CA CloudMinder™

Identity Management Administration Guide

1.53

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

CA Technologies Product References

This document references the following CA Technologies products:

- CA CloudMinder™ Identity Management
- CA CloudMinder™ Advanced Authentication
- CA CloudMinder™ Single Sign-On
- CA Directory
- CA IdentityMinder™
- CA AuthMinder™
- CA RiskMinder™
- CA SiteMinder®
- CA SiteMinder® for Secure Proxy Server
- CA Layer 7

Contents

Chapter 3: Admin Tasks 39

Chapter 4: Password Management 65

Chapter 1: Role Planning

To plan your roles, you decide what kind of roles your business or organization needs and how you will delegate the management of users and their application access. Based on these decisions, you determine each role's characteristics.

To use roles effectively, consider these types of questions about user needs and administrator responsibilities:

- Which departments and organizations have users to be managed?
- What additional accounts in managed endpoints will users need?
- Which users should be administrators of other users?
- Who should manage the administrators?
- What admin and access tasks are needed in each role?
- Who should create roles and tasks?
- How can I use roles to delegate work?

  The last question concerns sharing the work of managing users and granting application access. More information about the delegation model exists in Delegated Administration.

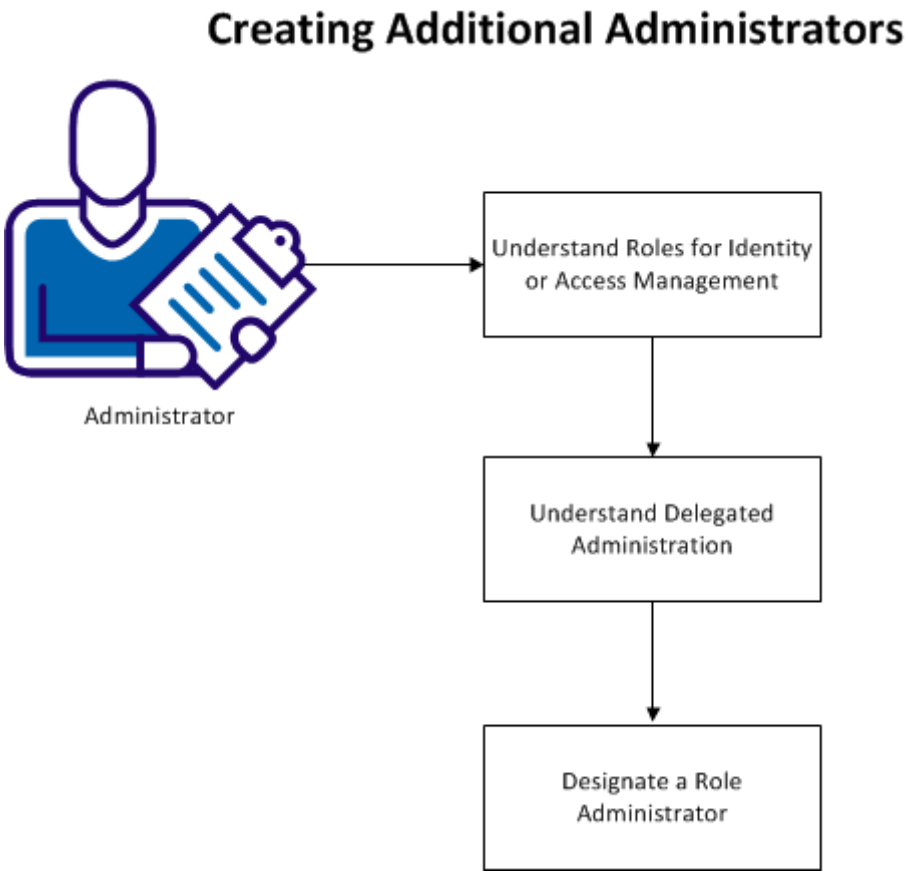
Based on your answers to these questions, you can decide how many and what kind of roles are needed.

This section contains the following topics:

Creating Administrators

You can be solely responsible for granting all roles to users in your system. You can also share the work of granting user roles by designating additional administrators. This approach is named *delegated administration*.

The following diagram shows the information to understand, and the steps to perform, in creating additional administrators.



The following topics explain how to create additional administrators:

- Roles for Identity or Access Management (see page 15)
- Delegated Administration (see page 15)
- Designate a Role Administrator (see page 16)

Roles for Identity or Access Management

To enable management of user identities and their access to other accounts, CA CloudMinder provides the following types of role:

Type of Role	Purpose
Admin role	Contains admin tasks that, when granted that role, a user can perform in CA CloudMinder, such as tasks for changing a user password or group membership. Admin roles can also include any task that appears in the User Console.
Provisioning role	Contains account templates that define accounts that exist in managed endpoints, such as an email system. The account templates also define how user attributes are mapped to these accounts.
Access role	Access roles provide an additional way to provide entitlements in Identity Management or another application. For example, you can use access roles to accomplish the following actions: ■ Provide indirect access to a user attribute. ■ Create complex expressions. ■ Set a profile attribute that another application can use to determine entitlements.

Delegated Administration

Delegated administration is the use of roles to share the work of managing users and granting application access.

For each role in the system, a user can serve one or more of the following functions:

Function	Definition
Role Owner	Modifies the role.
Role Administrator	Assigns the role to users and other role administrators.
Role Member	Uses the role to perform admin or access tasks or use an endpoint account.

By dividing these functions between users, you can share the work of managing a role. For example, you can have lower-level administrators manage role membership and higher-level administrators modify the role.

You can implement delegated administration in the following ways:

- Directly designate a user as an administrator for a given role.
- Configure *admin rules* for a role. Admin rules define which users can be administrators of a role. The system automatically creates additional administrators when users meet the criteria specified in the rules.

  Note: Only an administrator with privileges to modify a role can configure admin rules for that role. Typically, system administrators perform this activity. To configure admin rules that automatically delegate administration for a role, see the section entitled Admin Roles in the Reference Information section of the Online Help.

Designate a Role Administrator

You can designate a user as an administrator of a role. The administrator can then assign the role to other users.

Follow these steps:

1. Log in to the User Console as a user with role management tasks.
2. Select Tasks, Roles and Tasks.
3. Select one of the following tasks:
   - Admin Roles, Modify Admin Role Members/Administrators
   - Provisioning Roles, Modify Provisioning Role Members/Administrators
   - Access Roles, Modify Access Role Members/Administrators

   A search screen appears.
4. Select the role that you intend to assign to the user.
5. Click the Administrators tab.

   A list of current role administrators appears.
6. Click Add a User.

   A search screen appears.
7. Search for the user you want to add as an administrator and click Select.

   An updated list of role administrators appears.
8. Click Submit.

   The user becomes an administrator of the role. This step completes the process of delegating administration of a provisioning role. The administrator can now assign the role to other users, granting access to the associated endpoint accounts.

Delegation Steps

Delegated administration occurs as follows:

1. An administrator creates the role with rules for who is a role owner, administrator, or member.
2. A role owner modifies the role, when changes are needed.
3. A role administrator:
   - Assigns more role administrators (optional).
   - Assigns more role members (optional).

   Some users are already role administrators or members by meeting rules defined in the role.
4. A role member uses the role:
   - An admin role member manages users and other objects in the Identity Management environment.
   - An access role member performs functions in business applications.
   - A provisioning role member uses the accounts defined by policies in the role.

Delegation Example

You can create a role with rules for who can be a member or administrator. You can then assign the role, so that other users (who do not already meet the rules) can become a role member or administrator.

Consider the following example of administrators who manage the business application rights of end users:

- Jeff is a role owner for the Accountant role; so when the role requires changes, Jeff modifies the role.
- David and Lisa are role administrators for that role. They assign regional users as role members.
- Other users are role members without being assigned as role members. Instead, they meet the rule to be role members.

  The role members use the Accountant role to generate purchase orders and perform other tasks in financial applications.

The section Role Characteristics provides details on rules and other characteristics of a role.

Role Characteristics

When you create a role, you define the characteristics shown in the following table:

Characteristics	Definition
Role Profile	General characteristics of the role.
Tasks	Tasks for an admin role.
Account Templates	Templates that define accounts in managed endpoints for a provisioning role.
Member Rules, Member Policies	A member rule defines conditions for a user to be an access or admin role member. A member policy combines a member rule with scope rules. **Note:** Provisioning roles have no member rules and policies. To make a user a member, you use Modify Provisioning Role Members/Administrators.
Admin Rules, Admin Policies	■ An admin rule defines conditions for a user to be a role administrator. ■ An admin policy combines an admin rule with a scope rule and administrator privileges for assigning the role.
Owner Rules	Conditions for a user to be a role owner.
Scope Rules	Limits on which objects can be managed by the role.
Add Actions, Remove Actions	Changes to a user profile when a user is added or removed as a role member or administrator.

Role Profile

The role profile is the name and description of the role and whether or not the role is enabled. If enabled, the role is available for use as soon as it is created.

Tasks for the Role

For an admin role, you can choose one or more admin tasks, including external tasks, from one or more categories.

Account Templates

Each provisioning role contains account templates. They define the accounts that exist in managed endpoints. For example, an endpoint for an Exchange account might define the size of the mailbox. The account templates also define how user attributes are mapped to accounts.

You can choose one or more endpoints for each endpoint type. A user who is assigned the role receives an account in the endpoint.

Member, Admin, and Owner Rules

Each role includes rules about who can be a member, administrator, or owner of that role. Therefore, a user could be a member of one role, several roles, or no roles.

Member, admin, and owner rules use the conditions in the following table:

Rule Condition	Example	Rule Syntax
The user must match one attribute value.	Users where title starts with senior	where <user-filter>
The user must match multiple attribute values.	Users where title=manager and locality=east	where <user-filter>
The user must belong to named organizations.	Users in organization sales and lower	in <org-rule>
The user must belong to organizations that meet a condition specified by attributes on the organization.	Users in organizations where Business Type=gold or platinum	in organizations where <org-filter>
The user must belong to specific organizations and match specific user attributes.	Users where title=manager and locality=east and who are in organization sales or marketing	where <user-filter> and who are in <org-rule>
The user must belong to a specific group.	Users who are members of 401K group	who are members of group [set the product group or family]
The user must be a member of a role.	Users who are members of the Help Desk role	who are members of <role-rule>

Rule Condition	Example	Rule Syntax
The user must be an administrator of a role.	Users who are administrators of the Sales Manager role	who are administrators of <role-rule>
The user must be an owner of a role.	Users who are owners of the User Manager role	who are owners of <role-rule>
The user must belong to a group which meets a condition specified by attributes on the group.	Users who are members of groups where owner=CIO	who are members of group <group-filter>
The user must meet a condition based on an LDAP query.	(Use an LDAP directory for situations where a query created in the Identity Management User Console is insufficient)	user returned by the query ldap_query

Some rules may involve comparing a value to a multi-valued attribute. For the rule to apply, at least one value in a multi-valued attribute must satisfy the rule. For example, if the rule is Attribute A EQUALS 1, and the value of attribute A is 1, 2, 3 for User X, then User X satisfies the criteria.

The user who creates the role may be unable to modify the role. To be able to modify the role, that user must meet the conditions in the owner rules.

Note: In large implementations, it may take significant time to evaluate member, admin, and owner rules. To reduce the evaluation time for rules that include user-attributes, you can enable the in-memory evaluation option. For more information, see the *Configuration Guide*.

Scope Rules

You combine member and admin rules with scope rules. *Scope rules* limit objects on which the role can be used.

- For a role member, scope rules control which objects can be managed with the role.
- For a role administrator, scope rules control which users can become role members and administrators.

Scope applies to the primary object of the task. For example, user is the primary object of the Create User task. However, scope does not apply to the groups for that user, because group is a secondary object.

For most object types, you can specify the types of scope rules in the following table.

Rule Condition	Example	Rule Syntax
All	Role members can manage all objects	All
The object must match one or more attribute values.	Users where **title** starts with **senior**	where <filter>

When you select the filter option, Identity Management displays two types of filters:

<attribute> <comparator><value>

An attribute in the object's profile must match a specific value.

<attribute> <comparator> admin's <user-attribute>

An attribute in the object's profile must match an attribute on the administrator's profile. For example: Users where manager = admin's UserID.

Additional options, which are described in the following tables, are available for user, group, and organization objects.

Note: The following user scope rules are examples. You can create other rules to handle different relationships between the administrator and the users that the administrator can manage.

Rule Condition	Example	Rule Syntax
The user must match one attribute value.	Users where member of group sales or cell phone does not equal null	where <user-filter>
The user must match multiple attribute values.	Users where title=manager and locality=USA	where <user-filter>

Rule Condition	Example	Rule Syntax
The user must belong to named organizations.	Users in organization Australia or New Zealand **Note:** Organization scope rule apply to suborganizations of the organization that meets the rule. For example, if the organization rule is "in Organization1", the scope rule applies to Organization1.1 and Organization1.2, but does not apply to Organization1.	in <org-rule>
The user must belong to organizations that meet a condition specified by attributes on the organization.	Users in organizations where Business Type=gold or platinum	in organizations where <org-filter>
The user must belong to specific organizations and match specific user attributes.	Users where title=manager and locality=east and who are in organization sales or organization marketing	where <user-filter> and who are in <org-rule>
The attribute on a user's profile must match an attribute on the administrator's profile.	Users where manager = admin's UserID	where <user-attribute> <comparator> admin's <user-attribute> **Note:** Do use the Not Equal To comparator with a multi-valued attribute.
The user is in the same organization as the administrator.	Users in the organization where Jeff (the administrator) is a member	admin's organization
The user is in an organization which is listed on the administrator's attribute.	Users in sales or marketing	organization that is a value in admin's <admin-attr>

Note: The following group scope rules are only examples. You can create other rules to handle different relationships between the administrator and the groups that the administrator can manage.

Rule Condition	Example	Rule Syntax
The group must match one attribute value.	Group name where Group name = 401K	where <group-filter>
The groups must belong to named organizations.	Groups in organization accounting and lower	in <org-rule>
The group must match one attribute value and belong to named organizations.	Groups where BusinessType = finance and who are in organization sales and lower	where <group-filter> and who are in <org-rule>
The group must be listed in an attribute of the administrator.	Groups where Description = Engineering	where <group-attribute> <comparator> admin's <user-attribute> **Note:** Do use the Not Equal To comparator with a multi-valued attribute.

Note: The following organization scope rules are only examples. You can create other rules to handle different relationships between the administrator and the organizations that the administrator can manage.

Rule Condition	Example	Rule Syntax
The organization must match one attribute value.	organizations where org Name=finance	where <org-filter>
The organization must belong to named organization.	organizations in finance and lower	in <org-rule>
The organization must match one attribute value and must belong to named organization.	organizations where org Name=finance and are in finance and lower	where <org-filter> and are in <org-filter>

More information:

Common Guidelines about Rules (see page 24)

Common Guidelines about Rules

Whatever type of rule that you create, you should understand how Identity Management processes them.

Evaluation of Operators

In creating rules for a role, you may include >=, <=, <, and > operators. However, these operators are evaluated as strings by the LDAP directory or relational database. Most user stores compare strings based on the alphabet. Therefore, in comparing 500 to 1100, the user store may determine that 500 is greater because 5 is greater than 1.

You may be able to change the way strings are compared in the user store. Consult the documentation for the LDAP directory service or relational database software.

Identity Management processes OR statements before AND statements. Consider the following example:

```
where(company=CA and city=Boston or city=Framingham)
```

In this example, Identity Management processes (Boston or Framingham) first and then performs the logical AND with company=CA.

Case-Insensitivity of Rules

When you create admin or access roles, the rules that you create may be evaluated in a case-insensitive or case-sensitive manner depending on the user store.

However, at the end of a create or modify operation, the rules are evaluated internally in a case-insensitive manner before committing the changes to the user store. For example, if a rule has a condition where title=manager, the rule matches the user store object, whether it has a title value of manager or Manager.

Add and Remove Actions

You must specify an Add and Remove Action for Identity Management to correctly manage a role's membership when an administrator grants or revokes the role.

- The Add Action must make the user meet the criteria in one of the role's member rules. For example, if the member rule for the User Manager role states that role members have "User Manager" as a value of their Admin Roles attribute, the Add Action must add "User Manager" to the Admin Roles attribute.
- The Remove Action should alter the profile of a user so that the user no longer matches the member rule when the rule is revoked.

Each role can have two *add actions* and two *remove actions*.

If administrators can add and remove members of the role, you define add and remove actions. Otherwise, the user has the role by meeting the member rule, such as by belonging to the RoleAdmins group. For example:

- Role A can be assigned by an administrator, so add or remove actions will be defined.
- Role B has a rule that all members of Group "finance" have the role. This role cannot be assigned, so it has no add or remove action.

When you define add and remove actions, consider using the Admin Role attribute, which Identity Management can use to store a list of user's roles. For example, you can configure an add action that adds Employee to a user's Admin Role attribute when that user is added as a member of the Employee role. When an administrator assigns the Employee role to a manager who already has the Self Administrator and User Manager roles, the manager's Admin Role attribute would contain the following values: Self Administrator, User Manager, Employee.

To use the Admin Role attribute, the %ADMIN_ROLE_CONSTRAINT% well-known attribute must be mapped to a multi-valued attribute in user profiles. For more information, see the *Identity Management Configuration Guide*.

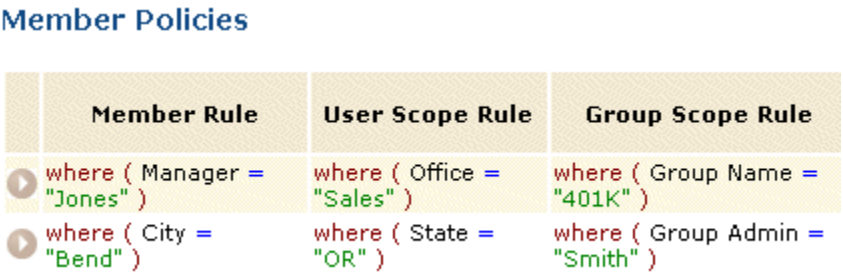
Important! When defining an add action, avoid setting up a rule that refers to the role you are defining. For example, do not define the add action that makes a member of Role A by being a member of Role A. This will create a recursive error that will cause the policy server to restart.

Member Policies

A *member policy* indicates that if a user meets the member rule, that user has the scope defined in that policy. The following figure shows a role that has two member policies.

- The first policy indicates that if a role member has the Manager Jones, that member can use the role on users in the Sales Office and manage them as members of the 401k group.

- The second policy indicates that if a role member is in the city Bend, that role member can use the role on users in the state of Oregon and manage them as members of the groups that have the Group Admin of Smith.

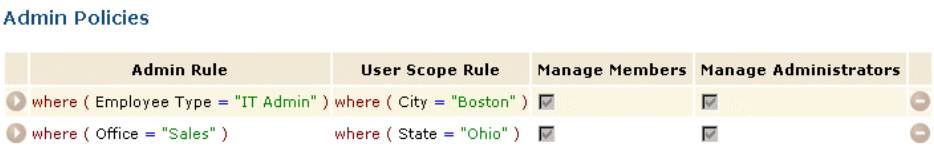


Admin Policies

An *admin policy* indicates that if a user meets the admin rule, that user has the user scope and administrator privileges defined in that policy. The user scope defines where the role is used. The administrator privileges determine if the role administrator can manage members or manage administrators of the role.

The following figure shows a role that has two admin policies, which are defined as follows:

- For the first policy, an IT Admin can add and remove role members and administrators from the users in the city of Boston.
- For the second policy, an administrator in Sales can add and remove members in the state of Ohio.



Role Planning Checklist

Before creating a role, use this checklist of role characteristics.

Role Characteristic	Details
Role Profile	Define a name and description for the role and set Enabled status.
Tasks	Include admin or access tasks.
Account Templates	Include account templates that define accounts that exist in endpoints (provisioning roles only).
Member Policies	For each member policy, define: ■ Member Rules -- Who can use the role ■ Scope Rules -- Which objects can a role member manage ■ Add Action -- What happens to the profile of a user who becomes a member ■ Remove Action -- What happens to the profile of a user who is removed as a member
Admin Policies	For each admin policy: ■ Admin Rules -- Who can manage the users as members or administrators ■ Scope Rules -- Which users can the administrator manage as members or administrators ■ Add Action -- What happens to the profile of a user who becomes an administrator ■ Remove Action -- What happens to the profile of a user who is removed as an administrator
Owner Rules	Define who can modify the role.

Chapter 2: Admin Roles

This section contains the following topics:

Admin Roles and Admin Tasks

You create roles that contain tasks for managing objects that are based on your individual business requirements. For example, you create several roles with tasks that manage users and other roles with tasks that manage the roles you create.

Alternatively, you create separate roles with:

- Tasks for administrators to manage users
- Tasks that manage the administrators
- Tasks to manage admin roles
- Tasks to manage access roles

Note: You can also use the default admin roles that are supplied with Identity Management. These roles have tasks that are grouped in categories similar to the preceding list.

Admin Roles and Identity Management Environments

When you log into a Identity Management environment, your user account has one or more admin roles. Each admin role contains tasks, such as Create User, that you use in that Identity Management environment.

For example, in the *central* Identity Management environment, an admin role, *Help Desk,* has tasks for resetting passwords. The role has a member rule that the user must be an IT employee. When IT employees log into the *central* Identity Management environment, they have the *Help Desk* role and can reset the passwords of users in that Identity Management environment.

Admin Roles and the User Console

A Identity Management environment is viewed through the User Console. Your assigned admin roles determine what you see in that console as shown in the following table:

Assigned Roles	Format of the User Console
System Manager role	The category list for all objects and all default admin tasks for managing those objects
Roles for managing more than one type of object	The category list with one item for each type of object you can manage
Roles for managing one type of object, such as Users	The tasks for that object (such as Modify User) *without* a category list
An approval role	The Work list screen Appears if the administrator has tasks pending approval (for example, self-registering users need approval)

If you can manage more than one object, the category list appears and shows the objects that you can modify, such as Users and Groups as tabs across the top of the screen. Select a tab to see tasks in your assigned roles.

Note: If your internet browser does not support Cascading Style Sheets (CSS), the User Console uses a different format. To control that format, see the *Configuration Guide*.

Create an Admin Role

You can create an admin role once you know the role requirements. These requirements concern the following questions:

- Users who need this role
- The objects that this role manages
- The Environment with the objects to manage

Begin Admin Role Creation

You create an admin role from the User Console.

To create an admin role

1. Log in to a Identity Management account that has a role with tasks for creating admin roles.

   For example, the first user of an Environment has the System Manager role, which has the Create Admin Role task.

2. Select Roles and Tasks, Admin Roles, and Create Admin Role.
3. Decide if you want to create or copy a role.

   The Profile tab appears where you begin defining the admin role.

4. Define the Admin Role Profile.

Define the Admin Role Profile

On the Profile tab, you define basic characteristics of the role.

