder should contain any deltas in table definitions from prior versions of the plugin.

Plugin User Interface Elements

Most plugins have some additional user interface component that appears in IdentityIQ. Images, CSS files, HTML templates, and JavaScript can all be used to provide the interactions and views required by the plugin. Plugins that use a fullPage element look for a file called `page.xhtml` in the build.

To extend the classes loaded with your plugin to the rest of IdentityIQ, you must specifically declare those classes in the manifest file.

Plugin Authorization

To prevent unauthorized access to your new endpoints, each should be guarded with an authorization mechanism. You can constrain which users can see and access the user interface components, and you can secure the REST endpoints you build into your plugin.

When you define snippets, including widget plugin components, in the manifest file for a plugin, you can specify a rightRequired attribute to constrain access. This attribute names a SailPoint SPRight which users must be assigned for the component to appear in their IdentityIQ instance.

You can also specify a rightRequired at the Plugin object level, in the manifest file, which will specify the required SPRight for a user to be able to access the full-page component of the plugin.

If you leave these rightRequired attributes off, all IdentityIQ users will be able to access those plugin components.

SPRights are the most granular permission object in IdentityIQ. In most cases, users are assigned SPRights in IdentityIQ by attaching those rights to one or more Capability objects and then granting the Capability to the appropriate users.

If the plugin contains a full-page component that users can access through a quicklink, the Quicklink access will be governed by Quicklink Populations, like any other IdentityIQ Quicklink.

User must also be authorized to the full page itself, with the rightRequired specified in the plugin manifest, to be able to view the page.

Other authorization points of note. First, whether or not you explicitly authorize system administrators to these components, they will have full visibility and access to them. Second, when you include a widget in your plugin, the widget will appear in the list of available widgets for all users when they are editing their home page and deciding which content to include there. However, if they are not authorized to the widget's snippet, and they add that widget to their Home page, IdentityIQ will add an empty widget and they will neither be able to see nor interact with any of its functional elements.

To secure the endpoints the plugin framework use Annotations. In Java, an annotation is a syntactic metadata that is added, often before a method signature, that describes the parameters used in that method.

An annotation should have at least three parts

- The HTTP method (GET, POST, PUT, DELETE, etc)
- The path or endpoint - this can be parameterized which is useful for pulling back a single record. The above example uses parameterization by adding the variable within {} tags to the end of the URL, and also declaring the @PathParam appName in the input arguments of the method signature
- The authorization of the method - the allowed values are:
  - **@AllowAll** - this allows anyone to interrogate the endpoint
  - **@RequiredRight("<SPRight>")** - allows users who possess the named SPRight to access the endpoint
  - **@SystemAdmin** - system administrator access only
  - **@Deferred** - Authorization is deferred to the method. When this option is selected, you must also create an Authorizer class that implements the sailpoint.authorization.Authorizer interface. The Authorizer class should overwrite the authorize(UserContext) method of the base Authorizer interface. Inside of the REST resource method, the author would then call authorize().

Plugin XML Artifacts

Any IdentityIQ objects that are required as part of a plugin need to be represented in XML artifacts. This could be something as small as a single new SPRight object or a complex workflow or rule. The mechanism that is used for importing these artifacts during installation is the same as any IdentityIQ object import, so the normal import actions are also available, merge, include, execute, logConfig.

Development of these XML artifacts can be done directly in the build folder, or in the IdentityIQ user interface and either exported using the console or copy and pasted from debug into the build.

When developing in the user interface and then migrating to your build folder using cut and paste, you must remove the id attribute assigned by Hibernate and any other hibernate ID value references. For this reason, it is preferable to export the artifacts using the IdentityIQ console command **./iiq export -clean**.

Everything in the `import` folder is imported The objects can be separated into individual files, or combined into a single file. When a plugin is uninstalled, the XML artifacts that were imported remain in the IdentityIQ database, but the `.zip` archive from which the plugin files was loaded, is removed from the spt_file_bucket and spt_persisted_file tables.

Plugin Java Classes

Plugins are a powerful productivity-enabler, that give users the ability to extend both the IdentityIQ user interface and server in a well-defined manner.

Plugin Java Classes - REST Classes

The plugin framework relies heavily on REST web services integration for the majority of CRUD (create, read, update, and delete) operations. To create a custom REST Resource:

- Extend the BasePluginResource class
- Secure the new endpoints

Extend BasePluginResource

The BasePluginResource class should be used as the base class for all resources. It provides access to utility methods for accessing plugin settings, getting database connections, and more.

- **getConnection** - gets connection to the datasource specified in the `iiq.properties` file for the plugins
- **getPluginName** - this method should be overwritten to return the correct name of the plugin
- **getSettingBool** - gets value of boolean plugin setting for plugin name returned by getPluginName()
- **getSettingInt** - gets value of int plugin setting for plugin name returned by getPluginName()
- **getSettingString** - gets value of String plugin setting for plugin name returned by getPluginName()
- **prepareStatement** - convenient security method for getting Java PreparedStatement object for any database queries that are required
  - signature is prepareStatement(Connection, String, Object...) where the String would be the SQL statement you want to execute
  - Object... would be a list of the parameters values, if any, to be used
- **authorize** - should be overwritten by implementers, but by default only ensures that SystemAdministrator can see everything

Introduce additional methods to handle the various endpoints required by the plugin.

Plugin Java Classes - Plugin Executors

The plugin framework enables you to include custom task implementations or services with your plugin. These items rely on executor classes that contain the business logic for these services. The following executors are currently available:

- Service Executors
- Task Executors
- Policy Executors

You must specifically declare the classes to be exported for each of the executors. Only classes specifically declared are accessible from the rest of IdentityIQ. If a class is not declared, it will fail to instantiate when you open the plugin. Classes are declared in the `manifest.xml` file.

Use the following attributes to declare classes:

- serviceExecutor
- taskExecutor
- policyExecutor

For example:

```
<entry key="taskExecutors">
  <value>
    <List>
      <String>com.acme.TaskExecutor1</String>
      <String>com.acme.TaskExecutor2</String>
    </List>
  </value>
</entry>
```

Plugin Object Properties

When defining your plugin object you must provide the list of service executors that are included. The list lives inside an attributes map under the key serviceExecutors.

- Plugin Helper methods
- All inherited Service methods
- BasePluginTaskExecutor
- Plugin Helper methods
- All inherited TaskExecutor methods
- BasePluginPolicyExecutor
- Plugin Helper methods
- All inherited PolicyExecutor methods.