To define the profile

1. Enter a name and description, and complete any other custom attributes that are defined for the role.

   Note: You can specify custom attributes on the Profile tab that specify additional information about admin roles. You can use this additional information to facilitate role searches in environments that include a significant number of roles.

2. Select Enabled if you are ready to make the role available for use as soon as you create it.
3. Select Admin Tasks for the Role (see page 32).

More Information:

User-defined Custom Attributes for Roles (see page 35)

Select Admin Tasks for the Role

On the Tasks tab, you select the admin tasks to include in the role. You can include tasks from different categories or copy tasks used in another role.

To select admin tasks

1. Select the category in the Filter by Category field.

   To view the list of available task categories, click the down arrow icon.

2. Select that task to include in the role in the Add Task field.

   Identity Management adds the task to the list of tasks in the role.

3. Add additional tasks by repeating steps 1 and 2.

4. Remove a task from the role by clicking the minus icon ( ) for that task.

5. Define Member Policies for an Admin Role (see page 33).

Define Member Policies for an Admin Role

On the Members tab, you create member policies, which determine who can be a role member.

To define member policies

1. Click Add to define member policies. A member policy contains these rules:
   - A member rule which defines the requirements for a user to be a role member.

     Note: The following operators treat numbers as characters in member rules:

     - Less than (<)
     - Less than or equal to (<=)
     - Greater than (>)
     - Greater than or equal to (=>)

     For example, '10' will come after '1' but before '2'.
   - Scope rules which limit the primary and secondary objects available to tasks in the role.

     For example, the role contains a task that modifies users by assigning them to groups. As a result, the user scope rule limits the users (primary object) that can be found and the group scope rule limits the groups (secondary object) that can be assigned.

   Note: Be sure to enter an answer to at least one scope question. The scope rules limit the primary and secondary objects available to tasks in the role. For example, the role contains a task that modifies users by assigning them to groups. As a result, the user scope rule limits the users (primary object) that can be found and the group scope rule limits the groups (secondary object) that can be assigned.
2. Verify that the Member Policy appears on the Members tab.
   - To edit a policy, click the right arrow symbol on the left.
   - To remove it, click the minus sign icon.
3. On the Members tab, optionally enable the checkbox labeled “Administrators can add and remove members of this role." Leaving this checkbox disabled means that users become members by meeting a member rule.

   Once you enable this feature, the screen expands.
4. In the expanded area, define the Add Action and Remove Action (see page 24) for when a user is added or removed as a role member.

   Important! For the add action, avoid setting up a rule that refers to the role you are defining. For example, do not define the add action that makes a member of Role A by being a member of Role A.
5. Define Admin Policies for an Admin Role (see page 34).

Define Admin Policies for an Admin Role

On the Administrators tab, you define who can add or remove users as members and administrators of this role.

To define admin policies

1. If you want to make the Manage Administrators option available, enable the check box labeled “Administrators can add and remove administrators of this role.”

   Once you enable this feature, the screen expands.

2. In the expanded area, define the Add Action and Remove Action for when a user is added or removed as an administrator of the role.

3. Define admin policies, which contain admin and scope rules and at least one administrator privilege (Manage Members or Manage Administrators).

   Note: You can add several admin policies with different rules and different privileges for administrators who meet the rule.

4. To edit a policy, click the arrow symbol on the left. To remove it, click the minus sign icon.

5. Define Owner Rules for an Admin Role (see page 34).

Define Owner Rules for an Admin Role

On the Owners tab, you define rules about who can be an owner of the role, a user who can modify the role.

To define owner rules

1. Define owner rules, which determine which users can modify the role.

2. Click Submit.

   A message appears to indicate that the task has been submitted. A momentary delay occurs before a user can use the role.

The role is available to be used. A user who meets conditions in the member rule can now log in to the environment and that user can use the tasks in the role.

Verify an Admin Role

Follow these steps:

1. Choose Admin Roles.
2. Choose View Admin Role.
3. Select the name of the role.

Alternatively, you can choose System, View Submitted Tasks to see if the role creation task has completed.

User-defined Custom Attributes for Roles

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

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

- Admin Roles
- Provisioning Roles
- Access Roles

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

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

More Information:

Configure Custom Attributes in Profile Tab for Roles (see page 36)
Add Custom Attributes to Search Screen Definitions (see page 36)

Configure Custom Attributes in Profile Tab for Roles

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

To configure custom attributes in the Profile tab

1. Click either:
   - Tasks, Roles and Tasks
   - Roles and Tasks
2. Click Admin Tasks, Modify Admin Tasks.

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

   Identity Management displays the task details for the selected admin task.
4. Click the Tabs tab.

   The tabs that are configured for use with this admin task appear.
5. Click the arrow icon to edit the Profile tab.

   The Configure Profile screen appears.
6. Select the checkbox next to each custom field to add to the Profile tab and enter a meaningful label.
7. Click OK.

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

   Note: To use the custom attributes in role searches, configure the search screen (see page 36) to display these custom attributes.

Add Custom Attributes to Search Screen Definitions

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

To add Custom Attributes to the Search Screens of roles

1. Click either:
   - Tasks, Roles and Tasks
   - Roles and Tasks
2. Click Admin Tasks, Modify Admin Tasks.

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

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

   Identity Management displays the task details for the selected admin task.
4. Click the Search tab in the Modify Admin Role screen.

   The search screen details appear.
5. Click the Browse button to display a list of search screen definitions that are available for the task.

   The Select Screen Definition page appears.
6. Select a search screen definition to edit, or create a copy of an existing search screen definition.

   The Configure Standard Search Screen appears.
7. Add the custom attributes to the following tables:
   - Select the fields that a user can search on
   - Select the fields that appear in the search results
8. Change the name of the custom attribute to match the name you specified when you configured the Profile tab.
9. Click OK to save the changes to the search screen definition.

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

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

Allow Users to Self-Assign Roles

Users can assign certain roles to themselves. For example, you may want to allow users to sign up for the Delegation Manager role so that they can delegate the work items of one user to another user.

To control the roles that users can assign to themselves, you configure criteria in the Roles Self-Manager task.

Follow these steps:

1. Modify the Roles Self-Manager task as follows:
   a. Select Roles and Tasks, Modify Admin Task, and search for the Roles Self-Manager task.
   b. Select the Tabs tab.

      Identity Management displays the list of tabs that apply to the task.
   c. Select the right arrow icon next to the Roles Self Manager tab to edit it.
   d. Complete the following fields:

      Show only Admin Roles Meeting the Following Rules

      Specifies the criteria that Identity Management uses to determine which roles to allow users to assign to themselves.

      To add additional rules, click the plus (+) icon.

      User to be used as Admin Role Administrator

      Specifies the administrator for roles that users can assign to themselves.

      The roles that users can assign to themselves must have the user you select in this field as an administrator *and* meet the criteria you specified in the Show Only Admin Roles Meeting the Following Rules field.

      List Screen

      Specifies the columns and format for the list of roles that a user can select to self-assign a role.
   e. Click OK, then click Submit.
2. Add the Roles Self-Manager task to a role, and assign that role to users who should have this capability.

Chapter 3: Admin Tasks

This section contains the following topics:

Admin Task Planning

Admin roles consist of admin tasks, which represent granular capabilities for managing objects. For example, you could manage a user object by using these admin tasks:

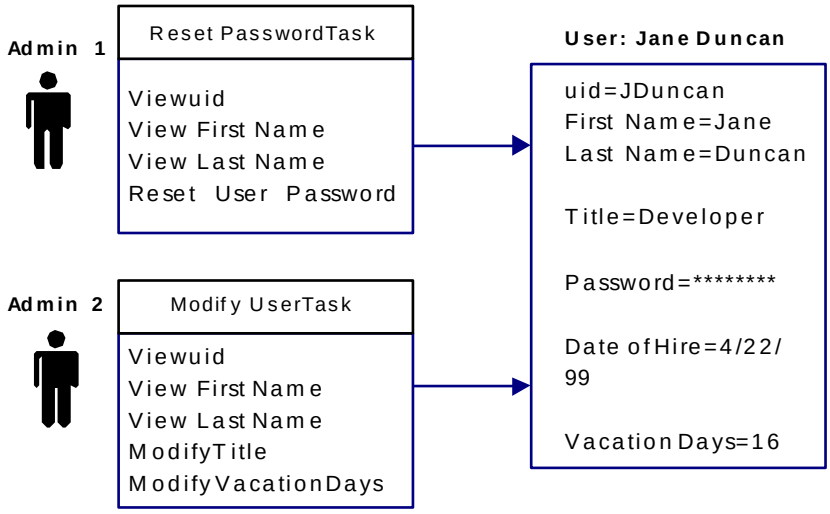
- Create User
- View User
- Modify User
- Reset User Password

You create or modify each task to match your exact requirements. Then, you combine the appropriate admin tasks into admin roles, which you assign to administrators. With these roles, administrators have the exact privileges they need to manage objects.

To plan admin task creation, decide which objects you need to manage (user, group, organization, role, or task) and which administrators will use these tasks. For example:

- To manage users, help desk administrators need tasks that manage user attributes, such as a user ID or title.
- To manage users' access to applications, other administrators need tasks that make users members of access roles.
- To manage the roles used by help desk administrators, higher-level administrators need tasks that manage admin roles.

For one type of object, such as users, you can create tasks so that different administrators manage different attributes. For example, the following figure shows a user who is managed by two administrators.



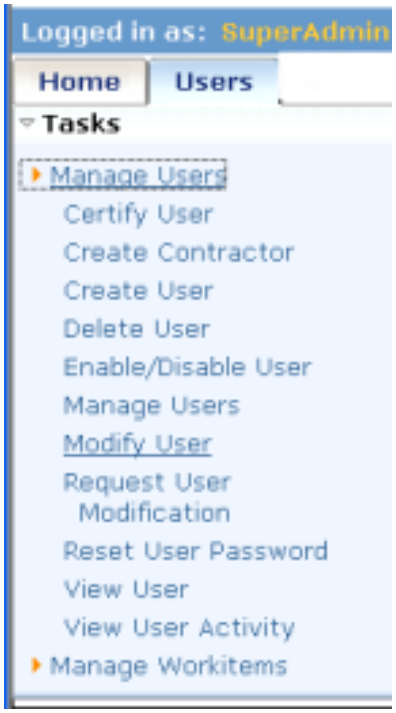
- Admin 1 has the Reset User Password task; that administrator can view the employee's user ID and name or reset her password.
- Admin 2 has the Modify User task; that administrator can view the employee's user ID and name or modify her title and vacation days.

A Sample Admin Task

When you create an admin task, you define the content and layout of screens in the task, including:

- The name of the task
- The category where the task appears
- The tabs and fields to use in the task, and field display properties
- The fields an administrator can use in a search query, and the fields displayed in the search results

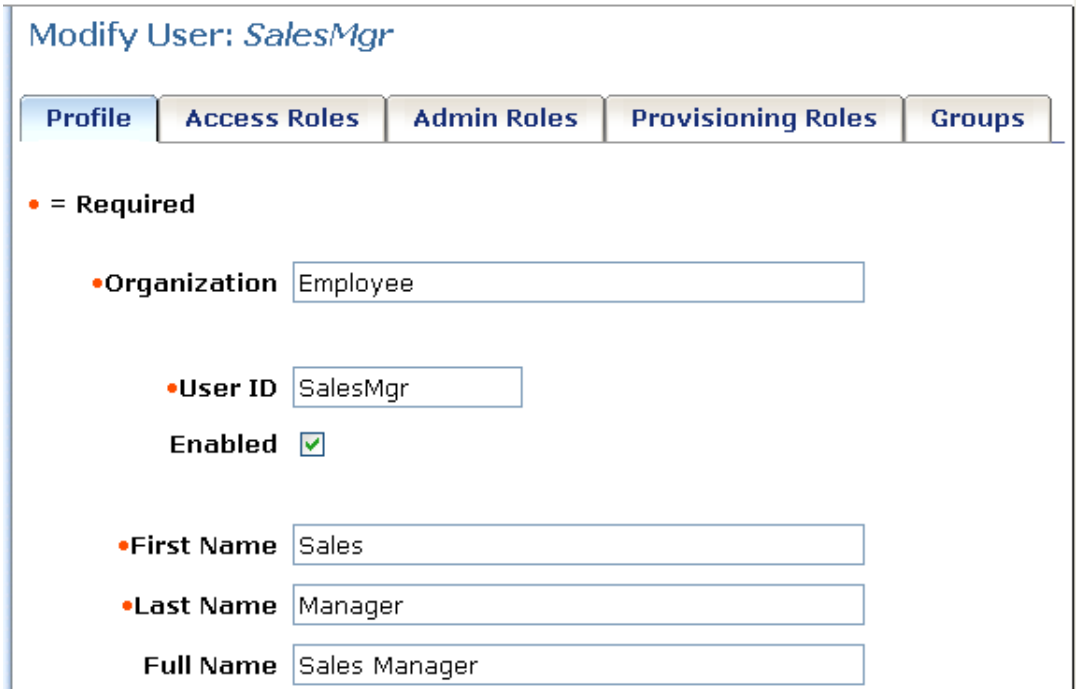
To understand the elements of a task, consider the Modify User task. In this case, Users is the category, Manage Users is a subcategory, and Modify User is the task. You create the category and task names when you create a task.



When you choose Modify User, a search screen appears. A *search screen* provides options for finding the object to view or modify. Each option is called a *filter*, which is a limit to the objects found by the search.

After you fill in the search screen, a screen with tabs appears. For example, the following figure shows the tabs for the Modify User task. The Profile tab appears first and shows user attributes; the other tabs show role and group privileges for the user.

For the task you create, you decide which tabs to include and determine their order and content.



For example, using the Modify User task as a template, you could create a Modify Contractor task, which has changes to:

- The fields on the Profile tab
- The tabs to include in the task and their content
- The category under which the task appears

  You might create this task under a new category, Contractor.

The Modify Contractor task includes some of the fields on the Profile tab in the Modify User task plus other fields, such as the start date of the contract and the contractor's company. Administrators can search for a contractor by searching on the contractor's name, company, and start date.

The new task also includes a Contractor Roles tab where you add roles for contractors.

Admin Task Usage Options

Identity Management provides two ways to use admin tasks:

- **Select the task**

  You select a category and task, and then search for the object to which the task applies.

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

- **Select the object**

  You use "Manage" tasks, such as Manage Users or Manage Groups to search for an object. Once you select the object, you can display a list of tasks that you can use to manage that object. This method is called *object-task navigation.*

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

  You can also configure task lists in tasks other than Manage tasks. For example, you can add a task list to a Membership tab. In this case, a task list is available for each member that appears on the Membership tab.

  Note: Only tasks that the current administrator can use appear in the task list for an object.

How to Create a Custom Admin Task

An *admin task* is an administrative function that a user can perform in Identity Management. Examples of admin tasks include Create User, Modify Group, and View Role Membership.

Identity Management includes default admin tasks that you can modify to suit your business needs.

When you create a custom admin task, you complete the following steps:

Note: The section Active Directory Prerequisites (see page 52) includes additional considerations if Identity Management is managing an Active Directory user store.

1. In the Identity Management User Console, select Roles and Tasks, Admin Tasks, Create Admin Task.

   Identity Management asks if you want to create a new task or create a task based on an existing task.

   For example, select the Modify User task as the basis of the new task.

2. Select Create a Copy of an Existing Task, and search for the task to copy.

   Note: We recommend modifying the copy of a default task, instead of modifying the default task directly.

3. Once you select Ok, you see a screen with the following six tabs:

Tab	Purpose	See this Topic
Profile	Define the profile of the task being created	Define the Profile of the Task (see page 45)

Tab	Purpose	See this Topic
Search	Limit the range of objects that are managed by the task	Define the Task Scope (see page 50)
Tabs	Choose and design the tabs for the task	Choose Tabs for the Task (see page 51)
Fields	Show the fields used on all tabs	View Fields in the Task (see page 51)
Events	Select a workflow process for each event if the Identity Management environment and the task uses workflow	Assign Workflow Processes for Events (see page 52)
Role Use	Displays the roles that include the task that you are modifying or viewing	View Role Use (see page 51)

Note: For more information about creating custom admin tasks, see the *User Console Design Guide*.

Define the Profile of the Task

The Profile tab includes general settings for the task.

Note: For more information about admin task profile settings, see the *User Console Design Guide*.

To define the profile of the task

1. Choose the type of object for the task, which is called the primary object, and the action to perform on it.
2. Complete the required fields and select the appropriate check boxes as needed for the task.

   Note: If you are creating a task that has similar profile settings as an existing task, click Copy Profile From Another Task. This option populates the profile settings for the task you are creating with the profile settings from any existing task that you select. You then add a name and description for the new task.
3. (Optional) Associate a business logic task handler with the task.
4. Once you complete this tab, proceed to the next step, Define the Task Scope (see page 50).

Admin Task Profile Tab

The Admin Task Profile Tab lets you define general settings for an admin task.

This tab contains the following fields:

- **Name**

  Defines the name of the task.

- **Tag**

  Defines a unique identifier for the task. It is used in URLs, web services, or properties files. The tag can contain ASCII characters (a-z, A-Z), numbers (0-9), or underscore characters, beginning with a letter or underscore.

- **Description**

  Specifies an optional note about the purpose of the task.

- **Task Order**

  Specifies the display order for the task. If no order is specified, the tasks are displayed in alphabetical order.

- **Category**

  Specifies a category for the task. Categories are displayed as tabs at the top of the screen.

- **Category Order**

  Specifies the order in which the category tab appears. For example, if you set the category order to 3, the category you specified will appear as the third tab.

- **Category 2**

  Specifies the second level category, which appears as a link below the list of category tabs. The second level category appears only when the tab for the first level category is selected. For example, if you created a task with the first level category of Employee and a second level category of Employee Management, the Employee Management category would appear only after you select the Employee tab.

- **Category 2 Order**

  Specifies the order in which the second level category appears, if more than one second level category exists in a primary category.

- **Category 3**

  Specifies the third level category, which appears in the left navigation pane. Tasks are listed under the third level category. For example, in a default environment, a user with the System Manager or User Manager role sees the third level category Manage Users when he selects the Users tab.

- **Category 3 Order**

  Specifies the order in which the third level category appears.

- **Primary Object**

  Specifies the object that the task operates on.

- **Action**

  Specifies the operation to perform on the object.

- **User Synchronization**

  Specifies whether the task synchronizes users with identity policies. You can select one of the following options:

  - **Off** (default)

    Specifies that this task does not trigger user synchronization.

  - **On Task Completion**

    Specifies that Identity Management starts the user synchronization process after all of the events in a task complete. This setting is the default synchronization option for the Create User, Modify User, and Delete User tasks. The default setting for all other tasks is Off.

    Note: If you select the On Task Completion option for a task that includes multiple events, Identity Management does not synchronize users until all of the events in the task complete. If one or more of those events require workflow approval, this may take several days. To prevent Identity Management from waiting to apply identity policies until all events complete, select the On Every Event option.

  - **On Every Event**

    Specifies that Identity Management starts the user synchronization process (see page 198) when each event in a task completes.

    For tasks with a primary and secondary event for the same user, setting user synchronization to On Every Event may result in more identity policies being applied to a user than if the On Task Completion option is selected.

- **Account Synchronization**

  Synchronizes accounts that exist in the Provisioning Server, if you have provisioning enabled.

  - **Off** (default)

    Specifies that this task does not trigger account synchronization.

  - **On Task Completion**

    Specifies that Identity Management starts the account synchronization process after all of the events in a task complete.

- **On Every Event**

  Specifies that Identity Management starts the account synchronization process when each event in a task completes.

  Note: For best performance, select On Task Completion. However, if you select the On Task Completion option for a task that includes multiple events, Identity Management does not synchronize accounts until all of the events in the task complete. If one or more of those events require workflow approval, this may take several days. To prevent Identity Management from waiting to synchronize accounts until all events complete, select the On Every Event option.

- **Hide in Menus**

  Prevents the task from appearing in menus. Enable this control if the task is only invoked by a URL or by another task.

- **Public Task**

  Makes the task available to users who have not logged in to Identity Management. The default public tasks are forgotten password and self-registration.

- **Enable Auditing**

  Records information about the task in an auditing database. Audit information can be used to generate reports. See the *Configuration Guide*.

- **Enable Workflow**

  Enables the Identity Management events associated with the task to trigger workflow processes, if you have the workflow engine installed. For example, the events associated with the Delete Group task may trigger a workflow process that includes an approval step.

- **Enable Web Services**

  Marks the task as one for which Web Services Description Language (WSDL) output can be generated from the Management Console. Enable this control if you want to use remote task submission. For more information, see the *Programming Guide for Java.*

- **Workflow Process**

  Enables configuration for task level workflow. Click the pencil icon to configure policy-based or non-policy based workflow.

- **Task Priority**

  Determines the order in which Identity Management executes tasks. Tasks with a High priority are executed before tasks with a Medium or Low priority. The default priority for a task is Medium.

  Note: You can use the View Submitted Tasks task to search for tasks with a specific priority, and then display their status.

- **Business Logic Task Handlers**

  Associates a business logic task handler (see page 56) with the task.

- **Workflow Action Buttons**

  Add custom action buttons to workflow approval tasks.

- **Copy Profile from another task**

  Copies data from the Profile tab of another task.

  For example, you might copy the Profile tab settings from the Modify User task, then add a name and description.

Task Configuration Properties

Task configuration properties control display properties and certain behaviors for the task.

Task Icon Path

Specifies the URL for a graphic to use as an icon for this task in task lists.

Task Icon Preview

Displays the icon for the task, as it appears in task lists.

Suppress Task Navigation

When selected, hides the top-level navigation and task list once a user selects a task. This prevents users from navigating away from the current task until they complete required actions or cancel the task.

Target Window

When you provide a value in this field, Identity Management opens this task in a new browser window. Use this field to open a new browser window for an external task that redirects users to another website.

You can specify any name for the window.

Note: Do not use this field to open Identity Management admin tasks in a separate browser window. Identity Management does not support multiple browser windows for a single Identity Management user session.

Define the Task Scope

On the Search tab, you define the task scope, which limits the objects available to the task. For example, if the object of a task is users, you might define the scope as users who are contractors.

Note: If the task has no primary object, or if the action is self-modify, self-view, or approve, the Search tab does not appear.

You configure the following settings on the Search tab:

Search Screen

The search screen limits the scope of the task based on filters. Click Browse to see available search screen options.

Note: You may want to create your own search screen. To create a modified version of an existing search screen, select the search screen and click Copy. You can then modify the search screen without changing the original search screen definition. To create a search screen, click New.

Search Options

The search options appear only when the object is a role or group.

- The first option limits the search based on fields that are defined on the search screen. Within these limits, the search locates all groups or roles in the administrator's scope.
- Other options limit the search as indicated.

Note the following:

- By default, the group search screens support filtering. This means that administrators can specify criteria to limit the scope of group searches. To remove the filtering capability, create a search screen that does not contain any fields to include in a search query.
- *Filtering not supported*, which appears on the Search tab when the object is a role, means that the task displays the roles that meet the criteria in the option you select. Search fields that are configured on the search screen are ignored.

Modified objects must remain in administrator's scope

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

Choose Tabs for the Task

On the Tabs tab, name and configure the tabs; each one is a set of fields that you include in the task. You can include default tabs or can create new ones. For example, the Modify User task includes the following tabs:

- Profile
- Access Roles
- Admin Roles
- Groups
- Delegate Work Items

To edit the definition of a tab, click the edit icon () next to the tab name.

View Fields in the Task

On the Fields tab, you view the fields that apply to this task. These fields are those created on the tabs for this task. To change the fields used, return to the Tabs tab and select the tab that requires the change.

Once you complete this tab, proceed to the next step, Assign Workflow Processes for Events (see page 52).

However, if this Identity Management environment does not use workflow, you can now click Submit. A message appears indicating if the task succeeded. If it succeeds, you can add the task to a role, so that role members can start using the task.

View Role Use

On the Role Use tab, you view the roles that include the task that you are viewing or modifying.

Role owners can add and remove tasks from roles.

Note: Default Admin Roles provides a list of tasks in the admin roles that are installed with Identity Management by default.

Assign Workflow Processes for Events

If you enabled workflow for this Identity Management environment, use the Events tab to select a workflow process for each event that the task initiates. The workflow process that you select overrides the one selected by default in the Identity Management Management Console.

For more detail on default workflow mappings, see the Advanced Settings chapter of the *Configuration Guide*.

To complete the creation of this task, click Submit. A message appears indicating if the task succeeded. If it succeeds, you can add the task to a role, so that role members can start using the task.

Manage an Active Directory User Store

If Active Directory is the user store, before creating admin tasks, you may need to configure certain Active Directory features.

The sAMAccountName Attribute

The sAMAccountName attribute applies to users and groups. This attribute is required, and must be included on task screens used to create users and groups.

Note: When creating users, the value of the sAMAccountName attribute cannot exceed 20 characters. This restriction does not apply to groups.

You can write a custom logical attribute handler that generates a unique sAMAccountName automatically when a user or group is created. In this case, you can include the sAMAccountName attribute as a hidden field on Create User and Create Group screens.

See the Logical Attributes chapter in the *Programming Guide for Java* for more information.

Group Type and Scope

In Active Directory, there are two types of groups:

- Security--Listed in Access Control Lists (ACLs), which define permissions for resources and objects.
- Distribution--Used to group objects, such as users and groups. Distribution groups cannot be used to grant privileges in Active Directory.

Each type of group has a scope that determines the following:

- Member location--Where potential members can reside
- Permissions--Where the group can be used for access privileges (if the group is a security group)
- Group Membership in Other Groups--The location of groups to which the group can belong

Each type of group can have one of the following scopes:

Scope	Member Location	Permissions	Group Membership in Other Groups
Universal	Group members can be Universal groups, Global groups, and users from any domain in the forest.	Can be used to grant access in any domain in a forest.	Can be members of Domain Local and Universal groups in any domain in the forest.
Global	Group members can be Global groups and users located in the same domain as the group.	Can be used to grant access in any domain in a forest.	Can be members of Global, Domain Local, and Universal groups in any domain in the forest.
Domain Local	Group members can be Universal groups, Global groups, and users from any domain in the forest. Members can also be Domain Local groups from the same domain.	Can only be used to grant access to the domain where the group resides.	Can only be a member of other Domain Local groups within the domain.

Group type and scope are not required attributes; however, if you do not specify group type and scope, Active Directory creates a security group with global scope.

To create groups of a different type, you can create a custom logical attribute handler. See the chapter on Logical Attributes in the *Programming Guide for Java*.

Once you have configured these Active Directory features, proceed to the next step: Create an Admin Task.

External Tasks for Application Functions

An external task does the following:

- Allows an administrator to perform a function in an application other than Identity Management from the User Console
- Optionally passes information to the application to generate user-, group-, or organization-specific tasks.

For example, an external task may pass information about an organization to an application that generates purchase orders. The administrator performing the task can view open purchase orders for the organization from the User Console.

You can view external tasks by opening the application in a new browser window, or by viewing them as tabs in a Identity Management admin task.

Two tabs are available for External tasks. These tabs are configured in the same way; however, they function differently.

- The External tab is a visual tab, which means that the task displays the contents of the URL within a tab.
- External URL is a non-visual tab, which means that the task redirects to the URL entered.

The External Tab

An external tab can be added to any Create, View, or Modify task to make it an external task. For example, if you add an External tab to a Create User task, the tab appears on that task.

For an External tab:

- No events are generated for an external task.
- You can optionally use managed objects.

- In the External URL field, you can specify the address of the application as:
  - A complete address, including the fully qualified domain name--for example:

    http://server1.mycompany.org/report/viewUserReport
  - A relative path--for example:

    /report/viewUserReport

    If you specify the relative path, Identity Management automatically appends the fully qualified domain name of the server where Identity Management is installed.
- You configure the attributes to pass to the application on the Profile tab.
- You can include or exclude the admin DN or task name in the URL.

The External URL Tab

You can add an external URL tab to a view task, such as View User. When you use the View User task, you are redirected to the web site identified by the URL. No other tabs are visible.

For an External URL tab:

- The external URL tab must be the only tab in the task. If there are other tabs associated with the same task, the external tab will not redirect users to the specified URL.
- The task can generate events which can be audited.
- In the External URL field, you can specify the address of the application as:
  - A complete address, including the fully qualified domain name--for example:

    http://server1.mycompany.org/report/viewUserReport
  - A relative path--for example:

    /report/viewUserReport

    If you specify the relative path, Identity Management automatically appends the fully qualified domain name of the server where Identity Management is installed.
- You can optionally use managed objects.
- You can configure attributes to pass to the URL.

  Supply a URL for the application that you want to start and include the attributes that you want to pass to the application.
- You can include or exclude the admin DN or task name in the URL.

Advanced Task Components

Advanced task components allow you to specify custom processing for a task:

- Task-Level Validation validates an attribute value against other attributes in the task. For example, you might validate that the area code in a user-supplied phone number is appropriate for the user's city and state.
- Business Logic Task Handlers (see page 56) perform custom business logic before a Identity Management task is submitted for processing. Typically, the custom business logic validates data. For example, a business logic task handler may check a group's membership limit before Identity Management adds a new member to the group. If the group membership limit is reached, the business logic task handler displays a message informing the group administrator that the new member could not be added.

Create Business Logic Task Handlers

You define a business logic task handler's fully qualified class name as follows:

1. Create or modify an admin task.
2. On the Admin Profile tab, click Business Logic Task Handlers.

   The Business Logic Task Handlers screen appears. This screen lists any existing business logic task handlers assigned to the task. Identity Management executes the handlers in the order in which they appear in the list.
3. Click Add.

   The Business Logic Task Handler Detail screen appears.

Use the Business Logic Task Handler Detail screen to define the following information for the business logic task handler you are assigning to the task:

Name

The name you are assigning to the business logic task handler.

Description

An optional description of the business logic task handler.

Java Class

If the business logic task handler is implemented in Java, the fully qualified business logic task handler class name--for example:

com.mycompany.MyJavaBLTH

Identity Management expects the class file to be located in the root directory designated for custom Java class files. For information on deploying Java class files, see the *Programming Guide for Java*.

JavaScript Filename

If the business logic task handler is implemented in JavaScript, and the JavaScript code is contained in a file, specify the file name in this field. For example, you might want to put the JavaScript in a file if the business logic task handler is to be used by several task screens.

Identity Management expects the file to be located in the root directory designated for custom JavaScript files. For information on deploying JavaScript files, see the *Programming Guide for Java*.

If you store the file in a subdirectory of the root, include the subdirectory name when you specify the JavaScript file name--for example:

JavaScriptSubDir\MyJavaScriptBLTH.js

The slashes must be appropriate for the platform where the JavaScript file is deployed.

JavaScript

You can implement a JavaScript business logic task handler by typing the complete JavaScript code in this field instead of in a file. For example, you might want to put the JavaScript in this field if the script is very short or if it is to be used with no other task screens.

Property and Value

With Java implementations, these fields are optional name/value pairs of data that are passed into the init() method of the Java business logic task handler, to be used in any way that the handler's business logic requires.

To add a user-defined property, specify a property name and value, and then click Add.

Note: If you add a Java business logic task handler, you restart the application server for the handler to be loaded.

Admin Tasks and Events

Admin tasks include *events*, actions that Identity Management performs to complete the task. A task may include multiple events. For example, the Create User task may include events that create the user's profile, add the user to a group, and assign roles.

Identity Management audits events, enforces customer-specific business rules associated with events, and, when events are mapped to workflow processes, requires approval for events.

If multiple events are generated for a task, and the events are mapped to workflow processes, all the workflow processes must be completed before Identity Management can complete the task.

Primary and Secondary Events

Generally, events are independent of other events. However, some tasks are associated with a primary event and one or more secondary events:

- A failure of a primary event results in the automatic rejection of all of its secondary events. For example, if a CreateUserEvent fails, there is no need for the AddToGroupEvent to occur for the user. It also results in the cancellation of the associated task.
- A failure of a secondary event does not affect the success or failure of any other events executed for the task or the execution of the task itself. For example, in a Create User task, an AddToGroupEvent may be rejected, meaning that the new user cannot be added to a particular group. The user can still be created (CreateUserEvent) and assigned to provisioning roles (AssignProvisioningRoleEvent), and even be added to other groups.

View the Events for a Task

You can view the events that are associated with a task in the Identity Management User Console.

To view the events for a task

1. Select Roles and Tasks, and View Admin Tasks in the User Console.
2. Search for and select the appropriate task.
3. Select the Events tab.

   Identity Management displays the events that are associated with the current task.

Events Generated for Unmodified Profiles

User, group, and organization objects each contain a set of physical attributes that are stored in the user directory. If a physical attribute of one of these objects is changed on a profile tab, Identity Management generates a Modify event after the user submits the task. For example, if a *Title* attribute is changed on a User Profile tab, Identity Management generates the event ModifyUserEvent.

If a user, group, or organization object is represented on a profile tab, but no physical attributes have been changed when the user clicks Submit, Identity Management does not generate a Modify event. Instead, the corresponding View event is generated, as follows:

- ViewUserEvent is generated instead of ModifyUserEvent
- ViewGroupEvent is generated instead of ModifyGroupEvent
- ViewOrganizationEvent is generated instead of ModifyOrganizationEvent

Admin Task Processing

The time it takes to process a task depends on the steps involved. When a task is submitted for processing, Identity Management performs the following steps:

1. Identity Management validates the data being submitted.

   This is called the *synchronous phase*.

2. If the task requires approval, Identity Management sends the task to the workflow engine.

   a. The workflow engine determines approvers, and places the approval task in the approvers' work lists.

   b. Optionally, Identity Management sends email notifying approvers of the pending work item.

   c. An approver reserves the work item (which removes the item from the work lists of other approvers), and approves or rejects the item.

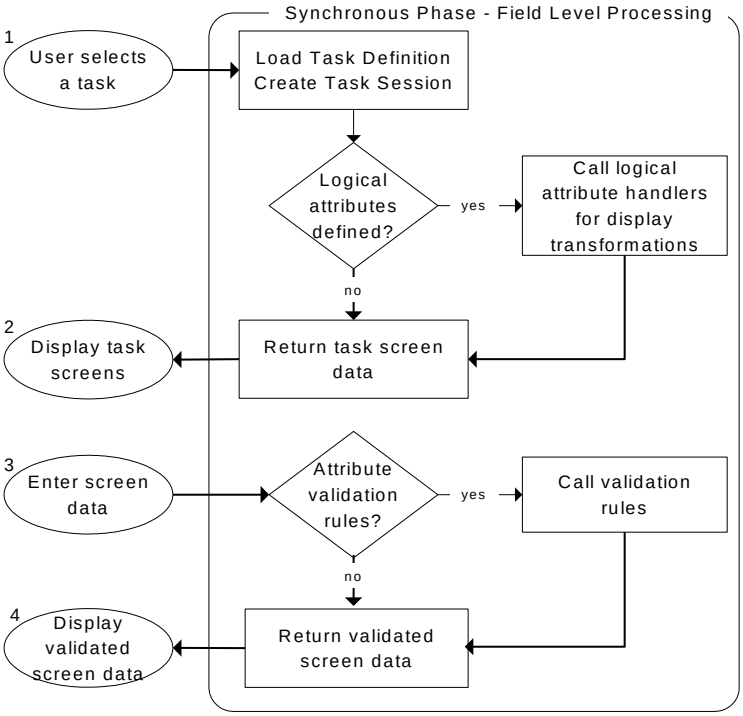
   d. Optionally, Identity Management sends email notifying involved users of the task's status.

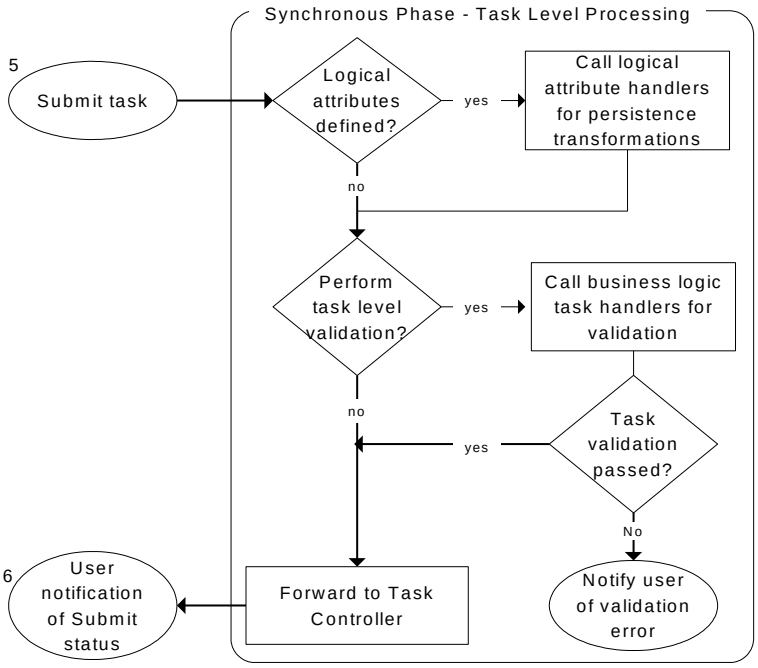
   This is called the *asynchronous phase*.

3. Identity Management carries out the task, if the task was not rejected.

Synchronous Phase Processing

During the synchronous phase, Identity Management can transform and validate data that users enter in task screens, and can enforce business logic on that data before the task is submitted for processing. The following diagram provides a high level description of what occurs during this phase.

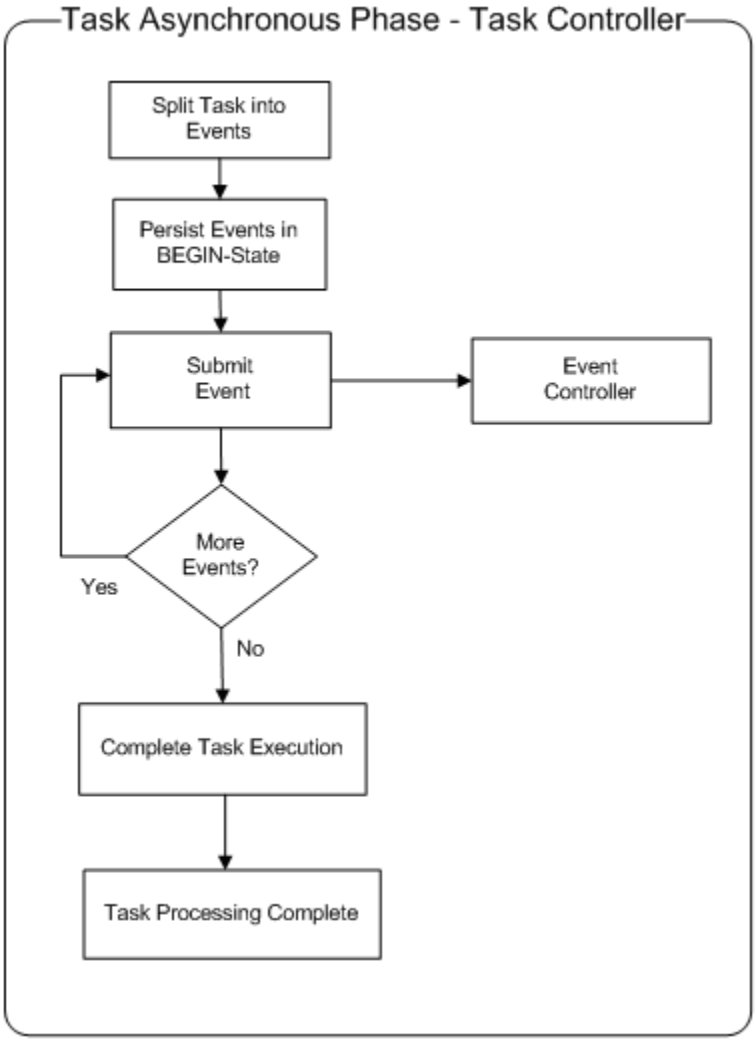




Asynchronous Phase Processing

Upon completion of the synchronous phase, the task enters the asynchronous phase for execution. During this phase, a task generates one or more events. These events may be user-defined, such as creating a user profile or adding a user to a group, or system-generated, such as writing information to the audit log.

The task controller, a component of the Identity Management Server, is responsible for the life cycle of a task and its events, as shown in the following illustration:



For most events, the life cycle, execution, and actions are independent from any other event. (Create tasks require the primary object's create event to execute before any secondary events.)

Typically, an event transitions through the following states:

- Begin
- Pending
- Approved
- Execute
- Completed
- Post

Note: Identity Management provides hooks, called EventListeners, that "listen for" a specific event or a group of events. When the event occurs, the event listener performs custom business logic that is appropriate for the event and the current event state. You can use the Event Listener API to write custom event listeners. See the *Programming Guide for Java* for more information.

Images for Admin Tasks

You can create images to use for admin tasks you put on the home page.

Chapter 4: Password Management

This section contains the following topics:

Password Management in Identity Management

Identity Management includes several features for managing user passwords:

- Password Policies--These policies manage user passwords by enforcing rules and restrictions governing password expiration, composition, and usage.
- Password Managers--Administrators who have the Password Manager role can reset a password when a user calls the Help desk.
- Self-service password management--Identity Management includes several self-service tasks that allow users to manage their own passwords. These tasks include:
  - Self Registration--Users specify a password when they register at a corporate Web site.
  - Change My Password--Users can modify their passwords without help from IT or helpdesk personnel.
  - Forgotten Password--Users can reset or retrieve a forgotten password after Identity Management verifies their identity.
  - Reset Password or Unlock Account--Users can reset or retrieve a forgotten password or unlock a windows account on a system where they access Identity Management.
  - Forgotten User ID--Users can retrieve a forgotten user ID after Identity Management verifies their identity.
- Password synchronization on endpoint accounts--Password changes are synchronized in Identity Management, the Provisioning Server, and its target systems. New passwords are verified against Identity Management password policies.

Password Policies Overview

A password policy is a set of rules and restrictions. These rules specify password creation and expiration. When you configure a password policy in a Identity Management environment, the policy applies to the user store associated with the environment. If a user directory is associated with multiple environments, a password policy defined in one environment can apply in other environments.

In a password policy, you can configure the following settings:

Note: Some of these settings require user directory mappings for certain attributes. See Enable Additional Password Policies (see page 67).

- Apply passwords to a specific set of users
- Password expiration—Define events, such as a number of days elapsing or a number of failed login attempts, that cause a password to expire. When a password expires, the user account is disabled.
- Password composition—Specify the content requirements for new passwords. For example, you can configure settings that require users to create passwords which are at least eight characters long and contain a number and a letter.
- Regular expressions—Provide an expression that determines the format of a valid password. You can specify whether passwords match or do not match that format. You can also specify multiple regular expressions.
- Password restrictions—Set limits on password reuse. For example, users must wait 90 days before reusing a password.
- Advanced password options—Specify actions that Identity Management takes, such as making passwords lower case, before processing a password. You can also specify the priority of a password policy when multiple password policies apply.

SiteMinder users can also configure password policies in the SiteMinder Administrative user interface. These policies appear in the Identity Management User Console.

Note: When Identity Management integrates with SiteMinder, SiteMinder enforces *all* password policies.

Create a Password Policy

You create password policies through the Identity Management User Console.

Note: The availability of some password policy options requires mapping certain well-known attributes. See Enable Additional Password Policies (see page 67).

Follow these steps:

1. In the User Console, choose either:
   - Policies, Manage Password Policies, Create Password Policy.
   - Tasks, Policies, Manage Password Policies, Create Password Policy.
2. Enter a unique name and an optional description for the password policy.
3. Configure these password policy settings as best suits your implementation:
   - Apply a Password Policy to a Set of Users (see page 68)
   - Configure Password Expiration (see page 70)
   - Configure Password Composition (see page 73)
   - Specify Regular Expressions (see page 74)
   - Set Password Restrictions (see page 76)
   - Configure Advanced Password Options (see page 79)

Enable Additional Password Policies

Identity Management lets you create basic password policies that manage user passwords by enforcing password expiration, composition, and usage. You can also define these additional password rules and restrictions:

- Password expiration:
  - Track failed logins or successful logins.
  - Authenticate a login.
  - Password expiration if not changed
  - Password inactivity
  - Incorrect password
  - Multiple regular expressions
- Password restrictions:
  - Minimum days before reuse
  - Minimum number of passwords before reuse

- Percent different from last password
- Ignore sequence when checking for differences.

Follow these steps:

1. Go to Directories, <name of the directory>, User in the Management Console.
2. Verify that the %PASSWORD DATA% and %ENABLED STATES% -> 'STATE' are mapped to physical attributes.
3. These attributes are mapped by default in the sample directory.xml files. If these attributes are not mapped, see the *CA Identity Manager Configuration Guide* for additional information.

Apply a Password Policy to a Set of Users

You can specify rules that determine the set of users to which a password policy applies. This ability allows you to have one password policy for general employees, and a stricter policy for high-level managers.

Follow these steps:

1. Create or modify a password policy in the User Console.
2. Select the type of filter to configure in the Directory Filter field.

   See the following table for a description of each filter type.

   Note: The type of user store to which the password policy applies determines the options for the Directory Filter list box. Some filter types are not available for relational databases and CA Directory user stores when Identity Management is integrated with CA SiteMinder.
3. Specify a condition by selecting an attribute and operator, and entering a value.
4. To add additional conditions, click the plus sign.

The following table describes the options for directory filter types, and provides examples of each filter type. Attributes on the left side of the "=" in the following examples are as they are prescribed in the user directory definition area. For Create-type user tasks, password policies with directory filters configured are only applied when both of the following conditions are met:

- Identity Management is not integrated with CA SiteMinder.
- The directory filter type is not User, Group, Group Filter, or Group Search.