Plugin Helper Methods

The list of methods that are included with the BasePlugin classes are as follows:

- getPluginName() - returns a string value of the name of the plugin
- getConnection() - returns a Connection object used to query the database
- getSettingString(String settingName) - returns a String setting value from the Plugin Settings
- getSettingBool(String settingName) - returns a boolean value from the Plugin Settings
- getSettingInt(String settingName) - returns a integer value from the Plugin Settings

You can think of the BasePlugin classes as foundation for creating your specific objects. By using them you gain access to the Plugin Helper Methods, but you are not required to use the BasePlugin classes. You can extend directly from the parent class object.

Implementing a Plugin Service Definition

To implement a Plugin Service there are two parts. The service class, containing the business logic that you want the service to actually do, and the service definition xml, that is loaded into IdentityIQ.

- BasePluginService Class — an abstract class that extends the Service class, as well as implements the PluginContext interface. You can use this class as the foundation for your custom Plugin Service
- Service Definition — specify a pluginName attribute. This tells IdentityIQ to use the plugin class loader for this executor. If this attribute is not specified the executor class will not be found

Implementing a Plugin Task Executor

To implement a Plugin Task Executor there are have two parts. The Task Executor class, which handles the business logic for your task, and the TaskDefinition xml object, which gets loaded into IdentityIQ.

- BasePluginTaskExecutor Class — an abstract class that extends the AbstractTaskExecutor class, as well as implements the PluginContext interface. You can use this class as the foundation for your custom Plugin Executor task
- TaskDefinition — include the pluginName attribute, as this attribute tells IdentityIQ to use the plugin class loader instead of default class loader. If the attribute is not specified the executor class will not be found

Implementing a Policy Executor

To implement a Policy Executor there are have two parts. The Policy Executor class, which handles the business logic for your policy, and the Policy xml object, which gets loaded into IdentityIQ.

- BasePluginPolicyExecutor Class — an abstract class that extends the AbstractPolicyExecutor class, as well as implements the PluginContext interface. You can use this class as the foundation for your custom Plugin Policy Executor
- Policy XML — include the pluginName attribute, as this attribute tells IdentityIQ to use the plugin class loader instead of default class loader. If the attribute is not specified the executor class will not be found.

Plugin Java Classes - Script Classes

Note: Beanshell executions are referred to as scripting in this document.

The classes installed for plugins can be made available to all beanshell executions. Beanshell (rules, scriptlets, workflows) scripting invocations are able to use all Java classes, from all plugins, that are declared as exported for scripting. The scripts should fail to load classes which are in plugins, but which are not explicitly exported.

Beanshell executions include:

- rules
- workflow steps (rules and scripts)
- scriptlets

Use the scriptPackages attribute to declare the Java packages exported for use by scripts as follows:

```
<entry key="scriptPackages">
  <value>
    <List>
      <String>com.acme.classy.util</String>
    </List>
  </value>
</entry>
```

Example: Using Plugin Classes From a Rule

This is a simple example of how to call the plugin classes from a rule.

The example Java class named `com.acme.classy.util.ClassyUtil` can be used from a rule if declared properly in the `manifest.xml` using the `sriptPackages` entry shown above.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Rule PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<Rule language="beanshell" name="ClassyRule" >
  <Description>Returns the current timestamp by calling class from
                      ClassyPlugin</Description>
```

```
  <Signature returnType="string"/>
  <Source>
import com.acme.classy.util.ClassyUtil;
long now = ClassyUtil.now();
return "The current timestamp is: " + now;
 </Source>
</Rule>
```

SailPoint Angular Components

To implement the SailPoint-styled angular components, your project needs to include the SailPointBundleLibrary JavaScript file. There are specific directive dependencies on this library when you use the SailPoint-styled components.

Widgets and snippets donot require this library in the plugin project, because for those, the plugin architecture automatically references this library from your IdentityIQ instance. But if you are going to use these in a full page plugin implementation, your plugin needs to include a copy of this JavaScript library, copied from your IdentityIQ version. Plus, your `page.xhtml` component needs to explicitly reference the JavaScript file in a script element so IdentityIQ can resolve the SailPoint Angular directives you are be using. This library needs to match the version of IdentityIQ where the plugin will be deployed.

Your angular module definition needs to specify any other modules that your module has dependencies on. This list will vary, depending on the elements you include. For example, if you are using modal boxes, you need to include `sailpoint.modal`, tree components rely on `sailpoint.tree`, and so on. If your user interface only contains some of the more basic of the field types, you need fewer modules in your dependency list.

An example plugin that demonstrates the different components that are part of the SailPoint Angular bundle, called the Kitchen Sink plugin is installed with IdentityIQ. The plugin is available for download from Compass. Once the plugin is downloaded, you can install and view the plugin through IdentityIQ to see how the different field types behave. Then you can examine the HTML and JavaScript files to see the directives involved and how they are populated.

The library contains a wide array of field types, just like you might see throughout core product pages. For example, selection lists, plain text entry fields or fields where you can enter multiple separate text values as a list, dropdown list boxes where you can select from a set of choices, and a special directive for when the choices should be Boolean true-false values, date pickers, checkboxes, and radio buttons.

The library offers SailPoint styled buttons, and the demo plugin shows how you can use them to attach logic to do things like open modal boxes of various types, post an alert notification to the page or navigate the user to another page in IdentityIQ while preserving navigation history.

Beyond the simpler field types and buttons, there are a few components that create more complex elements. For example, a tree directive for showing nested relationships in data, a directive for displaying a list of cards that is configured for paging when result sets are large, and a directive for a data table that matches the tables in the Angular pages of IdentityIQ.

Internationalization

Message catalog files are specified per language with the two character abbreviation for the corresponding language, or the four-character options when you have locale-specific message catalogs, for example, `TrainingPlugin_fr_ca.properties` for Canadian French. These files should be recorded in the messages folder of your plugin project.

To guard against collisions with IdentityIQ base product message key names, or message keys from other plugins, the best practice is to name your plugin's message keys with a prefix that makes them unique to your plugin. For example, consider using your plugin's name as a prefix.

Both the user interface and server side code can access the provided catalogs.

In the user interface, full page plugins and widget plugins use two different mechanisms, because of differences in library support that are included in the pages.

In a Full page plugin component, the HTML uses the `msgs` function to translate the text.

In Widgets or other snippets that display text, the `spTranslate` function in the SailPoint Bundle Library does the translation, by piping the message key through it.

It is possible to use this syntax in a full page plugin, but only if you have included the SailPoint Angular Bundle Library and have declared a dependency for your angular module on the `sailpoint.i18n module`. The `spTranslate` function is part of that module.

In server-side code, localization is done with the Message object's localize method, passing it the message catalog key to look up and translate.

Plugin Installation and Removal

To install a plugin, click the gear icon and select **Plugins** to navigate to the Installed Plugins page. Click **New** and drag and drop or upload the plugin `.zip` file.

If you downloaded a plugin, the `.zip` file should be included with the download. If you developed the plugin yourself, the `.zip` file is in your project directory under `build/`*`your plugin name`*`/dist`.

When a plugin is installed, the database scripts from the `db/install` folder run, which creates any tables necessary for the plugin, the XML configuration files are imported into the IdentityIQ database from the `import/install` folder, any compiled classes are loaded into the unique plugin classloader, and the manifest file is imported creating the Plugin object.

Remove a plugin by clicking **X** on the appropriate Plugin card on the Installed Plugins page. Database scripts in charge of cleaning up data run from the `db/uninstall` folder and the manifest file (the Plugin object) is removed.

Addition steps might need to be taken to edit the System Configuration file to remove objects created by the plugin, such as quicklinks, or to remove tables created in the plugin database.

Chapter 6: Define Home Page Quicklinks

Quicklinks are objects in IdentityIQ that enable you to place customized links on the IdentityIQ Home page and in the Quicklinks menu available on every page. Quicklinks are defined when IdentityIQ is deployed and are based on the needs of your enterprise. You can determine the behavior and availability of these links for different users. For example, IdentityIQ can be set up to limit access based on the user capabilities, rights, or workgroup membership.

Three objects control links. **QuickLink** objects define the links, the DynamicScope object controls who can view those links, and the **QuickLink**Option object references the first two to create the Quicklinks within the product.

Managing Quicklinks

QuickLinkOption

The **QuickLinkOption** object is created when a quicklink population, or DynamicScope object, is created on the Quicklink Populations page and associated with one or more **QuickLink** objects. The **QuickLinkObjects** objects do not control quicklink populations, nor the targets of the **QuickLink** objects, they are containers holding references to both.