Type of Filter	Use this filter to...	Example
In an Organization	Browse and select an Organization.	

Type of Filter	Use this filter to...	Example
In a Group	Browse and select a Group.	
A user	Browse and select a single user.	
User filter (Not available for relational databases when integrated with CA SiteMinder)	Specify a filter for users.	Employee Type = Contractor Department = Security
User Search Expression	Enter a search query for users.	uid=jsmith (for LDAP) TBLUSERS.ID = jsmith (for relational databases)
Group Filter (Not available for relational databases when integrated with CA SiteMinder)	Specify a filter for groups.	Self Subscribing = *
Group Search Expression	Enter a search query for groups.	cn=Sales (for LDAP) TBLGROUPS.NAME=GroupA (for relational databases
Organization Filter (Not available for relational databases when integrated with CA SiteMinder)	Specify a filter for organizations.	Organization name = *Marketing
Organization Search Expression	Enter a search query for organizations.	ou=Boston (for LDAP) TBLORGANIZATIONS.NAME=Boston (for relational databases)
Search	Specify a query that is not included in the other options for the filter type.	(&(uid=*smith)(ou=Boston))

Configure Password Expiration

To help manage user access, you can define events such as multiple failed login attempts or account inactivity. When these events occur, Identity Management disables the user account that is responsible. When Identity Management is integrated with SiteMinder, you can specify a redirection.

Note: These settings require additional configuration. See Enable Additional Password Policies (see page 67).

You can configure the following settings for password expiration:

- Track Failed/Successful Logins Check Box
- Authenticate on Login Tracking Failure Check Box
- Password Expires if Not Changed Settings
- Password Expires from Inactivity Settings
- Incorrect Password Settings

Track Failed/Successful Logins Check Box

This check box enables and disables tracking of user login attempts, including the time of the last login attempt. If you enable this check box, Identity Management writes login information to a password data attribute in the user store.

Note: This setting requires additional configuration. See Enable Additional Password Policies (see page 67).

When the Track Failed Logins check box is enabled, the Incorrect Password section and the Authenticate on Login Tracking Failure Check Box are active. When the Track Successful Logins check box is enabled, the Password Expires from Inactivity section and the Authenticate on Login Tracking Failure Check Box are active.

If you have multiple password policies, be sure that all applicable password policies disable login details. Otherwise, a single policy which enables the tracking of login details can cause password policies to behave incorrectly.

Authenticate on Login Tracking Failure Check Box

Selecting this check box enables logins when user tracking fails. By default, this check box is disabled. When log-in tracking is disabled, users cannot log in.

When you select this check box, be sure that you also select the Track Failed Logins or Track Successful Logins check box.

Note: This setting requires additional configuration. See Enable Additional Password Policies (see page 67).

Password Expires if Not Changed Settings

In the Password Expires if Not Changed fields, you can configure behavior for passwords that have expired. Optionally, you can specify how far in advance users are warned that their password is due to expire.

Note: This setting requires additional configuration. See Enable Additional Password Policies (see page 67).

You can configure the following fields:

After *<number>* Days

Determines the number of days after a password expires that Identity Management waits before disabling the user, or forcing a password change.

Note: Identity Management does not disable the user account until the user attempts to log in after the specified number of days has elapsed.

Disable User

Selecting this radio button disables the user when the password expires. Disabled users can be enabled by using:

- The Enable/Disable User task in the User Console. (The default System Manager, Organization Manager, and Security Manager roles include the Enable/Disable User task.)
- The CA SiteMinder administrative user interface.

  Note: For more information, see the *CA CA SiteMinder Policy Server Administration Guide*.

Force Password Change

Selecting this radio button forces a password to change when the user next attempts to log in.

Issue expiration warnings for *<number>* days

Enter the number of days in advance a user is notified that a password is due to expire.

Password Expires from Inactivity Settings

The Password Expires from Inactivity settings let you specify the time between the user log-in attempts. After this time elapses, a user account is considered inactive. You can also use this section to specify an action when a user whose account is considered inactive has permission to log in.

To configure settings in the Password Expires from Inactivity section, be sure to enable the Track login details check boxes.

Note: This setting requires additional configuration. See Enable Additional Password Policies (see page 67).

The Password Expires from Inactivity section contains the following settings:

- After *<number>* Days--Determines the number of days of inactivity after which a password expires.
- Disable User--Disables the user when the password d expires due to inactivity, the user account is disabled. Disabled users must then be enabled using the Enable/Disable Users task.
- Force Password Change--Forces a password change when a password expires due to inactivity. The user changes the password at the next log-in attempt.

Incorrect Password Settings

In the Incorrect Password settings section, you can specify how many failed logins are allowed before disabling the user account. You can also specify how long the account is disabled before a user can attempt to log in again. This section applies only when you have selected the Track Failed Logins check box.

Note: This setting requires additional configuration. See Enable Additional Password Policies (see page 67).

The Incorrect Password section contains the following fields:

Account disabled after *<number>* successive incorrect passwords

This setting determines the number of consecutive failed log-in attempts a user can make. Limiting the number of unsuccessful attempts protects against programs that are designed to access a resource by repeatedly trying passwords until the correct one is found. If a user fails to log in correctly after the specified number of attempts, Identity Management disables the account. An administrator is required to reenable the account.

After *<number>* minutes

This setting determines the length of time that a user waits before making another login attempt or their account is reenabled. If the user enters another incorrect password, Identity Management disables the account again. The user waits the specified amount of time before trying again.

Allow one login attempt

This setting specifies the number of minutes after a user enters an incorrect password before one additional log-in attempt.

Re-enable account

This setting reenables an account after the specified number of minutes.

Configure Password Composition

You can specify rules that determine the character composition of newly created passwords. Be sure to consider the maximum password length when determining values for character requirements. If the total number of letters and numbers exceeds the maximum password length, all passwords are rejected. For example, if Letters and Digits are both set to six, all passwords contain at least 12 characters (6 letters and 6 digits). In this example, if a maximum password length is eight characters, all passwords are rejected.

Password composition settings include:

Minimum password length

Specifies a minimum length for user passwords.

Maximum password length

Specifies the maximum length for user passwords.

Maximum repeating characters

Determines the maximum number of identical characters that can appear consecutively in a password.

For example, if this value is set to 3, then “aaaa” cannot appear anywhere in the password. However, “aaa” is acceptable within a password. Set this value to ensure that users cannot enter passwords of a single character.

Upper case letters

Specifies whether to allow upper case alphabetic characters and, if so, the minimum number a password must contain.

Lower case letters

Specifies whether to allow lower case alphabetic characters and, if so, the minimum number a password must contain.

Letters

Specifies whether to allow letters and if so, the minimum number a password must contain.

Note: The Letters check box is automatically selected when you allow upper or lower case letters.

Digits

Specifies whether to allow numbers and, if so, the minimum number a password must contain.

Letters and Digits

Specifies whether to allow letters and digits, and if so, the minimum number a password must contain. If this setting is set in conjunction with Digits, characters can satisfy both requirements. For example, if this setting and Digits are set to 4, the password “1234” is a valid password.

Note: The Letters and Digits check box is automatically selected when you allow upper or lower case letters, or numbers.

Punctuation

Specifies whether to allow punctuation marks, and if so, the minimum number a password can contain. Punctuation marks can be periods, commas, exclamation marks, slashes, dashes, and hyphens.

Non-printable

Specifies whether to allow non-printable characters, and if so, the minimum number a password can contain. These characters cannot be displayed on a computer screen.

Note: Certain browsers do not support non-printable characters.

Non-alphanumeric

Specifies whether to allow non-alphanumeric characters such as punctuation marks and other symbols (“@”, “$”, and “*”) and if so, the minimum number a password can contain. Non-printable characters are also included. A non-alphanumeric character also satisfies Punctuation and Non-printable character requirements.

Specify Regular Expressions

Password regular expressions let you specify regular expressions text patterns for string matching that each password matches, or does not match, to be valid. This test can be useful, for example, when you want to require that the first character is a digit, and the last character is not.

You configure multiple expressions for a single password policy. If you create multiple expressions, acceptable passwords match *all* specified expressions.

Follow these steps:

1. Type a descriptive tag for the expression (no white space) in the Name field.
2. Type a regular expression using the syntax described in Regular Expressions Syntax in the Must Match field.
3. If the password does not match the regular expression, select the check box in the Must Not Match column.

Note: You can specify multiple expressions by clicking the plus (+) sign to add the expression.

Example: The following regular expression definition can be used to require that all passwords start with an upper or lower case letter:Name: MustStartAlpha

Expression: [a-zA-Z].*

Regular Expressions Syntax

This section describes the syntax you use to construct regular expressions for password matching. This syntax is consistent with the regular expression syntax supported for resource matching when specifying realms.

Characters	Results
\	Used to quote a meta-character (like '*')
\\	Matches a single '\' character
(A)	Groups subexpressions (affects order of pattern evaluation)
[abc]	Simple character class (any character within brackets matches the target character)
[a-zA-Z]	Character class with ranges (any character range within the brackets matches the target character)
[^abc]	Negated character class
.	Matches any character other than newline
^	Matches only at the beginning of a line
$	Matches only at the end of a line
A*	Matches A 0 or more times (greedy)

Characters	Results
A+	Matches A 1 or more times (greedy)
A?	Matches A 1 or 0 times (greedy)
A*?	Matches A 0 or more times (reluctant)
A+?	Matches A 1 or more times (reluctant)
A??	Matches A 0 or 1 times (reluctant)
AB	Matches A followed by B
A|B	Matches either A or B
\1	Backreference to 1st parenthesized subexpression
n	Backreference to *n*th parenthesized subexpression

All closure operators (+, *, ?) are greedy by default, meaning that they match as many elements of the string as possible without causing the overall match to fail. If you want a closure to be reluctant (non-greedy), you can simply follow it with a '?'. A reluctant closure will match as few elements of the string as possible when finding matches.

Set Password Restrictions

You can place restrictions on password usage. The restrictions include how long a user must wait before reusing a password and how different the password must be from ones previously selected. You can also prevent users from specifying words that you determine are a security risk or contain personal information.

Note: This setting requires additional configuration. See Enable Additional Password Policies (see page 67).

The Restriction section includes the following fields:

Minimum number of days before reuse

Determines how many days a user must wait before reusing a password.

Minimum number of passwords before reuse

Determines how many passwords must be used before a password can be reused.

Note: If you specify a length of time and number of passwords, both criteria are satisfied before a password can be reused. For example, you can configure a password policy which requires users to wait 365 days and specify 12 passwords before reusing a password. After a year, if only six passwords have been used, another six are used before the user can reuse the first password.

Percent different from last password

Specifies the percentage of characters a new password is required to contain. You can set the value to 100. In this case, the new password cannot contain characters that were in the previous password.

Ignore sequence when checking for differences

Ignores the position of the characters in the password when determining the percentage.

For example, with an initial password is BASEBALL12 and the Ignore sequence when checking for differences check box is selected, 12BASEBALL is not acceptable. With the check box deselected, 12BASEBALL is an acceptable password because each letter occurs in a different position.

For increased security, Ignore sequence when checking for differences check box is selected.

Passwords	Percent different	Ignore sequence	Accepted
BASEBALL12 (Old) 12BASEBALL	0	Selected Deselected	Y Y
BASEBALL12 (Old) 12BASEBALL	100	Selected Deselected	N Y
BASEBALL12 (Old) 12SOFTBALL	0	Selected Deselected	Y Y
BASEBALL12 (Old) 12SOFTBALL	90	Selected Deselected	N Y
BASEBALL12 (Old) 12SOFTBALL	100	Selected Deselected	N N

Profile Attributes

Configuring the Match Length field prevents users from using personal information in their passwords. The Match Length field determines the minimum sequence length the password policy compares to attributes in the directory entry. For example, if this value is set to four, Identity Management verifies that the password does not include the last four characters of the user profile attributes, for example, last name or telephone number.

Dictionary

Specifies a list of strings that cannot be used in passwords.

Note: A carriage return follows The last line of the dictionary entry.

The Dictionary settings include the following fields:

- Path--Contains the full path and name of the dictionary file.
- Match Length--Controls the length of strings that are compared against values in the dictionary file. The comparison ignores the case of the strings. You can leave the Match Length field blank or can set it to zero. In these cases, Identity Management only rejects passwords that match a string in the dictionary exactly. When the match length is greater than zero, Identity Management rejects entries during the following conditions:
  - The password includes a substring which starts with the same series of characters as a dictionary entry.
  - The number of consecutive matching characters is greater than or equal to the number specified in the Match Length field.

  For example, consider a dictionary file that contains the following entries:
  - lion
  - tiger
  - bear

  When the Match Length field is set to four results in the following actions:

  "TeddyBear", rejected because Bear matches the bear entry in the dictionary file.

  "prestige", rejected because "tige" matches the first four characters of the tiger entry in the dictionary file.

  "Geiger Counter", accepted since "iger" does not include the first letter of the tiger entry in the dictionary file.

Configure Advanced Password Options

Advanced password policy options let you configure preprocessing of submitted passwords before validation and storage. You can also assign to the policy a priority to allow predictable evaluation of multiple password policies that apply to the same user directory or namespace.

Do Not Force Case | Force Upper Case | Force Lower Case

Determine whether passwords are forced to upper or lower case before processing and storage. Choose a case forcing option by clicking the Force Upper Case or Force Lower Case radio button. Otherwise, be sure that the Do Not Force Case radio button (the default) is selected.

Important! Be sure that any case forcing option that you specify is consistent with any case-related composition requirements you have defined.

Remove Leading White Space

Select to remove leading white space from passwords before processing.

Remove Trailing White Space

Select to remove trailing white space from passwords before processing.

Remove Embedded White Space

Select to remove all embedded white space before processing.

Note: Some user directory implementations automatically strip leading or trailing white space from attribute values (in which user passwords are stored) before storing them. The settings that you specify in your password policy have no effect.

Evaluation Priority

Specifies the evaluation priority for the password policy. The value is in the range 0 (the default) to 999. Applicable policies are evaluated in descending order (999 first; 0 last).

Apply Lower Priority Password Policies

Determines whether lower priority password policies are applied after this one.

Manage Password Policies

Administrators with the appropriate privileges can manage password policies using the View, Modify, Create, and Delete Password Policy tasks. By default, these tasks appear in the Policies category.

When you access one of these tasks, Identity Management displays a list of password policies that apply to the user store associated with the current Identity Management environment. If Identity Management integrates with SiteMinder, the list may include password policies that are created in the SiteMinder Administrative User Interface using Password Services. You can manage password policies that are created in Identity Management or SiteMinder.

Password Policies and Relational Databases

If you configure a password policy that applies to a relational database, you must use the following format to configure the Password Data attribute for the SiteMinder User Directory:

tablename.columnname

To avoid syntax problems during execution, we recommend that this field reside in the primary table.

Identity Management and CA SiteMinder Integration Password Criteria

When Identity Management is integrated with CA SiteMinder and uses CA SiteMinder's password handling capability, password policies are obtained from the CA SiteMinder Policy Store. In this case, construct passwords that meet CA SiteMinder's password criteria. The following punctuation characters are the only punctuation characters that meet <stmdr>'s password criteria:

'*', '(', '\',',','@','"',':','#','_','-','!','&','?',')','(','{','}','*','.','/'

Important! Identity Management does not impose any restriction on the use of punctuation characters in passwords. However, if you intend to use CA SiteMinderpassword capability, we recommended that you construct passwords that meet CA SiteMinder's restrictions.

Reset Password or Unlock Account

In the case users forget their passwords on Windows systems, you can configure Self-Service to prompt the user from the Windows logon screen. You can use this feature by installing the Credential Provider, for Windows VISTA and Windows 7 systems

With this feature, the user is logged into Self Service through the Cube web browser where a password change request page appears. After filling in this page, the user clicks Return to go back to the Windows Logon screen.

Install the Credential Provider

Follow these steps:

1. Locate the Identity Management Provisioning Components download or other installation media.
2. Run the installer under Agent.

   Note: For the Credential Provider on a 64-bit operating system, be sure to choose the 64-bit version of this software.
3. Follow the wizard prompts to answer the questions.
4. If you installed the Credential Provider on a 64-bit operating system, download Microsoft Visual C++ 2008 SP1 (64-bit).
5. Once the installation completes, Configure the Credential Provider.

Configure the Credential Provider

You can use a configuration tool to configure a system where you installed the Credential Provider.

To configure the Credential Provider

1. In Windows Explorer, go to the directory where you installed the Credential Provider. For example:

   C:\Program Files\CA\Identity Manager\Credential Provider
2. Double-click the following executable:

   CAIMCredProvConfig.exe

3. Select the first credential provider as the default.

   The logon screen may not honor this setting if a second credential provider is in use, such as the Microsoft password credential provider. If both providers attempt to be the default provider, the logon screen chooses a default provider.

4. Disable the default credential provider.

5. Fill in the Credential Provider Settings fields as follows:

   Link1 URL

   The URL used when a user clicks on the Forgot Password link. This link should be a URL to a web interface for password resetting.

   The following is a sample link:

   ```
   http://eastern.local:8080/iam/im/environment/ca12/index.jsp?
   task.tag=forgottenpassword&facesViewId=/app/page/screen/
   fp_identify_user.jsp&action.forgottenpassword.identify=1&USER_ID=%usernam
   e%
   ```

   For this URL, self registration must be working on the environment. Also, verify the Self Service URL for the Identity Management environment works from the system where you are installing the Credential Provider. Occurrences of %username% are replaced by the value in the username field on the Logon dialog.

   Link2 **URL**

   The URL used when a user clicks on the Unlock Account link. This link should be a URL to a web interface allowing a user to unlock an account. Occurrences of %username% are replaced by the value in the username field on the Logon dialog.

   Link3 URL

   The URL used when a user clicks on the New Account link. This link should be a URL to a web interface allowing a user to create an account. The %username% tag is not expected to be part of the URL

Use Custom Title

A customized string replaces the “Powered by …” string that appears on the title bar, or in the Return dialog of the Credential Provider. The location of the string is based on the Section 508 Compliance setting.

Domain

The Provisioning domain name.

Section 508 Compliance (Use Return in menu)

Enables the Return function in a menu. If unchecked, the Return dialog is used.

Disable All Dialogs

Prevents the Secure Browser from spawning the new dialog windows, such as pop-ups, errors, and print or save dialogs. *Disable All Dialogs* is enabled to improve the system security, but can be disabled for troubleshooting purposes.

6. Fill in the Secure Browser Settings fields as follows:

   Allow List

   A regular expression pattern matching URLs to which access should always be allowed.

   Deny List

   A regular expression pattern matching URLs to which access should always be denied.

7. (Optional) Click Export to export your settings to another system.
8. Click OK to save your settings.
9. Restart the system.

Credential Provider Registry Settings

If you choose not to use the Credential Provider configuration tool, you can edit the Windows registry settings in the following key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\CA\CAIMCredentialProvider]
```

link1_cmd

This link should be the URL to navigate to when a user clicks link 1.

link2_cmd

This link should be the URL to navigate to when a user clicks link 2. For example, you could add a link that would lead to a website for unlocking accounts.

If the link2_cmd is blank, only the link1_cmd appears in the Logon Dialog Window.

link3_cmd

This link should load a URL to a web interface allowing a user to create an account.

comp508

Enables the Return function in a menu. If unchecked, the Return dialog is used

domain

The Provisioning domain name.

langdir

The location of the localized language DLLs.

disablepwdcp

The Disable Microsoft Password Credential Provider option. 1 is disabled. 0 is enabled.

CredentialProviderInstallPath

The full directory path to where the Credential Provider is installed.

configdir

The full directory path to where the Credential Provider is installed.

selectdefaultcredential

Select the first credential provider as the default option. Yes is enabled. No is disabled.

Cube Browser Registry Settings

The Cube secure browser component has several registry values which control its behavior. These settings are in the following Registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\CA\Cube]
```

Type

```
REG_SZ(String)
```

404

The path to a standard HTML document to be displayed if the machine cannot contact the Identity Management at startup.

default

The default page to navigate to when no URL is included in Link1 Command or Link2 Command.

allow

Explicit Allow ACL. A regular expression for pattern matching URLs that is always allowed. For more information, see Cube Access Control Lists (see page 85).

close

Closes the secure browser and returns the user to the Credential Provider forgotten password dialog.

deny

Explicit Deny ACL. A regular expression for pattern matching URLs that should always be denied access. For more information, see Cube Access Control Lists (see page 85).

langdir

The location of the localized language DLLs.

rejectinvalidcerts

Controls whether the Credential Provider accepts only valid SSL certificates. When set to no, this option permits expired or invalid SSL certificates.

Valid values for this key are *yes* and *no.*

unreachable

Redirects to a URL when the Cube encounters connectivity issues.

Sample Value: file:///C:\unreachable.html

usecustomtitle

This enables the custom title for the Credential Provider.

customtitle

This a title you want to appear in the Credential Provider.

Cube Access Control Lists

Cube ACLs are regular expression patterns that explicitly allow or deny permission to navigate to a selected URL. ACLs evaluate in the following order:

1. Allow (Permission is automatically allowed first)
2. Deny (Denied URLs are checked second)

Access Control List Examples

"allow"="(.pdf)"

Allow all PDF documents to be displayed.

"deny"="(.doc|.xls)"

Deny access to Microsoft Word and Excel documents.

Customize the Powered by Message

You may notice a "Powered by..." message in the Return dialog or the Return menu option of the Credential Provider. You can edit or remove this message.

To customize the Powered By message

1. Download ResEdit, a freeware resource editor from http://www.resedit.net.
2. Start ResEdit.
3. Edit the file 1033.dll in the languages folder.
4. Double-click String Table.
5. Remove or modify resource ID 135, the English version of the resource for this message.

Reset a Password for a Windows Login

After the Credential Provider is installed on a Windows system, a Forgot Password link appears on the standard Microsoft Windows logon dialog. Use this link to reset your password or see clues to help you remember it.

To reset a password for a Windows login

1. Click Login from the Windows Security dialog. The Windows Login dialog appears.
2. Enter a valid user name.
3. Click Forgot Password.

   The Identity Management Password Clue page appears.

   If you remember your password, return to the login dialog to continue. Otherwise, perform step 4 to authenticate to Identity Management Self Service.
4. Type the answers to the authentication questions.

   Note: If you do not know the answers to all questions, click Request so that your password can be reset by an administrator.

   You are then prompted to change your password on the next screen.

Credential Provider Silent Install

The Credential Provider supports a silent mode of installation. Six properties are supported

LINK1

Refers to the SOFTWARE\CA\CAIMCredentialProvider\link1_cmd in the registry.

LINK2

Refers to the SOFTWARE\CA\CAIMCredentialProvider\link2_cmd in the registry.

LINK3

Refers to the SOFTWARE\CA\CAIMCredentialProvider\link3_cmd in the registry.

DOMAIN

Refers to the SOFTWARE\CA\CAIMCredentialProvider\domain in the registry.

COMP508

Refers to the SOFTWARE\CA\CAIMCredentialProvider\comp508 in the registry.