```
<QuickLinkOptions allowSelf="true" created="1443970828183"
id="2c90900950335ce70150335e4797010e">

  <DynamicScopeRef>

   <Reference class="iiq.object.DynamicScope" id="2c90900950335ce70150335e4783010c"
name="Everyone"/>

  </DynamicScopeRef>

  <QuickLinkRef>

    <Reference class="iiq.object.QuickLink" id="2c90900950335ce70150335e478a010d"
name="Access Reviews"/>

  </QuickLinkRef>

</QuickLinkOptions>
```

DynamicScope

The DynamicScope object define groups of users, quicklink populations, based on capability, rights, indirect capabilities and rights granted by a workgroup, population, or any attribute of the identity. These objects are defined on the Quicklinks Populations page. Refer to the *SailPoint IdentityIQ System Administration Guide* for more information.

DynamicScope objects are referenced by name or ID in a **QuickLinkOption** object. If the quicklink population applies to an identity, the Quicklink is visible to that identity. Only System Administrators can view Quicklinks with no scopes.

Note: DynamicScope objects are used to define the population of people who can view and run the Quicklink. DynamicScope objects are not the group of identities or objects that the Quicklink interacts with after the link is clicked.

Examples

The product ships with a DynamicScope that represents the **allowAll** option. The name of the DynamicScope is named **Everyone**. You can associate this option with any Quicklinks you want to enable the entire user population to view or use. The following **QuickLinkOptions** reference this DynamicScope by default:

- Access Reviews
- Approvals
- Signoffs
- Work Items
- Policy Violations

```
<DynamicScope allowAll="true" created="1443970828163"
id="2c90900950335ce70150335e4783010c" name="Everyone"/>
```

The following XML example of a DynamicScope restricts visibility to a specified Quicklink. Visibility is enabled for users in the IT department or who have the Help Desk Personnel capability. Visibility is also enabled for identities in the Inclusion list. Because Barbara.Wilson is in the Inclusions list, she can always see the Quicklink regardless of her capabilities or department.