USECUSTOMTITLE

Refers to the SOFTWARE\CA\Cube\usecustomtitle in the registry.

CUSTOMTITLE

refers to the SOFTWARE\CA\Cube\customtitle in the registry.

REJECTINVALIDCERTS

refers to the SOFTWARE\ComputerAssociates\Cube\rejectinvalidcerts in the registry.

UNREACHABLE

refers to the location of the unreachable page.

The syntax to set the value of these properties follows:

```
setup /s /v"/qn LICENSE=Yes INSTALLDIR=\"C:\Program Files\CA\Identity
Manager\Credential Provider\" LINK1=\"<url>\" LINK2=\"<url>\"LINK3=\"<url>\"
COMP508=\"yes\" REJECTINVALIDCERTS=\"yes\" USECUSTOMTITLE=\"yes\"
CUSTOMTITLE=\"custom cp title\""
```

or

```
setup /s /v"/qn LICENSE=Yes INSTALLDIR=\"C:\Program Files\CA\Identity
Manager\Credential Provider\" LINK1=\"<url>\" LINK2=\"<url>\" LINK3=\"<url>\"
COMP508=\"yes\" USECUSTOMTITLE=\"yes\" CUSTOMTITLE=\"custom cp title\"
SELECTDEFAULTCREDENTIAL=\"yes\" UNREACHABLE=\"<url>\""
```

or

```
setup /s /v"/qn LICENSE=Yes INSTALLDIR=\"C:\Program Files\CA\Identity
Manager\Credential Provider\"
LINK1=\"<url>\" LINK2=\" <url>\" LINK3=\"<url>\" COMP508=\"yes\"
USECUSTOMTITLE=\"yes\" CUSTOMTITLE=\"custom cp title\"
SELECTDEFAULTCREDENTIAL=\"yes\" UNREACHABLE=\"file:///[INSTALLDIR]<file
name>\"CUBE_ALLOW=\"\"CUBE_DENY=\"\""
```

[INSTALLDIR]

Refers to the value of the INSTALLDIR property.

<url>

Specifies the URL for a unlock account or a forget password.

<file name>

Defines the name of the unreachable file name.

CUBE_ALLOW

Refers to allowing the URL invocation from cube.

CUBE_DENY

Refers to restricting the URL invocation from cube.

Synchronizing Passwords on Endpoints

You can install a password synchronization agent on certain endpoints supported by Identity Management. The agent intercepts password change requests on the endpoint and submits the changes to the Provisioning Server.

Password Synchronization on Windows

Identity Management can intercept the password change of a native Windows account and propagate the new password to a user and all accounts belonging to that user.

When the Password Synchronization Agent detects a password change attempt, the agent intercepts the request and sends it to the Provisioning Server. The Provisioning Server then propagates the new password to the user and other accounts associated with that user.

Password synchronization has the following requirements:

- The Password Synchronization Agent must be installed on the system on which password changes are intercepted.
- The system must be managed as an acquired endpoint.
- The Password Synchronization Agent installed check box must be selected on the acquired Endpoint Settings tab.
- The accounts on the managed systems must be explored and correlated to Identity Management users.
- The environment must allow password changes to come from endpoint accounts. An administrator with access to the Management Console enables this feature.

Important! Use care in formulating password rules, so that one password applies to all systems. For example, if Windows passwords must be 12 characters, any system that accepts passwords only up to 10 characters will reject the change during synchronization.

The Identity Management Server is not aware of the password restrictions on the endpoint. When working with endpoint accounts, the password policy should be stricter than the password policy of the endpoints.

Install the Windows Password Synchronization Agent

You can install the Password Synchronization Agent on any managed Windows computer where global users log on. The Agent runs in the background on these machines.

Run the Installation Program

Note the following requirements:

- The Provisioning Server must manage the system on which you are installing the Agent.
- Create a user to act as the Administrator for password changes: suggested name is etapwsad. This user must have the PasswordAdministrator profile.
- Two Windows Password Synchronization Agents exist in the installation media: one for 32-bit Windows and one for 64-bit. The 32-bit Password Synchronization Agent is not supported on 64-bit Windows. FIPS is only supported by the 32-bit Password Synchronization Agent.

Follow these steps:

1. Locate the Identity Management installation media.
2. Browse to \Agent\PasswordSync or \Agent\PasswordSync-x64.
3. Run setup.exe.
4. Respond to the Configuration Wizard as follows:
   a. In the Host name field, enter the name of the Provisioning Server system.
   b. Change the port as required if your Provisioning Server installation uses a non-default port.

      The suggested LDAP port that is used to connect to the Provisioning Server is 20390.
   c. Click the Find domain button to retrieve the Provisioning Server Domain.
   d. If your Provisioning Server installation is configured for failover follow the on-screen instructions to add a comma separated list of servers.
   e. Click Next.
   f. In the Administrator field, enter etapwsad as the default global user name for the Password Synchronization Agent. This user must have the PasswordAdministrator profile. It does not exist by default.
   g. In the Password Administrator field, enter the password of the Administrator.
   h. Click Next.
   i. From the Endpoint Type drop-down list, select the Endpoint Type of the host on which you are installing the Agent.
   j. From the Endpoint Name drop-down list, select the Endpoint name that was used when creating the endpoint in the User Console.
   k. Click Configure.
5. Click Finish when prompted to complete the installation and reboot.

Update the Endpoint in the User Console

In the User Console, update the endpoint to indicate that the agent is installed.

Follow these steps:

1. Log in to the User Console.
2. Search for the endpoint with the agent installed.
3. Click the Endpoint Settings tab.
4. Select the Password Synchronization Agent Installed check box.

Enable an Environment for Password Synchronization

After you install the Password Synchronization Agent, you enable the environment to receive password changes that are made on the endpoints. For this task, an administrator needs access to the Management Console and CA Directory to enable the environment to accept these changes.

Follow these steps:

1. For new users, you use the Management Console as follows:
   a. Select the environment.
   b. Click Advanced Settings, Provisioning.
   c. Select the Enable Password Changes from Endpoint Accounts check box.
2. For existing users, set the eTPropagatePassword attribute to 1 in CA Directory.

Configure the Agent for Alternate Servers

To configure the Password Synchronization Agent to use an alternate server, you use the Password Synchronization Agent Configuration wizard.

To configure an alternate server for the Agent

1. Run PwdSyncConfig.exe located in *password_sync_folder*\bin.
2. Enter the following configuration information:

   Host

   Specify the name of the Provisioning Server system.

   This populates the Server URL field with the host name you specify.

   LDAP port

   Specify the LDAP port used to connect to the Provisioning Server is 20390. Change this port as required if your Provisioning Server installation uses a non-default port.

3. Click the Find domain button to retrieve the Provisioning Server Domain.

4. Add the host name and port of the alternate servers in the Server URLs field using the following format:

   ldaps://*primaryhost*:20390,ldaps://*alternatehost1*:20390

5. Click Next.
6. Complete the remaining fields in the configuration wizard.

How the Password Synchronization Agent Works

The propagation process begins when a user's password is changed on a Windows system using any method. After the password is entered, the following occurs:

1. The Windows operating system checks to make sure the password meets its password policy. If Windows does not accept the password, the change request is rejected, an error message appears, and no further action, including synchronization, is taken.
2. The Windows system passes the password change request to the Password Synchronization agent, which, if configured for password quality checking, submits the password to the Provisioning Server for password quality checking. If the password does not meet the Identity Management quality rules, the change request is rejected and an error message displays. The Windows password remains unchanged and no synchronization takes place.
3. A password that meets the quality rules of both Windows and Identity Management is submitted by the Password Synchronization Agent to the Provisioning Server for propagation.
4. Identity Management updates the global user password and propagates the new password accounts associated with the global user.

Note: Your password policies for Windows and Identity Management must be identical or consistent, because the error messages displayed are based on the Windows password policy, even if Identity Management rejects the request.

The password_update_timeout configuration parameter (eta_pwdsync.conf) specifies how long (in seconds) the PSA waits for the password-change-propagation confirmation from Identity Management. If the PSA does not receive a confirmation during that time, it proceeds as if the propagation succeeded and logs a warning (eta_pwdsync.log) that password change propagation could not be verified. The minimum value for the parameter is zero (0), which means that the PSA will not wait for confirmation.

Account-Level Password Quality Checking

Password quality checking is performed when accounts on managed endpoints are created or modified or when Identity Management user passwords are set. Password quality checking on accounts is limited to checks based on the characters in the password. Checks of global user passwords that are based on the history of recent changes (frequency of password update and frequency of password reuse) are not performed on accounts because Identity Management cannot intercept all password changes for account passwords. Therefore, it cannot have an accurate password change history with which to perform these checks.

The checking of account passwords is controlled by the following domain configuration parameters:

- Endpoint/Check Account Passwords
- Endpoint/Check Empty Account Passwords

The value for each parameter specifies for each managed endpoint the level of checking that should be performed. The endpoint can be specified in the following ways:

```
ALL
-ALL
<NamespaceName>
-<NamespaceName>
<NamespaceName>:<DirectoryName>
-<NamespaceName>:<DirectoryName>
```

The forms that include a minus (-) sign, disable the parameter. The forms without it enable the parameter. The [-]<NamespaceName> forms control all endpoints of the indicated endpoint type, while the [-]<NamespaceName>:<DirectoryName> forms control individual endpoints. The [-]ALL forms control all endpoints of all endpoint types. The default value for both parameters is -ALL.

Each of these parameters can be specified many times. If multiple values specify the same endpoint, the last value is used. You can place general rules first and specific rules later to override the general rule.

The Check Account Passwords parameter provides checking equivalent to global user password quality checking. With this parameter enabled for an endpoint, Identity Management checks any password in a requested change for an existing account, including attempts to set an empty password. During account creation, if no password is provided, password quality checking is not performed.

Check Empty Account Passwords provides the added checking of empty passwords when creating accounts. If the password profile is enabled and requires at least a single-character password, an empty password causes account creation to fail. This parameter is separate from Check Account Passwords because in some endpoint types it is acceptable to create an account with no password.

Note: Account password quality checking is skipped for synchronized account passwords if the supplied password matches the current global user password.

Password Quality Enforcement

The Password Synchronization option intercepts password changes requests on native systems (for example, Windows NT/ADS) and submits them to Identity Management. Identity Management synchronizes the global user password and account passwords associated with the global user. Both Identity Management password quality rules for a password profile and native system password quality rules (Windows NT/ADS) can be used to enforce password quality control.

Configure Password Synchronization

The Password Synchronization Agent is initially configured during installation and can be reconfigured at any time using the Password Synchronization configuration wizard. Further configuration is possible. For example, you can change settings for password quality checking or modifying timeouts, using the eta_pwdsync.conf file.

This file is located in the password_sync_folder\data\ folder. All keys in this configuration file are set during the installation of the Password Synchronization Agent. Therefore, change these keys only if necessary. See the text in this file for more information.

Important! As a precaution, create a backup of the configuration file before editing it.

[Server] Section

Key	Description	Default
host	Specifies the domain server that manages password propagation.	None
port	Specifies the LDAP listening port of the Provisioning Server.	20411
use_tls	Specifies whether TLS/SSL is used to secure communication between the Password Synchronization Agent and the Provisioning Server.	Yes

Key	Description	Default
admin_suffix	Specifies the domain suffix of the administrative user that the Password Synchronization Agent uses to log in to Identity Management.	None
admin	Specifies the account name of the administrative user that the Password Synchronization Agent uses to log in to Identity Management.	None
password	Specifies the password for the account name specified in the admin key.	None

[eTaDomain] Section

Key	Description	Default
Domain	Specifies the Provisioning domain where you installed the Password Synchronization Agent.	None
etrust_suffix	Specifies the suffix for the entire Identity Management product.	None
domain_suffix	Specifies the domain suffix for the Provisioning domain.	None
endpoint type	Specifies the endpoint type where you installed the Password Synchronization Agent.	None
endpoint	Specifies the endpoint for which the Password Synchronization Agent intercepts passwords.	None
endpoint_dn	Specifies the Distinguished Name of the endpoint.	None
container_dn	Specifies the Distinguished Name of the container that contains the accounts whose passwords are being changed.	None
acct_attribute_name	Specifies the attribute name of the account, for example, eTN16AccountName for Windows NT.	Depends on the endpoint type

Key	Description	Default
acct_object_class	Specifies the objectClass of the accounts.	Depends on the endpoint type

[PasswordProfile] Section

Key	Description	Default
profile_enabled	Specifies whether the password profile checking feature is enabled.	No
profile_dn	Specifies whether the Password Configuration Wizard generates a DN for the password profile.	eTPasswordProfileName=Password Profile,eTPasswordProfileContainerName=Password Profile,eTNamespaceName=CommonObjects,dc=cai,dc=eta

[Timeout] Section

Key	Description	Default
search_acct_dn	Specifies the timeout value when searching for the account DN.	120 seconds
pwd_update	Specifies the timeout value when propagating passwords.	400 seconds
pwd_quality_check	Specifies the timeout value (in seconds) when performing password quality checking.	1

[Logs] Section

Key	Description	Default
log_file	Specifies the log file that contains logged messages from the Password Synchronization Agent.	..\Program files\CA\Identity Manager Password Sync Agent

Key	Description	Default
log_level	Specifies the level of logging. Valid values are: 1--Init file 2--Password update success or failure 3--Connection debugging 4--Tracing	0, for no logging

Failover

If the Provisioning Server is down or it is heavily loaded, the Password Synchronization Agent can fail over to another server. Failover requires that multiple Provisioning Servers serve the same domain and the Agent uses those servers.

The section configuring the agent to use alternate servers (see page 91) provides the configuration instructions.

Enable Log Messages

To discover why a password modification was rejected, view the Password Synchronization Agent logs. All logged messages are stored in the eta_pwdsync.log file. By default, this file is located in the password_sync_folder\Logs folder.

Password Synchronization Agent logging can contain the following:

- Error messages, which are always logged.
- Diagnostic (process flow, trace) messages, which can be enabled or disabled based on the value of the logging_enabled=yes|no parameter in the eta_pwdsync.conf file.

For additional information, review the eta_pwdsync.log and the Provisioning Server logs for the same time period.

The previous log_level configuration parameter has been deprecated but left for backward compatibility: log_level=0 translates into logging_enabled=no and log_level=anything else translates into logging_enabled=yes. If both old and new parameters are present in the configuration file, the explicit setting of logging_enabled=yes|no parameter overrides the indirect setting performed through the old log_level=number.

Verify the Installation

After the Password Synchronization Agent installation is complete, change a password on the Windows system to verify that the global user password associated with the account is changed also.

Password Synchronization on UNIX and Linux

Identity Management can intercept an account's password change on a UNIX or Linux system, and propagate it to all other accounts associated with its Global User. The component used to authenticate passwords against external security systems is called Pluggable Authentication Module (PAM). With PAM, Identity Management authenticates passwords against external security systems so that global users can use their existing system passwords to log on to Identity Management.

UNIX Password Synchronization

A password synchronization module is provided that detects password change events through the UNIX PAM framework. The UNIX Password Synchronization module notifies the Provisioning Server of a password change. The Provisioning Server finds the associated Global User, and propagates changes to other related accounts automatically.

The UNIX operating systems that support the PAM framework include:

- AIX v5.3 on Power platform with PAM enabled
- HP-UX v11.00 on a PA-RISC platform, and Itanium® 2 platforms
- Solaris v2.6 and higher on Sparc and Intel platforms
- 32-bit Linux with glibc v2.2 and higher on s390 or Intel i386 platform

  Note: For Linux platforms, the test_sync binary must be on the PATH for all users, but only the root user, the owner, should have execute permission.

  To add this library to the path for all users, include this command in the global /etc/bashrc file:

  ```
  export PATH=$PATH:/etc/pam_CA_eta
  ```

How UNIX PAM Works

The following process describes the UNIX PAM feature's functions:

1. A UNIX user's password is to be changed for one of the following reasons:
   - Decision of the user.
   - The user is forced to change the password by system settings or manual intervention.
   - The user's password is changed by an administrator.
2. The new password is submitted to the PAM framework password service.
3. The PAM framework's password service invokes the PAM library to update the local UNIX security files.
4. The PAM framework's password service invokes the UNIX password synchronization module (pam_CA_eta) to notify the Provisioning Server of the password change.
5. The Provisioning Server updates the password of the associated Global User and all accounts associated with the Global User.

Requirements for Using UNIX Password Synchronization

The requirements for using the UNIX Password Synchronization feature are the following:

- The UNIX Password Synchronization agent must be installed on the UNIX system on which you want to detect password changes.
- The UNIX Remote agent and CAM must be installed on the UNIX system on which the UNIX Password Synchronization agent resides.
- The system must be managed as an acquired endpoint. The Password Synchronization agent is installed check box must be selected on the acquired endpoint's properties.
- The accounts on the managed systems must be explored and correlated to global users.
- The environment must allow password changes to come from endpoint accounts. An administrator with access to the Management Console enables this feature.

Install the UNIX PAM Feature

Perform the following procedure to install UNIX PAM.

To install the UNIX PAM feature

1. Select the package file that corresponds to your UNIX platform:

UNIX Operating System	Package File Name
HP-UX v11 PA-RISC	pam_CA_eta-1.1.HPUX.tar.Z

HP-UX Itanium2	pam_CA_eta1.1HPUX-IA64.tar.Z
AIX v5.3 Power	pam_CA_eta-1.1.AIX.tar.Z
Solaris Sparc	pam_CA_eta-1.1.Solaris.tar.Z
Solaris Intel	pam_CA_eta-1.1.SolarisIntel.tar.Z
Linux x86	pam_CA_eta-1.1.Linux.tar.gz
Linux s390	pam_CA_eta-1.1.LinuxS390.tar.gz

2. Transfer the chosen package file to a temporary folder (/tmp) on the UNIX server using FTP in binary mode, or any other file transfer tool that supports binary files. A sample transfer session might appear as follows:

```
W:\Pam>ftp user01
Connected to user01.company.com.
220 user01 FTP server (Version 1.2.3.4) ready.
User (user01.company.com:(none)): root
331 Password required for root.
Password:
230 User root logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> bin
200 Type set to I.
ftp> put pam_CA_eta-1.1.HPUX.tar.Z
200 PORT command successful.
150 Opening BINARY mode data connection for pam_CA_eta-1.1.HPUX.tar.Z.
226 Transfer complete.
ftp: 117562 bytes sent in 0,09Seconds 1306,24Kbytes/sec.
ftp> quit
```

3. Logon as the root user on the UNIX server and extract the package file:

```
# cd /tmp
# zcat pam_CA_eta-1.1.<platform>.tar.Z | tar -xf -
```

On Linux, use the command:

```
# tar -xzf pam_CA_eta-1.1.<platform-hardware>.tar.gz
```

4. Copy the configuration and TLS files to the default configuration folder:

```
# cd pam_CA_eta-1.1
# mv pam_CA_eta /etc
```

5. Copy the pam_CA_eta module to the Security libraries folder:

   On AIX, use the command:

   ```
   # cp -p pam_CA_eta.o /usr/lib/security/
   ```

   On HP-UX, use the command:

   ```
   # cp -p libpam_CA_eta.1 /usr/lib/security/
   ```

   On HP-UX Itanium2, use the command:

   ```
   # cp -p libpam_CA_eta.1 /usr/lib/security/hpux32
   ```

   On Linux i386 or s390, use the command:

   ```
   # cp -p pam_CA_eta.so /lib/security/
   ```

   On Solaris Sparc or Intel, use the command:

   ```
   # cp -p pam_CA_eta.so /usr/lib/security/
   ```

6. (Optional) Copy the Testing programs:

   ```
   # cp -p test_* /etc/pam_CA_eta
   # cp -p pam_test* (/usr)/lib/security/
   ```

More Information

Troubleshooting UNIX Password Synchronization (see page 108)

Update the Endpoint in the User Console

In the User Console, update the endpoint to indicate that the agent is installed.

Follow these steps:

1. Log in to the User Console.
2. Search for the endpoint with the agent installed.
3. Click the Endpoint Settings tab.
4. Select the Password Synchronization Agent Installed check box.

Enable an Environment for Password Synchronization

After you install the Password Synchronization Agent, you enable the environment to receive password changes that are made on the endpoints. For this task, an administrator needs access to the Management Console and CA Directory to enable the environment to accept these changes.

Follow these steps:

1. For new users, you use the Management Console as follows:
   a. Select the environment.
   b. Click Advanced Settings, Provisioning.
   c. Select the Enable Password Changes from Endpoint Accounts check box.
2. For existing users, set the eTPropagatePassword attribute to 1 in CA Directory.

Configuring the UNIX Password Synchronization Feature

Configuration the UNIX Password Synchronization feature involves setting parameters in the following files:

- /etc/pam_CA_eta/pam_CA_eta.conf
- /etc/pam.conf

Important! Because the password of a highly-privileged user is stored in the pam_CA_eta.conf configuration file, that file must be readable only by the root account. Note that the file settings in the package file include owner=root and mode=500 and that the -p switch of the cp command preserves them during installation.

Configure the pam_CA_eta.conf File

Perform the following procedure to configure the pam_CA_eta.conf file.