```
<DynamicScope created="1443973952475" id="2c90900950335ed60150338df3db000a"
name="MyDynamicScope">
  <Description></Description>
  <Inclusions>
    <Reference class="iiq.object.Identity" id="2c90900950336e720150336f0797010d"
name="Barbara.Wilson"/>
  </Inclusions>
  <PopulationRequestAuthority allowAll="true"/>
  <Selector>
    <IdentitySelector>
      <MatchExpression>
        <MatchTerm name="capabilities" value="Help Desk Personel"/>
        <MatchTerm name="Department" value="IT"/>
      </MatchExpression>
    </IdentitySelector>
  </Selector>
</DynamicScope>
```

By default, IdentityIQ assumes that any link defined as a top-level **QuickLink** object is for a non-Lifecycle Manager action which does not operate on a target identity, so no user selection options are presented.

Chapter 7: IdentityIQ Email Templates

Many events in IdentityIQ generate email notifications to notify users of actions required by them or actions taken that directly affects them. These email messages are created based on email templates. Basic templates are provided with the product to construct messages corresponding to each of the email-generating events, and these messages can be customized to meet your specific needs.

To customize any of these templates, copy and rename the template using a unique name.

1. Copy and rename the template using a unique name.
2. Associate the customized template to the email-generating event through the IdentityIQ user interface or configuration XML. See “Importing Email Templates into IdentityIQ” on page 109.

Note: The default email templates should not be modified directly because they might be overwritten during the IdentityIQ release upgrade process.

Accessing the Templates

The default email templates that ship with the product are located in the following area:

- Directory — iiq_*installation_directory*/WEB-INF/config directory
  where iiq_*installation_directory* is the location where you expanded the IdentityIQ installation media
- Files — `emailtemplates.xml` and `lcmemailtemplates.xml`

A third file, named `emailtemplatesSample.xml`, contains additional example templates that are not loaded into IdentityIQ during the initial load process. These templates describe how to create HTML email messages. Any of the templates in these files can be cloned to create custom templates.

After the email templates are loaded into IdentityIQ, either during the initial load or using the console or user interface import option, the templates are stored as XML objects. The templates can be viewed or modified through the IdentityIQ Debug pages. The default templates should not be modified from here because the template are overwritten during any IdentityIQ version update. However, customized templates can be edited directly through the Debug pages, if the organization's source code control procedures allows.

To view the list of email templates from the Debug pages, select **EmailTemplate** from the object list and click **List**. Select the desired template to view or edit its XML representation.

Importing Email Templates into IdentityIQ

When email templates are edited outside of IdentityIQ, they must be imported into the system before they can be used for any notifications. Email templates can be imported through the IdentityIQ user interface or console.

To import a template through the user interface, navigate to the **Gear** icon **-> Global Settings ->** IdentityIQ **Configuration -> Import from File**. Click **Browse** to select the email template's XML file from the file system and click **Import**. IdentityIQ parses the XML during the import process and recognizes the file's contents as an EmailTemplate object. The import fails if the XML is invalid.

To import the email template through the IdentityIQ console:

1. Navigate to the IdentityIQ *Installation Directory*/WEB-INF/bin directory.
2. Start the console and use the console import command to import the file.
   The import is successful only if the XML is valid. Any errors encountered are reported to the console.

   Note: Import files can contain one or more EmailTemplate objects. However, if a file contains more than one object, the import methods expect the set of objects to be wrapped in a `<sailpoint></sailpoint>` **block.**

Associating Templates with Events

Email templates are associated to their respective email-generating events in several places in the IdentityIQ user interface and configuration XML. The table below shows the notification type, the default template name, and their association location. To use a custom template for any of these notifications, specify the custom template name in place of the default template in that notification configuration.

Note: Different email templates accept and use different arguments. The selection lists in the user interface for each notification lists all email templates, no matter what arguments they require. Because IdentityIQ provides a fixed set of arguments for each notification type, only templates whose arguments list matches the provided arguments work correctly to create a useful event notification. Refer to the default template XML to see the arguments (names and variable types) for each notification, and ensure that the selected template's argument list matches the default template's arguments.

Table 9—Email Template Associations

Notification Type (Field Label or Configuration Key)	Default Email Template	Configuration Location
For reminder notices	Work Item Reminder	**Gear icon -> Global Settings ->** IdentityIQ **Configuration -> Mail Settings -> Email Templates section**
For escalation notices	Work Item Escalation	
For work item comment notices	Work Item Comment	
For work item forwarding notices	Work Item Forward	
For policy violation notices	Policy Violation	
For task and report signoff notices	Task Result Signoff	
For work item assignment notices	Work Item Assignment	
For work item assignment removal notices	Work Item Assignment Removal	
For remediation item assignment notices	Remediation Item Assignment	
For remediation item assignment removal notices	Remediation Item Assignment Removal	

Table 9—Email Template Associations

Notification Type (Field Label or Configuration Key)	Default Email Template	Configuration Location
For task status email notice	Task status	**Gear icon -> Global Settings ->** IdentityIQ **Configuration -> Mail Settings -> Email Templates section**
Initial Notification Email Template	Certification	**Gear icon -> Global Settings ->** IdentityIQ **Configuration -> Mail Settings -> Email Templates section** **NOTE: When the challenge period is enabled, Challenge-related notification email templates can be overwritten for individual certifications on the Lifecycle page in each certification configuration.** **NOTE: Initial Notification and Bulk Reassignment notices can be overwritten for each certification on the Notifications page in each certification configuration.**
Exceptions Expiration Notices	Mitigation Expiration	
Bulk Reassignment Modification notices	Bulk Reassignment	
Challenge Period Start Notices to Challengers	Challenge Period Start	
Challenge Period End Notices to Certifiers	Challenge Period End	
Challenge Creation Notices to Challengers	Challenge Creation Notification	
Challenged Decision Notices to Certifiers	Certification Decision Challenged Notification	
Challenge Expiration Notices to Challengers	Challenge Expiration	
Challenge Decision Expiration Notices to Challengers and Certifiers	Challenge Decision Expiration	
Challenge Accepted Notices to Challengers	Challenge Accepted	
Challenge Rejected Notices to Challengers	Challenge Rejected	
Sign-off Approval Notices to Approvers	Certification Sign-off Approval	
Reminder Email Template	Work Item Reminder	This is set in the **Notifications** step in each certification configuration when the associated reminder or escalation option is enabled.
Escalation Email Template	Work Item Escalation	
(Revocation) Reminder Email Template	Work Item Reminder	
(Revocation) Escalation Email Template	Work Item Escalation	
Report signoff initial notification	Task Result Signoff	Configured in report specification i**Require Signoff** is selected when the report is defined.
Report signoff reminder notice	No default in UI but argument list matches Work Item Reminder	
Report signoff escalation notice	No default in UI but argument list matches Work Item Escalation	

Table 9—Email Template Associations

Notification Type (Field Label or Configuration Key)	Default Email Template	Configuration Location
Send PDF of report to someone	Default Report Template	Not specified in the user interface. When a report is defined, its XML can be edited to add an emailTemplateId argument to change the email message template. **NOTE: Because this is cumbersome, organizations might choose to edit the Default Report Template directly rather than cloning it. This customization must be reapplied after any** IdentityIQ **version upgrade, as it is overwritten during the upgrade process.**

Table 9—Email Template Associations

Notification Type (Field Label or Configuration Key)	Default Email Template	Configuration Location
Various: Process Variables Step Arguments Approval Work Item Configuration Email Notification Template Workflows (including sub-processes) using these templates: Identity Correlation Do Manual Actions Do Provisioning Forms Assimilate Provisioning Form Provisioning Approval Subprocess Identity Request Notify Identity Request Provision LCM Create and Update LCM Manage Passwords LCM Provisioning Lifecycle Event - Leaver Lifecycle Event Reinstate Role Modeler - Impact Analysis Role Modeler - Owner Approval	LCM Requester Notification LCM Manager Notification LCM User Notification LCM Identity Update Approval LCM Pending Manual Changes LCM Password Change Notification Pending Manual Changes Account Selection Notification Provisioning Form Notification Role Modeler - Approval Role Modeler - Impact Analysis Review	Specified within workflow definitions in **Setup -> Business Processes** as a process variable, step argument, or work item configuration email notification template. **NOTE: For templates referenced from a workflow, all workflow variables, and all steps arguments and approval arguments defined for the step that invokes the email template are also available for use in the email message (subject, body, cc, etc.)**

Table 9—Email Template Associations

Notification Type (Field Label or Configuration Key)	Default Email Template	Configuration Location
key=delegationEmailTemplate	Delegation	Can not be configured through the user interface. Can only be edited through System Config XML From IdentityIQ Debug Pages, click the **Gear icon -> Global Settings ->** IdentityIQ **Configuration** and search the XML for these email template names or key values
key=delegationRevocationEmailTemplate	Delegation Revocation	
key=delegationFinishedEmailTemplate	Delegation Finished	
key=remediationEmailTemplate	Remediation Work Item	
key=remediationNotificationEmailTemplate	Remediation Notification Sent when Notify Users of Revocations is selected in a certification configuration.	
key= certificationReminderEmailTemplate	Certification Reminder Sent on demand from certification.	
key= policyViolationDelegationEmailTemplate	Policy Violation Delegation	
key= AccountGroupPermissions. challengeGenerationEmailTemplate	Account Group Challenge Creation Notification Specialized form of the Challenge Creation Notification email	
key= accessRequestReminderEmailTemplate	Access Request Reminder Sent on demand from the Access Requests page	
key=openCertsEmailTemplate	Open Certifications	

Email Template XML

The Email Template XML consists of an <EmailTemplate> element with a set of attributes and nested elements that specify the basic components of an email message, such as sender, subject, message body, etc.

The next two sections describe those attributes and nested elements.

EmailTemplate Attributes

The following table lists the components that are generally expressed as attributes on the Email Template.

Example:

```
<EmailTemplate name="Work Item Reminder" cc="$identity.Manager.email" >
<From>administrator@XYZCorp.com</From>
<Body>
```

. . . .

Table 10—Email Template Attributes

EmailTemplate Attribute	Purpose
name	Short but descriptive name for template that uniquely identifies email template
cc, bcc	Carbon Copy and Blind Carbon Copy recipients for the email **NOTE: The to attribute is not specified in the email template because it is determined programmatically as the email is sent and would be overridden**
from	Sender email address. If this entry is not specified in template, the default sender specified on the **Global Settings ->** IdentityIQ **Configuration -> Mail Settings -> Default From Address** is used.

EmailTemplate Nested Elements

The components listed in the following table are generally expressed as nested elements due to their complexity and length.

Table 11—Email Template Nested Attributes

Nested Element	Purpose
<subject>	Subject line for the email message
<body>	Body, or main content, of the email message
<signature>	Hashmap of arguments to the email template The signature for each template cannot be changed through the XML. Arguments to each template vary based on the associated system activity to which they apply. Properties and methods belonging to any object passed as an argument are available to include in the message, but other objects that are not part of the template signature cannot be retrieved to use in the email message.
<Inputs>	Nested element within Signature, signifying the input arguments to the template
<Argument>	Nested element within Signature and Inputs. This element names and specifies the type of each input argument to the template
<Description>	Indicates descriptive information for the reader of the XML. Describes the element in which it is nested For example: <Description> within <Argument> describes the argument usage. <Description> within the <EmailTemplate> describes the purpose and usage of the template)

At the most basic level, the contents of these elements and attributes can be written as straight text values with no variable substitutions. However, the real flexibility and usefulness of these templates is found when custom text is substituted into the message body, subject, and other attributes. This substitution is managed by the Apache Velocity Engine.

Apache Velocity Engine

IdentityIQ email templates are processed through an open-source engine called Apache Velocity. Velocity is a Java-based template engine that allows web page designers to reference methods defined in Java code. IdentityIQ email templates make use of the Velocity Template Language to dynamically specify the email messages' contents and generate custom email messages specific to the recipient, work item, and action involved.

The Velocity Template Language (VTL) is a fairly simple to use. Highlights are included below, and full documentation on the syntax is available in the Apache Velocity User Guide or Reference Guide.

References

As IdentityIQ prepares to send an email notification, the appropriate email template is loaded and its argument variables are passed into the VelocityContext where they can be accessed through VTL reference syntax. The contents of different variable types can be accessed through the syntax described in the table below.

Table 12—Velocity Reference Context Syntax

Reference Type	Examples	Additional Information
Variables	$identityName${identityName}$!identityName	These three syntaxes are generally interchangeable in VTL. Shorthand notation (the first example) is the most commonly used, but each of the other two is required in special cases. Refer to the Velocity User Guide for more information.
Hash table values	$customer.Address	Returns the value corresponding to the Address key in a customer hash table
Object properties	$identity.DisplayName	Invokes the getDisplayName() method on the identity object **NOTE: Property notation resolves to the getter method corresponding to the property name, not to an instance variable.Nested object properties can also be retrieved with this notation.** **Example:** **$item.Certification.Name invokes the getName() method on the Certification object retrieved through the getCertification() method on the item object**
Object methods	$identity.getBundles($application)$identity.hasRole($role,'true')	Used for all non-getter methods and for any methods that require arguments

Directives (Commands)

These are the key commands of the Velocity Template Language that are most frequently used in IdentityIQ email templates to dynamically determine the text that is printed in each email message.

Table 13—Velocity Template Language Directives

Command/Directive	Usage/Purpose	Example
#If… #elseif… #else… #end	Conditional evaluation	#if($requester) requested by $requester.displayableName. #{end}
#foreach… #end	Loop through a list of objects	#foreach ($attrReq in $acctReq.attributeRequests) Operation: $attrReq.operation Attribute:$attrReq.name Value(s): $attrReq.value#end
#set	Establish the value of a reference	#set ($identityName = “John.Smith”)#set ($book.Title = “War and Peace”)

Refer to the Velocity User Guide for additional information on the language, including the syntax for less commonly used directives.

VTL vs. $(variableName) Notation

The VTL reference syntax must not be confused with the $(variableName) notation used for variable referencing in other IdentityIQ XML objects, such as Workflows. Velocity does not recognize this syntax and is unable to parse text that uses it. When IdentityIQ detects this syntax in any element of an email template, that portion of the message is not passed to Velocity for rendering at all. Instead, its contents are rendered by a simpler mechanism that is capable of doing the variable substitution based on the template's arguments. However, none of the Velocity directives are interpreted. Any Velocity commands included in the same element with a variable that uses the $(variableName) notation is treated as normal text and printed as-is in the final message.

Incorporating VTL in Email Template XML

All input arguments in the template signature are automatically loaded into the VelocityContext and are therefore accessible through the VTL reference notation for inclusion in the message text. Additionally, Velocity commands (conditional statements, loops, etc.) can be used in determining the text to print in the messages. Excerpts from the default email templates in IdentityIQ are used as examples throughout the rest of this section to illustrate how reference variables and various command syntaxes can be used.

Where to Use VTL

The Velocity Template Language syntax can be specified in any attribute or element that is used to build the email message. Most commonly, this means the <Subject> and <Body> elements of the message, but the cc and bcc recipients (as well as the from email address) are often dynamically specified through reference variables as well.

Reference Variables

When a variable name is referenced within the text for any of the message elements, its value is substituted into the text in its place.

Example:

```
<Body>$certifierName has accepted the challenge for '$challengeItem' and will change
the decision.
</Body>
```

Variable substitution results in the email message content:

John Smith has accepted the challenge for 'Entitlements on Financials' and will change the decision. Velocity can also access data values in fields within objects passed as arguments and replace the variable notation with those values. Consider the <subject> and <body> elements shown below. The argument list for this email template includes a Certification object (named certification) and an Identity object (named certifier).