To configure the pam_CA_eta.conf file

1. Navigate to the /etc/pam_CA_eta folder.
2. Edit the pam_CA_eta.conf file. This configuration file contains its own documentation.

```
#
#   CA - Identity Management
#
#   pam_CA_eta.conf
#
#   Configuration file for the Unix PAM password module "pam_CA_eta"
#

# keyword: server
# description: the Identity Management LDAP server primary and optional alternate
server hostname
# value: a valid hostname and an optional server
# default: no default
server ETA_SERVER ALT_SERVER

#

# keyword: port
# description: the numeric TCP/IP port number of the Identity Management LDAP
server
# value: a valid TCP/IP port number
# default: 20390
# port 20390


# keyword: use-tls
# description: does it use the secured LDAP over TLS protocol ?
# value: yes or no
# default: yes
# use-tls yes
```

```
# keyword: time-limit
# description: the maximum time in seconds to wait for the end of an LDAP operation.
# value: a numeric value of seconds
# default: 300
# time-limit 300

# keyword: remote-server
# description: identifies whether on premise or cloud Identity Manager
#              server is used.
#              Cloud based server is accessed by proxying the requests
#              through the on-premise CS, requiring use of remote-server
#              set to 'yes'.
# value: yes or no
# default: no
# remote-server no

# keyword: size-limit
# description: the maximum number of entries returned by the Identity Management
server
# value: a numeric value
# default: 100
# size-limit 100


# keyword: root
# description: the root DN of the Identity Management server
# value: a valid DN string
# default: dc=eta
# root dc=eta


# keyword: domain
# description: the name of the Identity Management domain
# value: a string
# default: im
# domain    im


# keyword: user
# description: the Identity Management Global User name used to bind to the
Identity Management server
# value: a valid Global User name string
# default: etaadmin
# user etaadmin


# keyword: password
# description: the clear-text password of the "binding" Identity Management Global
User
# value: the password of the above Global User
# default: no default
password SECRET
```

```
# keyword: directory-type
# description: the Identity Management Unix Endpoint type of this Unix server
# value: ETC or NIS
# default: ETC
# endpoint-type ETC


# keyword: endpoint-name
# description: the Identity Management Unix Endpoint name of this Unix server
# value: a valid Unix Endpoint name string
# default:
# ETC: the result of the "hostname" command (ie: gethostname() system call)
# NIS: "domain [hostname]" where "domain" is the result of the "domainname" command
#  (ie: getdomainname() system call) and "hostname" the result of the "hostname"
#    command (ie: gethostname() system call)
# endpoint-name dirname


# keyword: tls-cacert-file
# description: the name of the Identity Management CA certificate file
# value: a valid full path file name
# default: /etc/pam_CA_eta/et2_cacert.pem
# tls-cacert-file /etc/pam_CA_eta/et2_cacert.pem


# keyword: tls-cert-file
# description: the name of the Identity Management client certificate file
# value: a valid full path file name
# default: /etc/pam_CA_eta/eta2_clientcert.pem
# tls-cert-file /etc/pam_CA_eta/eta2_clientcert.pem


# keyword: tls-key-file
# description: the name of the Identity Management client private key file
# value: a valid full path file name
# default: /etc/pam_CA_eta/eta2_clientkey.pem
# tls-key-file /etc/pam_CA_eta/eta2_clientkey.pem


# keyword: tls-random-file
# description: the name of the "pseudo random number generator" seed file
# value: a valid full path file name
# default: /etc/pam_CA_eta/prng_seed
# tls-random-file /etc/pam_CA_eta/prng_seed


# keyword: use-status
# description: this module will exit with a non-zero status code in case of failure.
# value: yes or no
# default: no
# use-status no
```

```
# keyword: verbose
# description: this module will display informational or error messages to the
user.
# value: yes or no
# default: yes
# verbose yes
```

Note: The server, domain and password parameters do not have a default value and need to be updated.

Configure the pam.conf File

The /etc/pam.conf file is the main PAM configuration file. You must edit the file to insert a line in the password service stack. On some Linux systems, the pam.conf file is replaced with /etc/pam.d, so you will need to edit the /etc/pam.d/system-auth file.

To configure the pam.conf file

1. Navigate to the /etc directory, or /etc/pam.d directory if you are configuring the PAM module on an appropriate Linux system.
2. Edit the pam.conf file to insert a Password Synchronization line in the password service stack. For platform-specific configurations, see the examples that follow:

   passwd password required /usr/lib/security/pam_unix.so

   passwd password optional /usr/lib/security/pam_CA_eta.so

3. (Optional) You can add the following optional parameters on the pam_CA_eta module line:

   config=/path/file

   Indicates the location of an alternate configuration file.

   syslog

   Sends error and informational messages to the local syslog service.

   trace

   Generates a trace file for each password update operation. The trace files are named /tmp/pam_CA_eta-trace.<nnnn> where <nnnn> is the PID number of the password process.

4. Implement the following platform-specific configuration changes:

   For AIX systems, add the following lines at the bottom of the /etc/pam.conf file:

```
#
# Identity Management Unix Password Synchronization
#
login   password  optional    /usr/lib/security/pam_CA_eta.so syslog
passwd  password  optional    /usr/lib/security/pam_CA_eta.so syslog
rlogin  password  optional    /usr/lib/security/pam_CA_eta.so syslog
su      password  optional    /usr/lib/security/pam_CA_eta.so syslog
telnet  password  optional    /usr/lib/security/pam_CA_eta.so syslog
sshd    password  optional    /usr/lib/security/pam_CA_eta.so syslog
OTHER   password  optional    /usr/lib/security/pam_CA_eta.so syslog
```

   For HP-UX systems, add the following lines at the bottom of the /etc/pam.conf file:

```
#
# Identity Management Unix Password Synchronization
#
login    password optional    /usr/lib/security/libpam_CA_eta.1 syslog
passwd   password optional    /usr/lib/security/libpam_CA_eta.1 syslog
dtlogin  password optional    /usr/lib/security/libpam_CA_eta.1 syslog
dtaction password optional    /usr/lib/security/libpam_CA_eta.1 syslog
OTHER    password optional    /usr/lib/security/libpam_CA_eta.1 syslog
```

   For HP-UX Itanium2, add the following lines at the bottom of the /etc/pam.conf file:

```
#
# Identity Management Unix Password Synchronization
#
login    password optional    /usr/lib/security/$ISA/libpam_CA_eta.1 syslog
passwd   password optional    /usr/lib/security/$ISA/libpam_CA_eta.1 syslog
dtlogin  password optional    /usr/lib/security/$ISA/libpam_CA_eta.1 syslog
```

```
dtaction password optional    /usr/lib/security/$ISA/libpam_CA_eta.1 syslog
OTHER    password optional    /usr/lib/security/$ISA/libpam_CA_eta.1 syslog
```

For Sun Solaris systems, add the pam_CA_eta line after the existing pam_unix line:

```
#
# Password management
#
other   password required       /usr/lib/security/pam_unix.so.1
other   password optional       /usr/lib/security/pam_CA_eta.so syslog
```

For Linux systems, add the pam_CA_eta line between the existing pam_cracklib and pam_unix lines:

```
password    required     /lib/security/pam_cracklib.so retry=3 type=
password    optional     /lib/security/pam_CA_eta.so syslog
password    sufficient   /lib/security/pam_unix.so nullok use_authtok md5 shadow
password    required     /lib/security/pam_deny.so
```

5. For AIX systems, edit the /etc/security/login.cfg file to set auth_type = PAM_AUTH. This enables the PAM framework, which is not enabled by default. This is a run-time setting so you do not have to reboot the system for it to take effect.

Troubleshooting UNIX Password Synchronization

You can troubleshoot the UNIX PAM feature using syslog and trace messages, and by testing the configuration, LDAP/TLS connection, the password synchronization, and the PAM framework.

More Information

Activating Syslog Messages (see page 108)
Activating Trace Messages (see page 109)

Activating Syslog Messages

Add the syslog parameter to the pam_CA_eta line in the /etc/pam.conf file to let the pam_CA_eta module generate informational and error messages. When the logging option is in use, the UNIX administrator sees information messages in the syslog files each time a UNIX account changes its password. These messages should provide enough information to diagnose basic problems.

You could set this option permanently on production systems as it does not require many more resources than when running in silent service.

Activating Trace Messages

If the syslog messages do not provide enough information, the trace mode can provide more details. For each password update operation, the trace module generates a file named /tmp/pam_CA_eta-trace.<nnnn> (where <nnnn> is the PID of the passwd process) with an entry for most of the function calls used by the module and the data used or returned by those functions.

Even though the trace files are only readable by the root account, they will contain the clear-text new passwords. For this reason, this parameter should not be used permanently on a production system.

Testing the Configuration File

You can use the test_config tool, which is located in the /etc/pam_CA_eta directory, to verify the configuration file. First, you set up the folder structure as follows:

1. Move the pam_CA_eta folder under /etc.
2. Copy everything under pam_CA_eta-1.1 to /etc/pam_CA_eta.

A sample command line entry follows:

```
/etc/pam_CA_eta/test_config [config=/path/to/config_file]
```

An example session follows:

```
./test_config [config=/path/to/config_file]
# ./test_config
./test_config: succeeded
Trace file is /tmp/test_config-trace.1274
```

As the command output shows, a trace file was generated which contains all the details of the configuration file parsing.

View the CAM Service

You can perform the following procedure to find out who started the service.

To view the CAM service

1. Log on to your UNIX machine as root by using the Telnet or SSH client.
2. Issue the following UNIX command:

   ```
   ps -ef | grep cam
   ```

   A display similar to the following one appears:

   ```
   root 13822        1 11 11:30:12 ?   0:00 cam
   root 13843 13753  3 11:56:31 pts/5  0:00 grep cam
   ```

Note: If the system's root user does not start the services, they will appear started, but you will be unable to use them. Identity Management issues the following message: “Permission denied: user must be root”.

Testing the LDAP/TLS Connection

You can use the test_ldap tool, located in the /etc/pam_CA_eta directory, to verify the connection to the Provisioning Server (using the configuration file parameters). A sample command line entry follows:

```
/etc/pam_CA_eta/test_ldap [config=/path/to/config_file]
```

An example session follows:

```
./test_ldap [config=/path/to/config_file]
# ./test_ldap: succeeded
Trace file is /tmp/test_ldap-trace.1277
```

As the command output shows, a trace file was generated which contains all the details of the configuration file parsing and the connection to the Provisioning Server.

Testing the Password Synchronization

You can use the test_sync tool, located in the /etc/pam_CA_eta folder, to verify that the password update of a local account is effectively propagated by the Provisioning Server. A sample command line entry follows:

```
/etc/pam_CA_eta/test_sync <user> <password> [config=/path/to/config_file]
```

An example session follows:

```
# /etc/pam_CA_eta/test_sync pam002 newpass1234
Identity Management password synchronization started.
:ETA_S_0245<MGU>, Global User 'pam002' and associated account passwords updated
successfully: (accounts updated: 2, unchanged: 0, failures: 0)
Identity Management password synchronization succeeded.
/etc/pam_CA_eta/test_sync: succeeded
Trace file is /tmp/test_sync-trace.2244
```

As the command output shows, a trace file was generated which contains all of the details of the configuration file parsing, the connection to the Provisioning Server, and the update of the account.

When using the verbose mode (by using the default verbose yes parameter in the configuration file), the command provides informational and potential error messages about the password propagation.

Test the PAM Framework

A PAM test library is available to verify that the password changes are correctly detected by the PAM framework.

To test the PAM framework

1. Copy the pam_test file to the /usr/lib/security(/hpux32) folder.
2. Add a password class line for the pam_test library with no parameters.

   An example for Solaris follows:

   ```
   other password optional /usr/lib/security/pam_test
   ```
3. Issue a passwd command on a test user and then search for the pam_test[<pid>] tagged line in the syslog file.

   The command output shows the name of the generated trace file, for example:

   ```
   pam_test[1417]: Succeeded, trace file is /tmp/pam_test-trace.1417
   ```

Password Synchronization on OS400

The Password Synchronization agent lets password changes, made on the OS/400 endpoint system, be propagated to your other accounts managed by Identity Management. The Password Synchronization agent works as follows:

1. Install and execute the agent on the OS/400 endpoint system

   As part of the installation, the program is registered with the OS/400 system so that when users change their passwords, the agent sends the password changes on to the Provisioning Server.

2. The Provisioning Server propagates the password change to the associated accounts.

   Password changes initiated from the Change Password command (CHGPWD) or Change Password (QSYCHGPW) API are received by the agent.

3. The agent logs operation success or failure to a log file located in PWDSYNCH/LOG.

To install the Password Synchronization Agent

1. Locate the Provisioning Component installation media.
2. Run the Password Sync Agent installer or OS/400 under \Agent
3. Follow the onscreen instructions to complete the installation.

   Note: The installation instructions in the Endpoint Agent Software link are included in the following sections.

Install the OS400 Password Synchronization Agent

You must have *ADDOBJ privileges and the following are necessary for the agent to receive password change notifications:

- System value QPWDVLDPGM must be set to *REGFAC
- Program must be registered with the command WRKREGINF EXITPNT(QIBM_QSY_VLD_PASSWRD)
- The environment must allow password changes to come from endpoint accounts. An administrator with access to the Management Console enables this feature.

The agent is initiated only when a password change is made. To change the password, issue the CHGPWD command.

Note: The Global User must be flagged for password synchronization.

On the iSeries

1. Log on as a user with *ALLOBJ and *SECADM privileges (for example, QSECOFR).
2. Create a user called PWDSYNCH:

   ```
   CRTUSRPRF USRPRF(PWDSYNCH) PWDEXP(*YES)
   ```

   Note: As a security measure, the user is created with the password expired.
3. Create a savefile to store the installation package in a library of your choice (for example, MYLIB):

   ```
   CRTSAVF MYLIB/PWDSYNCH
   ```
4. On the Windows machine with the savefile, use FTP to transfer the savefile to the iSeries:

   ```
   ftp <hostname>
   binary
   cd MYLIB
   put PWDSYNCH.FILE
   ```
5. On the iSeries, extract the program from the savefile:

   ```
   RSTLIB SAVLIB(PWDSYNCH) DEV(*SAVF) SAVF(MYLIB/PWDSYNCH)
   ```

   This command extracts and installs the synch agent into the PWDSYNCH library.
6. Verify the installation:

   ```
   DSPLIB PWDSYNCH
   ```

   The following objects should be displayed:

Object	Type	Attribute
PWDSYNCH	*PGM	CLE
CONFIG	*FILE	PF
LOG	*FILE	PF

7. Set up the iSeries to use PWDSYNCH as the password validation exit program:

   ```
   CHGSYSVAL SYSVAL(QPWDVLDPGM) VALUE(*REGFAC)
   ADDEXITPGM EXITPNT(QIBM_QSY_VLD_PASSWRD) FORMAT(VLDP0100) PGMNBR(1)
   PGM(PWDSYNCH/PWDSYNCH) TEXT('eTrust Admin Password Synch Agent')
   ```
8. On the iSeries, specify the connection parameters for your CA IAM Connector Server:

   ```
   EDTF FILE(PWDSYNCH/CONFIG)
   ```

Install the OS400 Password Synchronization Agent

You must have *ADDOBJ privileges and the following are necessary for the agent to receive password change notifications:

- System value QPWDVLDPGM must be set to *REGFAC
- Program must be registered with the command WRKREGINF EXITPNT(QIBM_QSY_VLD_PASSWRD)
- The environment must allow password changes to come from endpoint accounts. An administrator with access to the Management Console enables this feature.

The agent is initiated only when a password change is made. To change the password, issue the CHGPWD command.

Note: The Global User must be flagged for password synchronization.

On the iSeries

1. Log on as a user with *ALLOBJ and *SECADM privileges (for example, QSECOFR).
2. Create a user called PWDSYNCH:

   ```
   CRTUSRPRF USRPRF(PWDSYNCH) PWDEXP(*YES)
   ```

   Note: As a security measure, the user is created with the password expired.
3. Create a savefile to store the installation package in a library of your choice (for example, MYLIB):

   ```
   CRTSAVF MYLIB/PWDSYNCH
   ```
4. On the Windows machine with the savefile, use FTP to transfer the savefile to the iSeries:

   ```
   ftp <hostname>
   binary
   cd MYLIB
   put PWDSYNCH.FILE
   ```
5. On the iSeries, extract the program from the savefile:

   ```
   RSTLIB SAVLIB(PWDSYNCH) DEV(*SAVF) SAVF(MYLIB/PWDSYNCH)
   ```

   This command extracts and installs the synch agent into the PWDSYNCH library.
6. Verify the installation:

   ```
   DSPLIB PWDSYNCH
   ```

   The following objects should be displayed:

Object	Type	Attribute
PWDSYNCH	*PGM	CLE
CONFIG	*FILE	PF
LOG	*FILE	PF

7. Set up the iSeries to use PWDSYNCH as the password validation exit program:

```
CHGSYSVAL SYSVAL(QPWDVLDPGM) VALUE(*REGFAC)
ADDEXITPGM EXITPNT(QIBM_QSY_VLD_PASSWRD) FORMAT(VLDP0100) PGMNBR(1)
PGM(PWDSYNCH/PWDSYNCH) TEXT('eTrust Admin Password Synch Agent')
```

8. On the iSeries, specify the connection parameters for your CA IAM Connector Server (CA IAM CS):

```
EDTF FILE(PWDSYNCH/CONFIG)
```

Update the Endpoint in the User Console

In the User Console, update the endpoint to indicate that the agent is installed.

Follow these steps:

1. Log in to the User Console.
2. Search for the endpoint with the agent installed.
3. Click the Endpoint Settings tab.
4. Select the Password Synchronization Agent Installed check box.

Enable an Environment for Password Synchronization

After you install the Password Synchronization Agent, you enable the environment to receive password changes that are made on the endpoints. For this task, an administrator needs access to the Management Console and CA Directory to enable the environment to accept these changes.

Follow these steps:

1. For new users, you use the Management Console as follows:
   a. Select the environment.
   b. Click Advanced Settings, Provisioning.
   c. Select the Enable Password Changes from Endpoint Accounts check box.
2. For existing users, set the eTPropagatePassword attribute to 1 in CA Directory.

SSL Configuration

SSL is used to encrypt communication between the synch agent and the Provisioning server. This is important for the synch agent because SSL sends passwords across the network. SSL is recommended to always be used.

The synch agent must trust the Provisioning Server's certificate in order to connect with SSL. Therefore, the certificate must be installed on the iSeries machine and configured so that the certificate is trusted by the synch agent. These tasks are performed by the Digital Certificate Manager, an optional component of OS/400. Please follow the OS/400 documentation regarding installation and setup of the Digital Certificate Manager.

Install the Provisioning Server Certificate

The following operating system components must be installed on your iSeries machine to use SSL:

- Cryptographic access provider licensed program (5722-AC3)
- Digital Certificate Manager (Option 34 of OS/400)
- IBM HTTP Server for iSeries (5722-DG1)

On the iSeries

1. Upload the Provisioning server certificate from the Provisioning server machine to the iSeries. The certificate can be found at:

   ```
   C:\Program Files\CA\Identity Manager\Provisioning
   Server\Data\Tls\server\et2_cacert.pem
   ```

2. Log in to the DCM.

   Using a web browser, go to http://<hostname>:2001. When prompted, log on as QSECOFR and click Digital Certificate Manager.

3. Click Select a Certificate Store and select the *SYSTEM certificate store. If this store does not exist, create a store called *SYSTEM, then enter the certificate store password.

4. Import the certificate as a CA Certificate using the DCM.

   Click Manage Certificates, Import Certificate. Select the Certificate Authority (CA) option and enter the file name of the Provisioning server certificate. (This is where you uploaded the certificate in step 1). Enter the label Provisioning Server for the certificate.

5. After importing the CA certificate to the endpoint *SYSTEM keystore, verify that the IBM Directory client QIBM_GLD_DIRSRV_CLIENT can access the *SYSTEM keystore. Otherwise, the SSL initialization call of the PSA fails.

6. Configure the Directory Services client application to trust the Provisioning Server certificate by opening Manage Applications, Define CA Trust List and choosing Directory Services Client.

   The Provisioning Server certificate should be listed here if imported correctly from step 4.

   Click Trusted for the Provisioning Server certificate, then click OK.

7. Give PUBLIC read permission to the SSL files and grant read access to the *SYSTEM certificate store:

   `(/QIBM/userdata/ICSS/Cert/Server/default.kdb)`

   Grant read and execute permission to the parent folder

   `(/QIBM/userdata/ICCS/Cert/Server)`

   Note: Adopting authority of user PWDSYNCH does not work in the / file system, so access must be granted for all users.

Un-Install the Password Synchronization Agent

If you need to un-install the Password Synchronization Agent, follow this procedure.

From the password validation exit point

1. Remove PWDSYNCH:

   ```
   RMVEXITPGM EXITPNT(QIBM_QSY_VLD_PASSWRD) FORMAT(VLDP0100) PGMNBR(1)
   ```

2. Delete the synch agent library:

   ```
   DLTLIB PWDSYNCH
   ```

3. Delete PWDSYNCH user

   ```
   DLTUSRPRF PWDSYNCH
   ```

4. Remove the Provisioning server certificate by following the SSL instructions to log onto the DCM and work with the *SYSTEM certificate store:

   Click Manage Certificates, Delete Certificate and select 'Certificate Authority (CA)'

   Select 'Provisioning Server' certificate and click Delete.

OS/400 Password Agent Parameter Must be Set Correctly

The "pwd_case_action" parameter must have the value set correctly for it to work. Correct values include:

- pwd_case_action = pwd_case_unchanged
- pwd_case_action = pwd_to_uppercase
- pwd_case_action = pwd_to_lowercase

If pwd_case_action = [invalid value] the password will be forced to uppercase.

Note: When setting the flag pwd_case_action to pwd_to_uppercase or pwd_to_lowercase in the OS400 PSA configuration file, the password might not be propagated back to the global user if the supplied passwords are not compliant to the password policy settings in Provisioning Server. For example, some password policies may require the password values to at least contain 1 uppercase or lowercase value.

Note: Note the QPWDLVL (Password Level) system value when configuring the Password Synch Agent

- When QPWDLVL is set to 0 (default value) on the AS400 system, passwords with a length of 1 to 10 uppercase characters are supported.
- When QPWDLVL is set to 2 or 3, passwords from 1 to 128 characters with mixed case are allowed.

By default, the PSA propagates the unchanged password to the Provisioning Server. However, regardless of the value of QPWDLVL, you can force the PSA to propagate passwords with upper case or lower case by setting “pwd_case_action” to “pwd_to_uppercase” or “pwd_to_lowercase” respectively.

Chapter 5: Groups

You can create several types of groups, or a combination of these types:

- Static group--A list of users who are added interactively
- Dynamic group--Users belong to the group if they meet an LDAP query (Requires an LDAP directory as the user store)

  Note: The Dynamic Group Query field is not included in the Create Group task or other group tasks even if this field exists in the directory.xml for a group. You include Dynamic Group Query field in the task by editing the associated profile screen.
- Nested group--A group containing other groups (Requires an LDAP directory as the user store)

Note: To view the static, dynamic, and nested groups to which a user belongs, use the Groups tab for the User object. This tab appears in the View and Modify User tasks by default.

This section contains the following topics:

Create a Static Group

You can associate a collection of users in a *static group*. You manage the static group by adding or removing individual users from the group's membership list. To see the list of members for a group, use the Membership tab, which is included with the View and Modify Group tasks by default.

Note: The Membership tab displays only the members who are explicitly added to the group. It does not display members who are added dynamically.

To create a static group:

1. In the User Console, select Groups, Create Group.
2. Choose to create a new group or a copy of a group and click **OK**.

3. On the Profile tab, enter a group name, group organization, description, and group administrator name.
4. Click the Membership tab.
5. Click Add a user.
6. Search for users to include.
7. Put a check next to the users and click Select.
8. Click Submit.

Create a Dynamic Group

You can create a *dynamic group* by defining an LDAP filter query using the User Console to dynamically determine group membership at runtime without having to search and add users individually.

For example, if you wanted to generate a group that lists all U.S. employees of NeteAuto, you could define an LDAP search filter similar to the following in the Dynamic Group Query field of the User Console:

ldap:///cn=Employees,o=NeteAuto,c=US??sub

You could also modify this query to locate employees outside the United States.

Static, Dynamic, and Nested Groups Example (see page 125) shows an example of a group created by static, dynamic, and nested groups.

You include Dynamic Group Query field in the task by editing the associated profile screen. It is not included by default in the Create Group task.

Note: To enable dynamic groups, system administrators configure support in the directory configuration file (directory.xml):

- Add the GroupTypes element in the Directory Groups Behavior section as follows:

  <GroupTypes type=*type*>

  type can be NESTED (see page 123), DYNAMIC, or ALL.

  GroupTypes is case-sensitive.

- Map the %DYNAMIC_GROUP_MEMBERSHIP% well-known attribute to a physical attribute that exists in the user store.

To create a dynamic group:

1. In the User Console, select Groups, Create Group.
2. Choose to create a new group or a copy of a group and click **OK**.

3. On the Profile tab, enter a group name, group organization, description, and group administrator name.
4. Enter an LDAP search filter like the following example in the Dynamic Group Query field:

   ldap:///cn=Employees,o=NeteAuto,c=US??sub?
5. Click Submit.

Note: Only an administrator with the Modify Group task can change a group's dynamic membership.

Dynamic Group Query Parameters

You can use the following dynamic query parameters in the search:

ldap:///*<search_base_DN>*??*<search_scope>*?*<searchfilter>*

- *<search_base_DN>* is the point from where you begin the search in the LDAP directory. If you do not specify the base DN in the query, then the group's organization is the default base DN.
- *<search_scope>* specifies the extent of the search and includes:
  - sub -- Returns entries at the base DN level and below
  - one -- Returns entries one level below the base DN you specify in the URL. (default)
  - base -- Uses one instead, ignoring base as a search option

  Using one or base obtains only the users in the Base DN organization.

  Using sub obtains all users under the Base DN organization and all sub-organizations in the tree.

- *<searchfilter>* is the filter that you want to apply to entries within the scope of the search. When you enter a search filter, use the standard LDAP query syntax as follows:

  (<logical operator ><comparison><comparison...>)

  - <logical operator> is one of the following:

    Logical OR: |

    Logical AND: &

    Logical NOT: !

  - <comparison> indicates <attribute><operator><value>

  For example:

  (&(city=boston)(state=Massachusetts))

  The default search filter is (objectclass=*).

Note the following when creating a dynamic query:

- The "ldap" prefix must be lowercase, for example:

  ldap:///o=MyCorporation??sub?(title=Manager)

- You cannot specify the LDAP server host name or port number. All searches occur within the LDAP directory that is associated with the environment.

The following table includes sample LDAP queries:

Description	Query
All users who are managers.	ldap:///o=MyCorporation??sub?(title=Manager)
All managers in the New York West branch office	ldap:///o=MyCorporation??one?(&(title=Manager) (roomNumber=NYWest))
All technicians with a cell phone	ldap:///o=MyCorporation??one? (&(employeetype=technician) (mobile=*))
All employees whose employee numbers are between 1000 and 2000	ldap:///o=MyCorporation, (& (ou=employee) (employeenumber >=1000) (employeenumber <=2000))
All help desk administrators who have been employed at the company for more than 6 months	ldap:///o=MyCorporation,(& (cn=helpdeskadmin) (DOH => 2004/04/22) **Note:** This query requires that you create a DOH attribute for the user's date of hire.

Note: The > and < (greater than and less than) comparisons are lexicographic, not arithmetic. For details on their use, see the documentation for your LDAP directory server.

Create a Nested Group

If the user store is an LDAP directory, you can add a group as a member of another group. The group is called a *nested group*.

The group containing the nested group is called a *parent group*. Members of the nested group become members of the parent group. However, members of the parent group do not become members of a nested group.

Nested groups are similar to email distribution lists where one list can be a member of another. With nested groups, you can add groups and users as members in the group. By nesting a group in another group's membership list, you could include all nested groups members.

For example, if you created separate groups for the manufacturing, design, shipping, and accounting divisions of a company, you can construct a parent group for the entire company by nesting all the separate division groups as members of the company parent group. As a result, any changes you made to the manufacturing, design, shipping, and accounting nested groups would be automatically reflected in the nested group for the entire company. A group that is nested within another group can be dynamic and/or contain other nested groups.

The figure in Static, Dynamic, and Nested Groups Example (see page 125) shows a parent group created by static, dynamic, and nested groups.

Be aware of the following before creating a nested group:

- Only an administrator with the Modify Group Members task can add or change nested groups from the group's static member list in the User Console.
- Only users with the appropriate administrator privileges can modify, add, or remove members from a group.

  For example, if parent Group A is created by nested groups B and C, the Group A administrator can only modify the members of Group A and not B and C. Groups B and C can only be modified by their appropriate administrators.
- To enable nested groups, system administrators configure nested group support in the directory configuration file (directory.xml):
  - Add the GroupTypes element in the Directory Groups Behavior section as follows:

    <GroupTypes type=*type*>

    type can be NESTED, DYNAMIC (see page 120), or ALL.

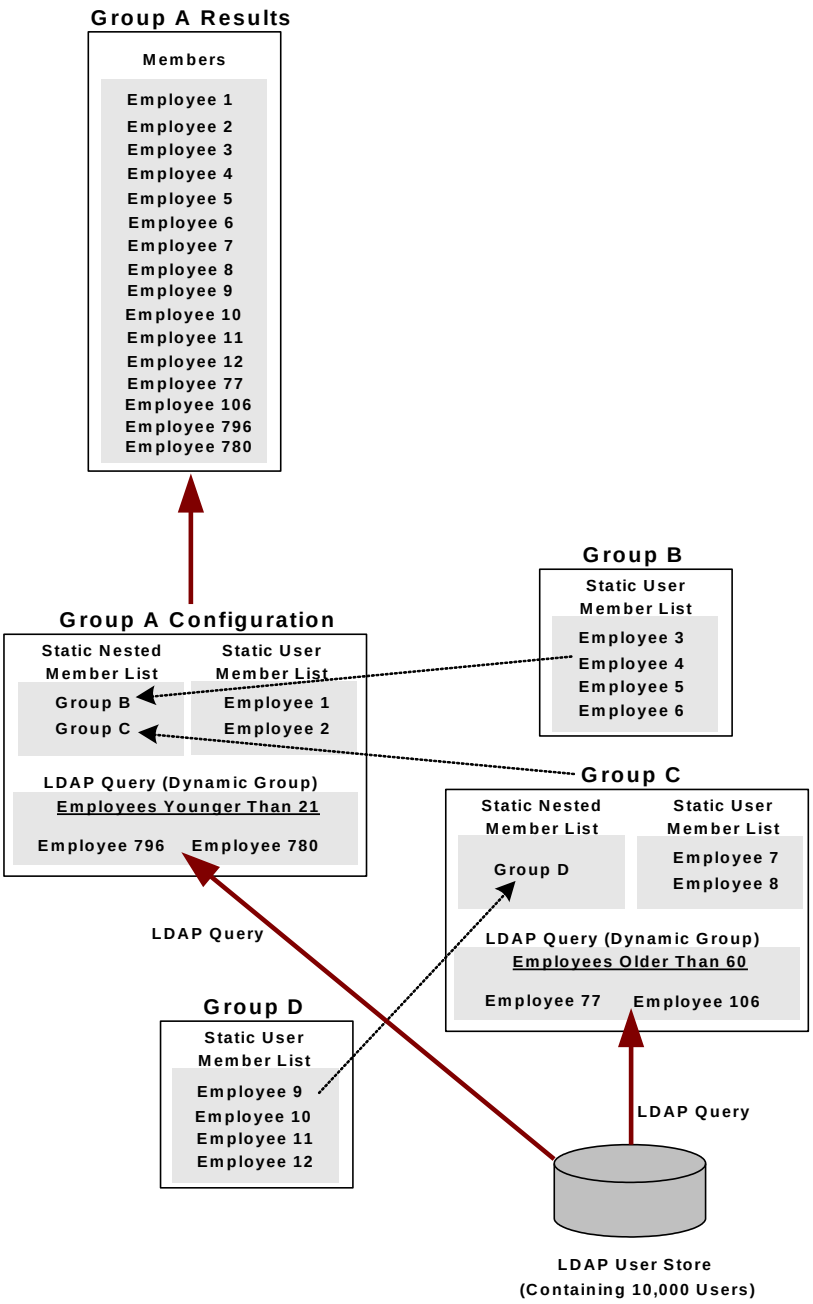
    GroupTypes is case-sensitive.
  - Map the %NESTED_GROUP_MEMBERSHIP% well-known attribute to a physical attribute that exists in the user store.

To create a nested group:

1. In the User Console, select Groups, Create Group.
2. Choose to create a new group or a copy of a group and click **OK**.
3. On the Profile tab, enter a group name, group organization, description, and group administrator name.
4. On the Membership tab:
   a. Click Add a group to add a nested group to this group.
   b. Search for an existing group.
   c. Put a check next to the group and click Select.
   d. Click Submit.

Static, Dynamic, and Nested Groups Example

Groups can be complex, consisting of a combination of dynamic, static, or nested groups. The following figure shows an example of a parent group created by static, dynamic, and nested groups.



In the previous figure:

- Parent Group A contains nested groups B and C, two static users, and a dynamic LDAP query that lists all employees who are younger than 21 years old.
- Group B is composed of four static users.
- Parent Group C contains nested Group D, two static users, and a dynamic LDAP query that lists all employees who are older than 60 years old.
- Group D contains four static users.
- The top of the figure lists the Group A members that result from the nested groups, dynamic queries, and static user member lists from Groups B, C, and D.

Group Administrators

On the Administrators tab of the Create or Modify Group tasks, you can specify users and groups as administrators of a group. When you assign a user as a group administrator, make sure that the administrator has a role with appropriate scope for managing the group. For example:

1. Use Modify Group to assign a user as an administrator of a group.
2. Assign that user an admin role with group management tasks, such as Modify Group Members, or user management tasks with a Groups tab.
3. Check that the role has appropriate scope over the group.
    a. Use View Admin Role on the role that you assigned with group management tasks.
    b. On the Members tab, verify that a policy exists with the following:

        A member rule that the group administrator meets

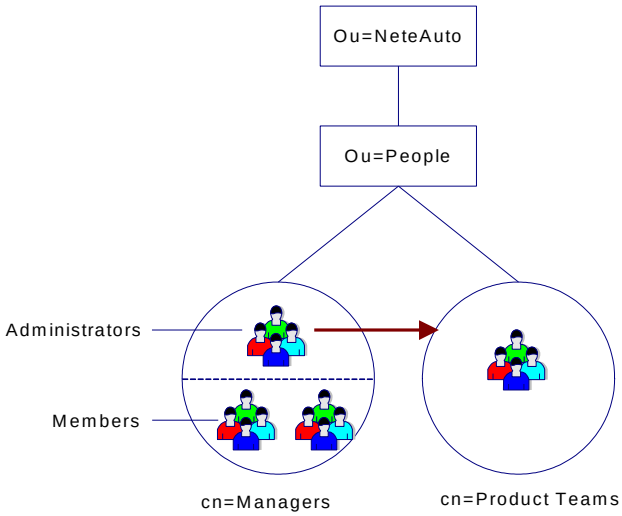
        A scope rule that includes the group

        A scope rule that includes some users to be added to the group

Note: To enable groups to be administrators of other groups, system administrators configure group administrator support in the directory configuration file (directory.xml):

- Set the AdminGroupTypes type=ALL in the Directory AdminGroups Behavior section. AdminGroupTypes is case-sensitive.
- Map the %GROUP_ADMIN_GROUP% well-known attribute to a physical attribute that exists in the user store.

When you assign a group as an administrator, only administrators of that group will be administrators of the group you are creating or modifying. Members of the administrator group you specify will not have privileges to manage the group. The following illustration shows a group as an administrator of another group.



In this example:

- The group Managers is an administrator of the group Product Teams.
- Administrators of the Managers can manage the Product Teams group. Members of the Managers group cannot.

Troubleshooting: Groups Do Not Appear Under Search Results

Symptom:

When I search for a group using the Modify Group Membership task, the group does not appear in the search results.

Solution:

By default, the following option is selected on the Search tab of the Modify Group Members task:

All groups for which the user is an administrator

When this option is selected, a group only appears in search results when the administrator who is using the Modify Group Members task is an administrator of the group. To cause all groups that an administrator can manage to appear in the search results for the Modify Group Members task, select the following option:

All groups in the administrator's scope

Chapter 6: Provisioning Roles

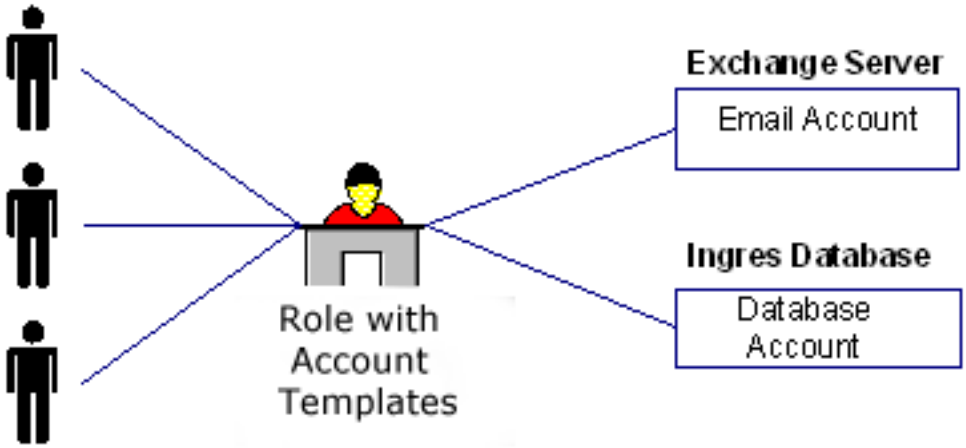
This section contains the following topics:

Creating Roles to Assign Accounts

In most organizations, administrators spend significant time providing users with login accounts for different systems and applications. To simplify this repetitive activity, you can create provisioning roles, which are roles that contain account templates. The templates define the attributes that exist in one type of account. For example, an account template for an Exchange account defines attributes such as the size of the mailbox. Account templates also define how user attributes are mapped to accounts.

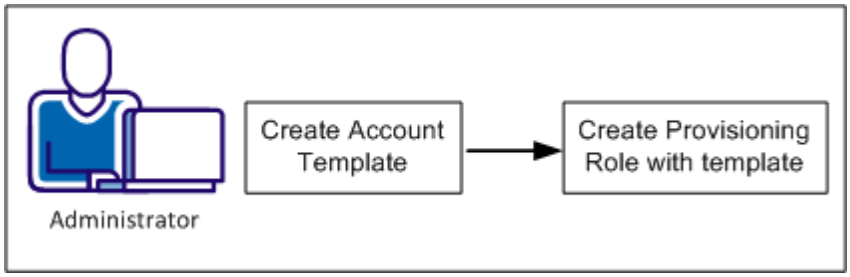
Consider an example where every employee at Forward, Inc needs access to a database and email. An administrator wants to avoid creating a database account and an email account for each employee one at a time. Therefore, the administrator creates a provisioning role for that company. The role contains an account template for a Microsoft Exchange server, to provide email accounts, and a template for an Oracle database. In this example, the Exchange server and the Oracle database are named endpoints, which are the system or application where the accounts exist.

Note: Forward, Inc. is a fictitious company name which is used strictly for instructional purposes only and is not meant to reference an existing company.



After the roles are created, business administrators, such as managers or support personnel, can assign those roles to users to give them accounts in endpoints. After users receive the role, they can log in to the endpoint.

Creating a provisioning role that includes an account template is a two-step process as follows:



The following sections explain how to create a role that can be used to assign accounts:

1. Create an Account Template (see page 131)
2. Create a Provisioning Role (see page 132)

Create an Account Template

To simplify account management, you create and maintain accounts using account templates, which are used in provisioning roles. A provisioning role contains one or more account templates. When you apply that role to a user, the user receives the accounts as defined by the templates.

These templates provide the basis for accounts on a specific endpoint type.

Using account templates, you can:

- Control what account attributes users have on an endpoint when their accounts are created
- Define attributes using rule strings or values
- Combine account attributes from different provisioning roles, so users have only one account, on a specific endpoint, with all the necessary account attributes
- Create or update account attributes as global users change provisioning roles

A default account template for each endpoint type is installed with the Identity Management server. In a provisioning role, you can use the default account template or you can create your own account templates for any endpoint that you have configured.

To create an account template

1. Navigate to Endpoints, which may be listed under tasks, and click Account Templates, Create Account Template.
2. Select an endpoint type for the template.
3. Define Endpoint Name as the system name of the endpoint or localhost if that applies.
4. Select an endpoint to use on the Endpoints tab.
5. Complete the fields in the tabs or use the default values.

   Each endpoint type has a different set of tabs. Click Help for field definitions.
6. Click Submit.

Note: If more than one endpoint is specified while searching for endpoint objects in the Account Template, the common subset (intersection) of the related objects is returned. An example is an Active Directory group that exists on each of the selected endpoints that are associated with the Account Template. When the search results show attributes other than the object name, it shows the attribute values of the objects that are associated with the first endpoint. An example is the description attribute for the language object in a PeopleSoft connector.

Create a Provisioning Role

You create a provisioning role once you decide about the role requirements:

- Which users need other accounts
- Which accounts are associated with the role
- Who the members, administrators, and owners of the role are

To create a provisioning role

1. In the User Console, navigate to Roles and Tasks, Provisioning Roles, Create Provisioning Role.

   For details on each tab, click the Help link on the screen.

2. Complete the Profile tab. Only the Name field is required.

   Note: You can specify custom attributes on the Profile tab that specify additional information about provisioning roles. You can use this additional information to facilitate role searches in environments that include a significant number of roles.

3. Complete the Account Templates tab.

   a. Click an Endpoint Type, such as an ActiveDirectory.

   b. Click an account template.

      The templates that you can click are based on Endpoint Type.

   c. Add more account templates as needed for different endpoint types.

4. Complete the Provisioning Roles tab if you want to nest provisioning roles in this tab.

   This step requires that you have enabled nested roles (see page 135) for this environment.

5. Complete the Administrators tab by adding admin rules that control who manages members and administrators of this role.

6. Complete the Owners tab by adding owner rules that control who can modify this role.

7. Click Submit.

8. To verify that the role was created, click Provisioning Roles, View Provisioning Role.

Role and Template Tasks

In the User Console, you can create and manage provisioning roles by choosing Roles and Tasks and selecting a task under Provisioning Roles. Tasks exist for the standard operations, such as making a user a member of a role and modifying or deleting a role.

Before creating a provisioning role, you need an account template to include in that role or a provisioning role that you want to import. You can import roles that were created in the Provisioning Manager or eTrust Admin. However, Identity Management does not support nested roles that were created in eTrust Admin.

Assign New Owners for Provisioning Roles

You can select one or more provisioning roles and assign owner policies to control who can modify the roles.

To assign new owners for provisioning roles

1. Log into the User Console as a user with the System Manager role.
2. Click Tasks, Roles, or click Roles, and Tasks.
3. Click Provisioning Roles, Create Owner Policies for Provisioning Roles.
4. Select one or more provisioning roles.
5. Complete the Owners tab by adding owner rules that control who can modify this role.
6. Click Submit.

Users who meet the new owner policies can modify the selected provisioning roles.

Passwords for Accounts Created by Provisioning Roles

When a user is assigned a provisioning role, account creation for that user fails if the Identity Management user's password does not meet the endpoint's password requirements. This situation includes creation of a new user with a temporary password.

Therefore, set the password policy to match, or be stricter than, the endpoint password requirements. You can set the password policy by using the Identity Management Password Policy or the Provisioning Password Profile. If both methods are used, the policies must match.

Provisioning Role Event Processing Order

Some default Identity Management tasks include *events*, actions that Identity Management performs to complete a task, that determines provisioning role membership. For example, the default Modify User task includes the AssignProvisioningRoleEvent and the RevokeProvisioningRoleEvent. Assigning or revoking a provisioning role may add or remove an account on an endpoint. In some cases, the endpoint may require that all Add actions occur before Remove actions.

To make Identity Management process Add actions first, you enable the Accumulation of Provisioning Role Membership Events setting in the Management Console. When this setting is enabled, Identity Management accumulates all of the Add and Remove actions into a single event, called the AccumulatedProvisioningRolesEvent. For example, if the Modify User task assigns a user to three provisioning roles and removes that user from two other provisioning roles, an AccumulatedProvisioningRolesEvent will be generated which contains five actions: 3 Add actions and 2 remove actions.

When this event executes, all Add actions are combined into a single operation and sent to the Provisioning Server for processing. Once processing of the Add actions completes, Identity Management combines the Remove actions into a single operation and sends that operation to the Provisioning Server.

Enabling this setting affects the following Identity Management functionality:

- **Provisioning Roles Tab in User Tasks**

  When an administrator adds or removes a user from a provisioning role using the Provisioning Roles tab, Identity Management accumulates those actions into a single event.

- **Identity Policies**

  All provisioning role membership events (AssignProvisioningRoleEvent or RevokeProvisioningRoleEvent) that are generated as a result of an Identity Policy evaluation are accumulated into a single AccumulatedProvisioningRolesEvent. Identity Management executes this event like any other secondary event. For example, consider an identity policy set that includes two identity policies: Policy A revokes membership in the Provisioning Role A and Policy B makes users members of Provisioning Role B. If Identity Management determines that a user no longer satisfies Policy A, but now satisfies PolicyB, an AccumulatedProvisioningRolesEvent that contains two actions (one for the remove action and one for the add action) is generated. The Add action is executed first and then the Remove action is executed.

- **View Submitted Tasks**

  To view the status of the AccumulatedProvisioningRolesEvent and the status for each of the individual actions, use the View Submitted Tasks task to view event details.

  If one of the individual actions fails, the status of the event is failed, which moves the task to a failed state.

- **Workflow**

  You can associate a workflow process with the AccumulatedProvisioningRolesEvent. In this case, an approver can approve or reject the entire event, which approves or rejects each of the individual events.

  Additional configuration is required to enable workflow for individual events within the AccumulatedProvisioningRolesEvent.

- **Auditing**

  Identity Management audits information about the AccumulatedProvisioningRolesEvent and each individual event.

Enable Provisioning Role Membership Event Accumulation

Identity Management provides a configuration setting in the Management Console that enables the combination of all Add and Remove actions for a provisioning role membership event into a single operation. Once combined, Identity Management processes the Add actions as a single operation before processing the Remove actions.

This setting allows sequencing of events required by some endpoint types.

Note: This feature is disabled by default.

To enable Provisioning Role Membership Event Accumulation

1. Access the Identity Management Management Console.
2. Click Environments.
3. Select the environment that you want to configure.
4. Open Advanced Settings, Provisioning.
5. Select the Enable Accumulation of Provisioning Role Membership Events check box.
6. Restart the application server.

Enable Nested Roles in an Environment

You can include a provisioning role within another provisioning role. The included role is named a nested role.

For example, you could create an Employee provisioning role. The Employee role would provide accounts needed by all employees, such as email accounts. You include the Employee role in department-specific provisioning roles, such as a Finance role and a Sales role. The department provisioning roles would provide accounts related only to that department. This combination of roles provides the right accounts for each user.

To enable Nested Roles in an environment

1. In the Management Console, select the environment.
2. Click Role and Task Settings, Import.
3. Select Nested Provisioning Roles Support.
4. Click Finish.
5. Restart the environment.