```
<Subject>$certification.name requires approval</Subject>
<Body>$certification.name was signed by $certifier.displayableName and requires your
approval.
Login and view your work item inbox to complete this request.
</Body>
```

To resolve these variable references, Velocity calls the getName() method on the certification object and the getDisplayableName() method on the certifier's identity object. When the substitutions are made, the final email message looks like this:

> **Subject**: Manager Access Review for Catherine Simmons requires approval
>
> Manager Access Review for Catherine Simmons was signed by Catherine Simmons and requires your approval.
>
> Login and view your work item inbox to complete this request.

Any attribute or method on any of a template's input arguments can be accessed through the reference variables.

Extended attributes on IdentityIQ objects can be accessed through the attributes hash map or by providing the attribute name as an argument to the appropriate getter method. Identity extended attributes, for example, are accessible through the Identity's attributes hash map or through the getAttribute() method on the Identity object (e.g. $certifier.attributes.region and $certifier.getAttribute("region") both return the value in the "region" extended attribute).

Note: The list of available methods for IdentityIQ objects (Identity, Certification, ProvisioningPlan, etc.) can be found in the SailPoint JavaDocs that ship with the IdentityIQ product. These can be viewed through a browser at URL: [IdentityIQ base URL]/doc/javadoc/.

Conditional Statements

Conditional statements can be used to determine whether text should be included in the message or to choose alternate wording based on attribute values.

Whole paragraphs can be included or omitted based on conditional tests.

```
#if ($remindersRemaining > 0)
This work item will escalate after $remindersRemaining more reminder(s).
#end
```

Additionally, parts of a paragraph or sentence can be suppressed or altered based on conditional evaluations. In this example, if $requester is null, the portion of the text “requested by $requester.displayableName, and” is suppressed. Specifying the #if statement in-line with the rest of the text prevents extra line breaks in the middle of the sentence in the resulting email message.

```
<Body>This is your $ordinalNumReminders reminder that the work item $workItemName
#if($requester)requested by $requester.displayableName, and #{end}created on
…
```

Attribute values can also be evaluated to determine which of multiple text selections to include in a message:

```
#if ( $launcher != $identityName )
$launcher requested the following password changes be made to your account(s).
#else
The following password changes were made to your account(s) at your request.
#end
```

Method Calls

Methods within object arguments can be accessed directly through the method reference syntax.

```
#if($expiration)
#if($expiration.getTime() > $nowDate.getTime())
is due on $spTools.formatDate($expiration,3,1).
#{else}
was due on $spTools.formatDate($expiration,3,1).
#{end}
#{else}
was due on $spTools.formatDate($oldDueDate,3,1).
#{end}
```

```
#if ( $item.level )
          Priority: $item.level
#else
          Priority: Normal
#end
```

This block checks to see if $expiration is null. If it is not null, it prints “is due on…” or “was due on…” based on whether the expiration date/time is before or after the current date/time. If $expiration is null, this is an older expired work Item so the message uses the $oldDueDate field as work item due date in the message. It also checks to see if the priority was set in. If the $item.level is null, the priority is set to Normal.

Throughout this example, the printed date/time is formatted with the spTools.formatDate() method. The spTools reference variable is discussed in the next section.

Note: This example was altered from its original format in the default Work Item Reminder email template. In the template, this #if statement was specified in-line to prevent unwanted line breaks in the message. Line breaks have been inserted here for readability and should not be included in the message body unless they are desired in the resulting message text.

SPTools Function Library

Immediately before any template is submitted for evaluation by the Velocity engine, the spTools argument is added to the VelocityContext so the template can access its methods. SpTools is a function library that contains a few localization utility methods to help with message formatting -- primarily date formatting. The methods available within spTools are listed in the table below:

Table 14—spTools Methods

Method	Description
String formatDate(Object date)	Formats the passed-in date object to a string representation using the IIQ default date and time styles (both the java.util.dateformat SHORT formats), formatted per the norms of the server's default locale and timezone
String formatDate(Object date, Integer dateStyle, Integer timeStyle)	Formats the passed-in date object to a string representation using the specified date and time styles, formatted per the norms of the server's default locale and timezone **NOTE: The styles are represented by constant values:** **SHORT = 3** **MEDIUM = 2** **LONG = 1** **FULL = 0** **dateStyle and timeStyle correspond to java.text.DateFormat constants. See the Sun Javadocs for details**
String formatDate(Object date, String formatString)	Formats the date according to the specified formatString (uses the java.text.SimpleDateFormat method)
String getMessage(String key)	Returns an internationalized message from the message catalog corresponding to the provided key
String escapeHtml(String string)	Converts HTML special characters to their entity equivalents Example: escapeHtml(' This is an article ') Returns: <div class="article">This is an article</div>

In the out-of-the-box email templates, the most commonly used method from this library is the formatDate() method that takes a date object and two integers as arguments:

```
$spTools.formatDate($expiration,3,1)
```

After the reference shown above is resolved by Velocity, the date/time value in the expiration argument is printed in the email message in MM/dd/yy hh:mm:ssPM format (or the appropriate equivalent for the server's locale).

CDATA Blocks

When any component of the email message (body, subject, cc, etc.) contains characters that are illegal in XML text (e.g. characters like < and & that are interpreted by the parser as the start of an XML element or character entity, respectively), the entire component must be expressed in a CDATA block to prevent it from being parsed. For example, any message body written as HTML must be contained within a CDATA section.

```
<Body><![CDATA[

<html>

<body style="background:#FFF;margin:0;padding:0;text-align:left;">

<p style="margin:20px 0 0;padding:0;color:#333;font:bold 10pt
Arial;line-height:15pt;">$!{workItem.owner.firstname},</p>

<p style="margin:0 0 20px;padding:0;color:#333;font:normal 10pt
Arial;line-height:15pt;">As part of our periodic compliance efforts, you are
responsible for certifying the access your employees have to enterprise applications.
</p>

<p style="margin:0;padding:0;color:#333;font:normal 10pt Arial;line-height:15pt;">A
specific access certification is named <b>$!{workItemName}</b> has been created for
you, and is due on <strong>$spTools.formatDate(
$certification.expiration,3,3)</strong>. <a
href="http://localhost:8080/iiq/manage/certification/entityList.jsf?certificationId
=$!{certification.id}">Click here to get started on this task.</a></p>

</body>

</html>

    ]]> </Body>
```

The marked up text can then be passed to Velocity for variable substitution and can be rendered as an HTML email message.

Sending an Email from a Rule

Some installations may require notifications to be sent based on events that are not covered by the automated system notifications. Rules can often be used to drive these notifications. The example below shows how to send an email from a rule.

```
<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE sailpoint PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<sailpoint>

<Rule language="beanshell" name="Test Email Sending" type="BuildMap">

  <Description>Debugging Tool - Sends a sample email out via the email
server.</Description>

  <Signature returnType="Map">
```

```
    <Inputs>
      <Argument name="log">
        <Description>
          The log object associated with the SailPointContext.
        </Description>
      </Argument>
      <Argument name="context">
        <Description>
         A sailpoint.api.SailPointContext object that can be used to query the database
if necessary.
        </Description>
      </Argument>
    </Inputs>
   </Signature>
   <Source>
       // Library inclusions for BeanShell
import sailpoint.api.*;
import sailpoint.object.*;
import sailpoint.tools.*;

import java.util.*;
import java.lang.*;
import java.text.*;

// Point this to the "To" email address
String emailDest = "adam.hampton@example.com";

// Specify the email template name in tplName
String tplName = "SailPoint - Test Email Sending";
EmailTemplate template = context.getObjectByName(EmailTemplate.class, tplName);
if (null == template) {
   log.error("ERROR: could not find email template [ " + tplName + "]");
   return;
}
template = (EmailTemplate) template.deepCopy(context);
if (null == template) {
   log.error("ERROR: failed to deepCopy template [ " + tplName + "]");
   return;
```

```
}
Map args = new HashMap();
// Add all args needed by the template like this
args.put("testField1", "This is a test of template parameters.");

EmailOptions ops = new EmailOptions(emailDest, args);
context.sendEmailNotification(template, ops);

return;
 </Source>
 </Rule>
</sailpoint>
```

The beanshell from this example rule can be used as a template for sending an email from any defined rule. Simply change the email template name, recipient, and template arguments to create the desired notification.

Using a Rule to Test Templates and Email Configuration

This example rule can also be used to test the email server configuration or to test any email template. Complete these steps to use the rule for testing purposes:

1. Set up an email server on **Gear icon -> Global Settings ->** IdentityIQ **Configuration -> Mail Settings**.

   Note: To test email templates independently from the email server configuration without actually sending emails through the server, choose Email Notification Type: Redirect to File. This writes the email text to the specified file.

2. Edit an email template to contain the desired message and import it.
3. Edit the “Test Email Sending” rule to include the desired “To” email address, email template, and arguments, and import the rule. (Rules are imported the same way as templates, as described in Importing Email Templates into IdentityIQ.)
4. Run the “Test Email Sending” rule from the IdentityIQ console to send the email.

   ```
   > rule “Test Email Sending”
   ```

5. Examine the resultant email (either in the recipient's inbox or the redirect email file) to verify that the message appears as expected.

Chapter 8: Data Encryption

Data encryptions is done using four basic concepts: the keystore, master password, encrypted data synchronization, and the keystore console.

- KeyStore — the location where the encryption keys used by IdentityIQ are persisted.
- Master Password — the entire keystore can be encrypted with an ascii password. This is the keystore or master password. You can change the keystore password using the keystore console command. Only one master password can exist. When the master password changes the entire keystore and master password file are re-encrypted and rewritten.
- Encryped Data Synchronization — the process of re-encrypting existing data with the newest key in the keystore.
- Keystore Console — the tool (iiq keystore) used to manage the keystore and master password.

The keystore and master password are file based and secured by the file system. They are stored in two separate files. The files can be located in the IdentityIQ deployment directory or placed in an alternative directory during configuration. By default the files are stored in the following location:

```
keystorePassword = WEB-INF/classes/iiq.cfg
keystore = WEB-INF/classes/iiq.dat
```

An alternate keystore file location, password file, or just password in clear text can be specified in the `iiq.properties` file under these keys:

```
keyStore.file
keyStore.passwordFile
```

iiq KeyStore Console Commands

The iiq **keystore** command is the interface to update the keystore and keystore password. A master password can to be entered into the console or generated when it is being updated.

The keyStore console supports the following commands:

Table 1— KeyStore Console Commands

Command	Definition
`use KeyStoreFile masterFile`	Specify the keystore and master file to use when interacting with an alternate keystore. The *keyStoreFile* argument in position 1 specifies the path to the file to be used when creating/updating the keystore. If this argument is not specified the command uses `$SPHOME/WEB-INF/classes/iiq.dat`. The *masterFile* argument in position 2 specifies the path and filename used to store the master file. The **use** command gives you the ability to build the keystores outside your operating running environment and merge them in when scheduled. **Note: If you do not call the use command, the changes are positioned in the configured paths.**
`addKey [ -q ]`	Generate a new encryption key, the key is securely generated and random. `-q` as argument in position 1 generatesa new key without prompting for confirmation. **Note: If no argument is included, you are prompted for confirmation before the key is generated.**
`list`	List the contents of the keystore.
`master [newPassword newPasswordConfirmation]`	**Note: Passwords must be at least 8 characters.** Change the master password and re-encrypt the keystore using the new password. **Note: If no argument is included, you are prompted for confirmation.** If `newPassword` and `newPasswordConfirmation` are in argument position 1 and 2, you are not prompted for confirmation. `-g` is in argument position 1 a new password is generated without confirmation.
`about`	Specifies the two files that being modified.

Encrypted Data Synchronization

The Encrypted Data Synchronization task goes over the objects re-encrypting the values using the newest key.

Note: The Encrypted Data Synchronization task is not enabled upon installation, you must create the task from the New Task drop-down menu.

The task encrypts the following attributes/types by default:

- Application secret configuration attributes
- User passwords
- Password history
- Users challenge questions
- Activity/Target source configurations
- Integration configuration password attributes

In cases such as integration configuration and unstructured target sources the task looks for encrypted values with the password in the name. You can also add a configuration attribute, IIQ`SecretAttributes`, to either type names to define which attributes are targeted during a re-synchronization.

```
<entry key="IIQSecretAttributes">
        <value>
          <List>
            <String>mySecret1</String>
            <String>mySecret2</String>
            <String>password</String>
          </List>
        </value>
      </entry>
```

The task enables you do disable the following three categories of objects:

- Applications — which enabled application, activity and target source updates
- Identity
- Integration configuration

Using IdentityIQ KeyStore

Note: Make sure to store copies of the iiq.dat and iiq.cfg files in a safe place. When you upgrade or reinstall IdentityIQ, the files are readily available to be restored.

Note: Make sure that the file permissions are set to allow access only by the application server that runs IdentityIQ.

In a standard installation of IdentityIQ, passwords are all encrypted using the same encryption secret. Encrypted passwords used in one installation can be reused (decrypted) by any other installation ofIdentityIQ. The keystore feature enables the use of a site specific key. With the keystore feature enabled, a password used on one site cannot be decrypted on another site without having the site specific encryption keys.

Configuration

The keystore is stored in `WEB-INF/classes/iiq.dat` with an accompanying configuration file `WEB-INF/classes/iiq.cfg`.

The `iiq.properties` file provides two options to specify an alternative location for `iiq.dat` and `iiq.cfg`. In the default `iiq.properties`, these options (`keyStore.file` and `keyStore.passwordFile`) are commented out.

```
# IIQ Keystore and Master Password properties
#

# file location of the IIQ keystore
# (override of the default $SPHOME/WEB-INF/classes/iiq.dat )
#
#keyStore.file = /example/path/filename

# file location of the IIQ master password file
# (override of the default $SPHOME/WEB-INF/classes/iiq.cfg )
#
#keyStore.passwordFile = /example/path/filename
```

To put the files in an alternative location, for example `/etc/access` governance suite, enable and change these options as follows.

Note: You may need to modify your application server or Java sandbox security settings to allow access to the key files outside the application server installation directories.

```
# IIQ Keystore and Master Password properties
#

# file location of the IIQ keystore
# (override of the default $SPHOME/WEB-INF/classes/iiq.dat )
#
 keyStore.file = /etc/access governance suite/iiq.dat

# file location of the IIQ master password file
# (override of the default $SPHOME/WEB-INF/classes/iiq.cfg )
#
 keyStore.passwordFile = /etc/access governance suite/iiq.cfg
```

Key Creation

To create or manage the keystore: navigate to the `WEB-INF/bin` folder and start the IdentityIQ KeyStore console with the **keystore** command:

1. Navigate to the `WEB-INF/bin` folder and start the IdentityIQ Keystore console with the keystore command

   ```
   iiq keystore
   ```

2. The console displays a prompt similar to the IdentityIQ console. Use the **help** to list all accepted KeyStore Console commands. For example, use the **addKey** command to create a new key and the **list** command to view the contents of the keystore.

   ```
   > addKey
   Generate a new encryption key (y/n)?
   y
   ```

```
Generating a new encryption key for keystore
[/var/tomcat/webapps/access governance suite/WEB-INF/classes/iiq.dat].
New encrpytion key successfully saved to keystore.
All application servers must be restarted for changes to take effect.
>
```

Note: If the keystore file does not exist, it is created and a new, randomly generated key is added.

3. The **list** command displays the newly created key:

```
> list
Listing contents for keystore
[/var/tomcat/webapps/iiq6/WEB-INF/classes/iiq.dat].
KeyAlias   Algorithm Format       Object

2          AES    RAW          javax.crypto.spec.SecretKeySpec@fffe81cd
>
```

4. Use the **exit** command to leave the console.
5. Restart your application server.
   After you restart the application server, any newly set password is encrypted using the new encryption key. Without the files `iiq.dat` and `iiq.cfg`, passwords cannot be decrypted by IdentityIQ.
   If you run more than one instance of IdentityIQ, you must place the following files in the `WEB-INF/classes` folder of each instance, or in the location specified in `iiq.properties`.

Re-Encrypt Passwords

The new encryption key is used for newly encrypted passwords. However, because existing passwords can also be decrypted using the default method on any system, you must re-encrypt existing passwords. Ro re-encrypt existing password, you must create a new Encrypted Data Synchronization Task in IdentityIQ.

1. From the Navigation menu bar, select **Intelligence -> Tasks**.
2. From the **New Task** drop-down list select **Encrypted Data Synchronization Task** from the drop-down list.
3. Enter a name for the new task.
4. OPTIONAL: If needed, you can exclude types such as applications, identities or integration configurations from processing.
5. **Save and Execute** to immediately run the task.

After the task has completed, all selected encrypted data is changed. A password encrypted with the default key is prefixed with 1. Items encrypted with the new encryption key are prefixed with 2 or another number if multiple encryption keys are stored.

For example, when you look up the Administrator's password in the console, the displays is similar to the following:

```
> search identity password where name admin
2:WpTZ2hmNaInTAJzeK9Swcw==
```

Using the Different Encryption Keys

After a new key is added to the keystore, the key is used as the default encryption key. Everything encrypted inside IdentityIQ then uses the new key. For example:

```
$ ./iiq console
> encrypt test
```

```
2:bt7YJA6iovzF5Uu6RIjueg==
>
```

There is one exception. The command iiq **encrypt**, continues to use the original default encryption key:

```
$ ./iiq encrypt test
1:8zJwAXqvK5/b92JbPXLLKw==
$
```

Although the syntax reported by the bare command does not indicates this, the command accepts an extra parameter to select the encryption key to use. For example:

```
iiq encrypt string [key]
```

Note: The encrypt command in the iiq console does NOT accept this extra parameter.

The *key* is the number that displays in the list command and used as prefix for the keys.

- To select the newly created key, use 2. If multiple keys are in the keystore, use any available higher number.
- To select the original default key use 1 or nothing.

For example:

```
$ ./iiq encrypt test 1
1:8zJwAXqvK5/b92JbPXLLKw==
$ ./iiq encrypt test 2
2:bt7YJA6iovzF5Uu6RIjueg==
```