Include a Role in a Provisioning Role

To include a role in a Provisioning Role

1. Navigate to Roles and Tasks, Provisioning Roles, Modify Provisioning Roles.
2. Complete the Provisioning Roles tab by clicking Add a Role and select a provisioning role.

   For performance reasons, we recommend limiting role nesting to three levels. For example, you are including in the current provisioning role (the first-level role) another role (the second-level role), which can contain a third-level role. We recommend that the third-level role contains no role.
3. Complete the owner policy by modifying the owner rule.

   The scope must be equal to or broader than the scope for the role you added.
4. Click Submit.

Attributes in Account Templates

The attributes in account templates determine how attributes are defined in the account.

Capability and Initial Attributes

Account templates include two types of attributes:

- *Capability attributes* represent account information, such as storage size, quantity, frequency limits, or group memberships. Provisioning Manager bolds the capability attributes on all account template screens to make identifying capability attributes easy.
- *Initial attributes* represent all information that is initially set for an account, such as account name, password, and account status and personal information such as name, address and telephone numbers.

Accounts are considered synchronized with their account templates when all the capability attributes are synchronized. These are attributes that differ from endpoint type to endpoint type such as group memberships, privileges, quotas, login-restrictions; they control what the user can do when logging into the account.

Synchronization does not update other account attributes. They are initialized from the account templates during account creation and they can also be updated during propagation functions. The Provisioning Server provides two propagation functions (an immediate update of accounts at the time the account template is changed and an update of accounts at the time global user attributes change).

Finding Capability and Initial Attributes

To find out which attributes are defined as capabilities and which are initial, you need to generate the eTACapability.txt file. Enter the following command from a Windows Command prompt:

```
PS_HOME\dumpptt.exe -c > eTACapability.txt
```

PS_Home

Specifies C:\Program Files\CA\Identity Manager\Provisioning Server\bin

A version of the file is generated for all of the connectors that you have installed.

Rule Strings in Account Templates

When you create an account template, you use rules strings to define the format of many account attributes. Rule strings are variables for the actual value. Rules strings are useful when you want to generate attributes that change from one account to another. When rules are evaluated, Identity Management replaces the rule strings entered in the account templates with data specified in the user object.

Note: Rule evaluation is not performed on accounts created during an exploration or on accounts created without provisioning roles.

The following table lists the rule strings in Identity Management:

Rule String	Description
%AC%	Account name
%D%	Current date in the format *dd/mm/yyyy* (the date is a computed value that does not involve the global user information*).* This rule string is equivalent to one of the following: %$$DATE()% %$$DATE%
%EXCHAB%	Mailbox hide from exchange address book
%EXCHS%	Mailbox home server name
%EXCMS%	Mailbox store name
%GENUID%	Numeric UNIX/POSIX user identifier. This rule variable is the same as %UID% as long as the global user UID value is set. However, if the global user has no assigned UID value, and UID-generation is enabled (Global Properties on System Task), several actions occur. The next available UID value is allocated, assigned to the global user, and used as the value of this rule variable.
%P%	Password
%U%	Global user name
%UA%	Full address (generated from street, city, state, and postal code)
%UB%	Building
%UC%	City
%UCOMP%	Company name
%UCOUNTRY%	Country

Rule String	Description
%UCU*xx*% or %UCU*xxx*%	Custom field (*xx* or *xxx* represents the two-digit or three-digit field ID as specified on the Custom User Fields tab in the System Task frame)
%UD%	Description
%UDEPT%	Department
%UE%	Email address
%UEP%	Primary email address
%UES%	Secondary email addresses
%UF%	First name
%UFAX%	Facsimile number
%UHP%	Home page
%UI%	Initials
%UID%	Numeric UNIX/POSIX User Identifier
%UL%	Last name
%ULOC%	Location
%UMI%	Middle initial
%UMN%	Middle name
%UMP%	Mobile telephone number
%UN%	Full name
%UO%	Office name
%UP%	Telephone number
%UPAGE%	Pager number
%UPC%	Postal code, ZIP Code
%UPE%	Telephone number extension
%US%	State
%USA%	Street address
%UT%	Job title

Rule String	Description
%XD%	Generates the current timestamp in XML dateTimeValue format, a fixed-length string format. In a dateValue or timeValue attribute, you can write an (:offset,length) substring expression to extract the date or time parts of the dateTimeValue. For example, %XD:1,10% yields YYYY-MM-DD; and %XD:12,8% yields HH:MM:SS.

Values for Attributes

To use a specific, constant value for an account attribute, enter the value in the account template field instead of in a rule string. For example, you can enter values for specifying frequency limits or quantity size.

If the constant attribute value must contain more than one percent sign, enter two percent signs (%%) each time. Identity Management translates them to one percent sign (%) when building the account attribute value. If the account template value contains only one percent sign, Identity Management does not generate an error. The rule states that if you want a literal value of 25%, you must specify 25%%. However, as a special case, 25% will be accepted.

Advanced Rule Expressions

To provide greater flexibility than simple global user attribute substitution, you can enter advanced rule expressions, including the following:

- Substrings of rule expressions using Offset and Length
- Combinations of rule strings and values
- Rule expressions to set multiple values for multivalued account attributes
- Rule variables for other global user attributes
- Invocation of Built-in functions
- Invocation of customer-written Program Exit functions

Combining Rule Strings and Values

You can combine rule strings and constant values into an account template attribute value. For example, if there were no %UI% rule string, you could obtain the same effect by concatenating multiple rule expressions as follows:

```
%UF:,1%%UMI:,1%%UL:,1%
```

The %UA% rule string is equivalent to the following:

```
%USA%, %UC%, %US%, %UPC%
```

You can also combine a rule string with a constant value to create a UNIX home endpoint attribute as follows:

```
/u/home/%AC%
```

Rule Substrings

The following is the syntax for creating a substring value of a rule variable:

```
%var[:offset,length]%
```

var

Represents the name of the predefined rule variable as defined in the table shown previously.

offset

(Optional) Defines the starting offset of the substring suffix. The number 1 represents the first character.

length

(Optional) Defines the ending offset of the substring suffix. A length value of asterisk (*) indicates to the end of the value.

For example, to set an account attribute to the first 4 characters of a global user's Building attribute, use the following to define the variable:

```
%UB:1,4%
```

If the Building attribute is empty or has fewer than four characters, the resulting account attribute value will have fewer than four characters.

Multivalued Rule Expressions

Most rule expressions are single-valued. They start from one user attribute value (possibly empty) and result in one account attribute value (also possibly empty). However, sometimes you want to consider an empty user attribute as 0 values. Sometimes you may want to generate multiple values to populate a multivalued account attribute value.

The following rule syntax lets you work with zero or more values that a user attribute may contain:

%*var*%

The optional multivalued flag asterisk (*) immediately after the first percent sign % of a rule expression indicates that the result of this rule expression should be 0, 1 or more than 1 value depending on how many values the referenced user attribute contains.

Most user attribute values are single-valued, so they may only contain 0 or 1 values. However, the custom attributes (CustomField01 through CustomField99) are multivalued attributes, so a rule variable referencing these attributes may contain 0, 1, or more than 1 value.

If a user attribute has more than 1 value, but you fail to include the asterisk (*) in your rule expression, then the result of the rule evaluation will be that of the first value. However, in most cases attribute values are officially unordered and as a result the value that Identity Management considers first may not be predictable.

If a user attribute has more than one value, and you include the * in your rule expression, multiple values are generated for the account attribute. Do not define such a multivalued rule expression in an account template if the account attribute being set from that account template attribute is not itself multivalued.

You can define an extended account attribute in the ADS endpoint type to be multivalued; and use this multivalued rule expression syntax to set that attribute. For example, consider an environment that defines an extended ADS account attribute named patents and custom user attribute number three also named patents.

An ADS account template could define, for the patents attribute, the rule string %*UCU03%. Then, you could change a user's patents attribute by adding one or more values. When applying the changes to the user, select the option of updating the user's accounts. This consults the account's account template, finds the rule variable %*UCU03%, and knows to copy all of the user's patents to the account's patents attribute.

Similarly, during account creation, rule strings are evaluated. Furthermore, during account template change, if the rule string has been changed, you can choose to recompute the rule for all accounts associated to the account template.

The %*var*% syntax is also meaningful for variables *var* that refer to single-valued user attributes. This is true only when concatenation is involved and if the referenced attributes are unset on users.

The optional multivalued flag asterisk (*) indicates that the rule containing a %*var*% rule variable evaluates to no value if the user attribute has no values. This is different from the single-valued rule expression %*var*%, which always evaluates to a single value, even if it is an empty string.

To understand this difference, consider the following rule strings:

```
(310)%UP%
 (310)%*UP%
```

Both rule strings appear to append area code 310 to the telephone number. However, they are different because if users have no value for their telephone number, the first rule evaluates to the account value of (310). The second rule string generates no value and leaves the account attribute unset.

On the other hand, consider the following rule strings that appear to append the telephone extension to the telephone number:

```
%UP% %UPE%
%UP% %*UPE%
```

If everyone has a telephone number, but some do not have extensions, the first rule string generates a value that includes the phone number for each user with no extension. The second rule string generates no values. In this case, use the first rule with %UPE%.

Explicit Global User Attribute Rules

Each user has many more attributes than are listed in the previous rule table. You will probably have no need to create rule expressions referencing any of these other attributes. However, should the need arise, you can use the following syntax to refer to a specific user attribute:

```
%#ldap-attribute%
```

For instance, if you must determine the value of the user's Suspended field, you would determine the corresponding LDAP attribute name for this field (which is eTSuspended) and create the rule expression that evaluates to 0 or 1, like eTSuspended:

```
%#eTSuspended%
```

As another example, you can obtain the user's assigned provisioning roles with the following rule expression:

```
%*#eTRoleDN%
```

These provisioning roles are full LDAP distinguished name values. Perhaps in conjunction with the built-in function RDNVALUE (see the table that follows), the values would be a little more useful. Note the multi-value indicator asterisk (*) so as to obtain all of the user's assigned provisioning roles as multiple values.

The substring syntax is also applicable to these rule expressions, so you could use %#eTTelephone:6,*% to mean the same thing as %UP:6,*. Each asks Identity Management to strip off the first five characters of the user's telephone field.

Built-in Rule Functions

You may use built-in rule functions in your rule expressions to perform various transformations on the values. The general form of built-in rule function invocation is

```
%[*]$$function(arg[,…])[:offset,length]%
```

where the multivalued indicator asterisk (*) and the offset and length substring specifications are once again optional.

The recognized built-in functions are as follows:

Built-in Rule Function	Description
ALLOF	Merges all the parameters into a multivalued attribute. Order is preserved and duplicates are removed. For example, if user attributes are set to the following: eTCustomField01: { A, B } eTCustomField02: { A, C } Then, the rule: %*ALLOF(%*UCU01%,%*UCU02%)% evaluates to three values { A, B, C }.
DATE	Evaluates to the current date in *dd/mm/yyyy* format. The rule expression %D% is equivalent to one of the following: %$$DATE()% %$$DATE%
FIRSTOF	Returns the first value of any of the parameters. Used to insert a default value if an attribute is not set: %$$FIRSTOF(%UCU01%,'unknown')% %$$FIRSTOF(%LN%,%UCU01%,%U%)% If none of the values is set, the result is no values. To enter a constant string in an argument, enclose it in single quotes.
INDEX	Returns one value of a multivalued attribute. Index 1 is the first value. If the index is greater than the number of values, the result is the unset (empty) value. The following rules are equivalent to the following: %$$INDEX(%*UCU01%,1)% %$$FIRSTOF(%*UCU01%)%

Built-in Rule Function	Description
NOTEMPTY	Returns the single value of its one argument, but reports a failure if this attribute value is not set. Example 1: Fail the account creation or update if the user does not have an assigned UID attribute: %$$NOTEMPTY(%UID%)% Example 2: Use the first name, unless it is not set, in which case use the last name. If neither is set, fail the account creation or update. %$$NOTEMPTY(%$$FIRSTOF(%UF%, %UL%)%)%
PRIMARYEMAIL	Returns the primary email address extracted from the multiple email addresses. The expression %UE% is equivalent to the following: %$$PRIMARYEMAIL(%UEP%)%
RDNVALUE	Treats the attribute value as an LDAP distinguished name and extracts the common name of the object from that DN: %*$$RDNVALUE(%#eTRoleDN%)% This returns the common names of all assigned provisioning roles. If the user belongs to two provisioning roles with the same common name, that role name is listed once.
TOLOWER	Converts uppercase text to lowercase: %$$TOLOWER(%AC%)%
TOUPPER	Converts lowercase text to uppercase: %$$TOUPPER(%U%)%

Built-in Rule Function	Description
TRIM	Removes leading and trailing blank characters from an attribute value. For example, “%UF %UL%” would generally create a value with a first and last name separated by a blank character. However, if the user had an empty first name attribute, this rule would generate a value ending with a trailing blank. However, using “%$$TRIM(%UF% %UL%)% ensures that no leading or trailing blank exists in the account attribute value even if one or the other of First Name and Last Name was unset.

Provisioning Role Performance

When using Identity Management with a Provisioning Server, there are some provisioning performance enhancements you may want to consider.

JIAM Object Cache

Identity Management communicates with the Provisioning Server using the Java IAM (JIAM) API. To improve communication performance, you configure a cache for objects retrieved from the Provisioning Server.

Enable the JIAM Cache

To enable the JIAM Cache

1. Access the environment settings through the Management Console. Click Advanced Settings, Miscellaneous.
2. Configure the User Defined Property for the JIAM Cache.
   - **Property**—JIAMCache
   - **Value**—true
3. Click Add.
4. Click Save.

   The User Defined Property is saved.

Define the JIAM Cache TTL (Time-to-live)

The JIAM Cache stores information for a specified period of time before the data expires. This period of time is referred to as time-to-live (TTL). You set the JIAM Cache TTL value (in seconds) to define how long data remains in the cache.

To gain the maximum benefit from locally cached data, you balance performance gains against timely data. We recommend a minimum TTL value of 1 day with a maximum value of 7 days. See the following table for time-to-live values to use:

Desired Lifetime	TTL Settings (secs)
24 hours (1 day)	86,400
72 hours (3 days)	259,200
120 hours (5 days)	432,000
168 hours (7 days)	604,800

To define the JIAM Cache TTL

1. Access the Environment through the Management Console. Click Advanced Settings, Miscellaneous.
2. Configure the User Defined Property for the JIAM Cache TTL.
   - **Property**—JIAMCacheTTL
   - **Value**—number of seconds that data remains in the JIAM Cache

     Default: 300
3. Click Add.
4. Click Save.

   The User Defined Property is saved.

Session Pooling

To improve performance, Identity Management can pre-allocate a number of sessions to be pooled for use when communicating with the Provisioning Server.

For more information on Session Pooling, see the *Management Console Online Help.*

Provisioning Tasks for Existing Environments

If you import custom roles definitions and want to enable provisioning on an environment, you must *also* import the Provisioning Only role definitions in the Management Console. These role definitions can be found in this folder:

`iam_im.ear\management_console.war\WEB-INF\Template\environment`

Note: For more information on importing role definitions, see the *Configuration Guide*.

Chapter 7: Managed Services (Basic Access Requests)

This section contains the following topics:

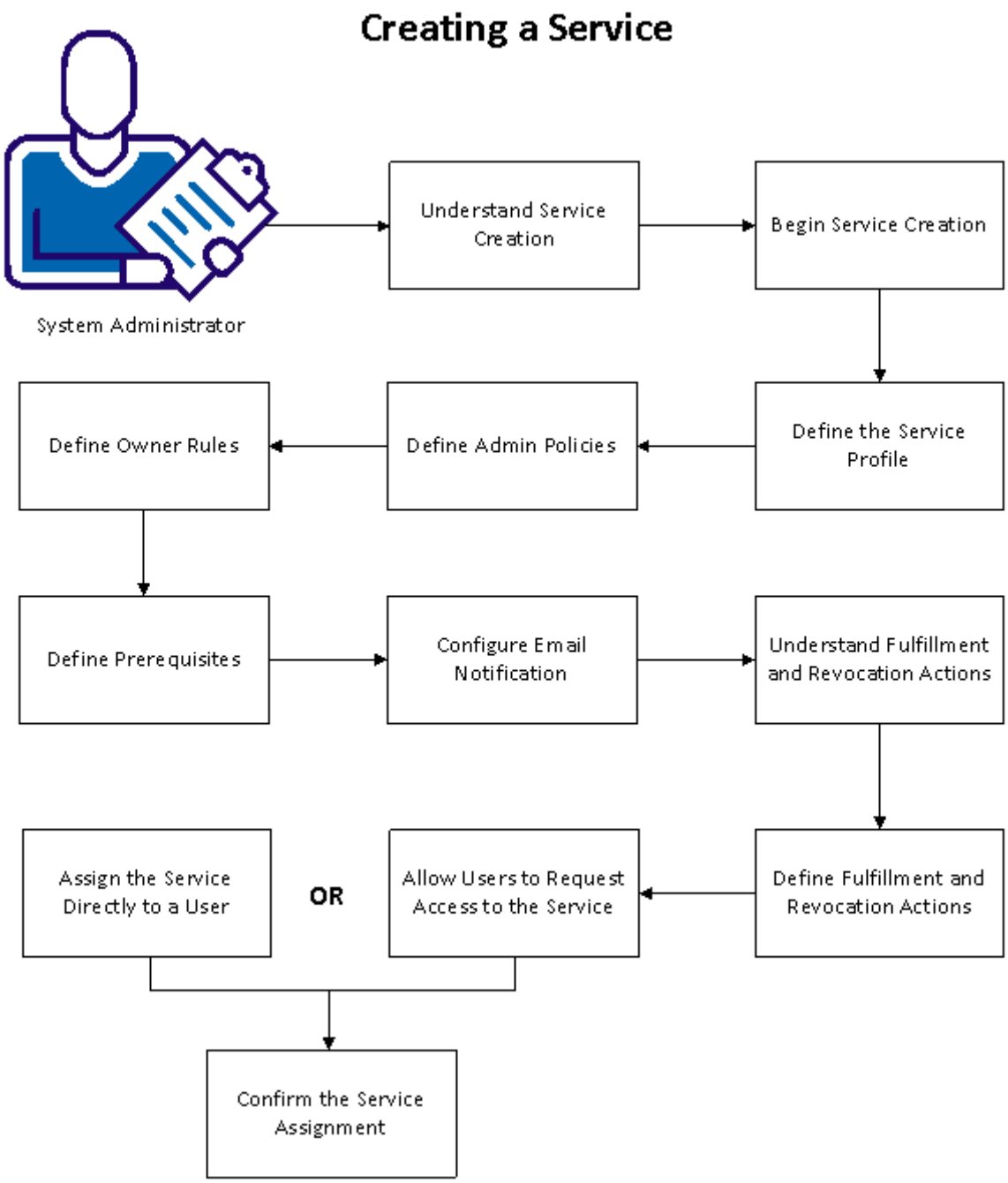
Creating a Service

Services simplify entitlement management. A service bundles together all the entitlements - tasks, roles, groups, and attributes - a user needs for a given business role. Services are available to the user through Access request tasks in the CA CloudMinder User Console. Access request tasks enable a user or administrator to request, assign, revoke and renew a service.

Services allow an administrator to combine user entitlements into a single package, which are managed as a set. For example, all new Sales employees need access to a defined set of tasks and accounts on specific endpoint systems. They also need specific information added to their user account profiles. An administrator creates a service named Sales Administration, containing all the required tasks, roles, groups, and profile attribute information for a new Sales employee. When an administrator assigns the Sales Administration service to a user, that user receives the entire set of roles, tasks, groups and account attributes that are defined by the service.

Another way users can access services is to request access themselves. In the User Console, each user has a list of services available for their request. This list is populated with services marked as "Self Subscribing" by an administrator with the appropriate privileges, typically during service creation. From the list of available services, users can request access to the services they need. When the user requests access to a service, the request is fulfilled automatically, and the associated entitlements are assigned to the user immediately. An administrator with the appropriate privileges can also configure service fulfillment to require workflow approval, or to generate email notifications.

The following diagram shows the information to understand, and the steps to perform, to create a service.


Creating a Service
System Administrator
Understand Service Creation
Begin Service Creation
Define the Service Profile
Define Admin Policies
Define Owner Rules
Define Prerequisites
Configure Email Notification
Understand Fulfillment and Revocation Actions
Define Fulfillment and Revocation Actions
Allow Users to Request Access to the Service
OR
Assign the Service Directly to a User
Confirm the Service Assignment

The following topics explain how to create a service and make it available to users:

1. Understand Service Creation (see page 154).
2. Begin Service Creation (see page 155).
3. Define the Service Profile (see page 155).
4. Define Admin Policies for the Service (see page 157).
5. Define Owner Rules for the Service (see page 157).
6. Define Prerequisites for the Service (see page 158).
7. Configure Email Notification for Service Renewal. (see page 158)
8. Understand Fulfillment and Revocation Actions (see page 159).
9. Define Fulfillment and Revocation Actions for the Service. (see page 160)
10. Allow users to request access to services.

    In the User Console, when the user clicks My Access, then Request & View Access, the user sees a list of services available for their request. The services that appear in this list are those marked "Self Subscribing" by an administrator with the appropriate privileges, typically during service creation.
11. Assign a Service Directly to a User (see page 161).
12. Confirm the Service Assignment.

Understand Service Creation

Before you create a service, consider the prerequisite information and entitlements that are required to create and fulfill the service.

Consider the following questions:

1. What business need does this service address? For example, you can create a service that makes an account on Salesforce.com available to all new employees.
2. Do members of the service need certain admin roles? If so, create or identify those admin roles.
3. Must members of the service receive access to one or more endpoints? If so, create or identify those endpoints.
4. If members of the service need access to endpoints, create or identify the associated provisioning roles and account templates.

5. Must members of the service be members of certain groups? If so, create or identify those groups.
6. Must certain user attributes be referenced or modified when a user becomes a member of the service? For example, when a user receives the Salesforce.com service, is it necessary to confirm whether the department attribute for that user is set to Sales? If so, create or identify those user attributes.

Once you have created or identified these prerequisites, you can begin service creation (see page 155).

Begin Service Creation

You create a service from the User Console.

Follow these steps:

1. Log in to an account that has service management privileges.

   For example, the first user of an environment has the System Manager role, which has the Create Service task.
2. From the navigation menu, select Services, which may be listed under Tasks.
3. Click Manage Services, Create Service.
4. Define the Service Profile.

Define the Service Profile

On the Profile tab, you define basic characteristics of the service.

Follow these steps:

1. Enter a name and tag. A tag is a unique identifier for the service.

   Note: Tags can only contain alphanumeric and underscore characters, and cannot start with a number. Once created, a tagname cannot be changed, or reused, even if a service is later deleted.
2. Select Enabled if you want to make the service available to users as soon as you create it.
3. Select Self Subscribing if you want this service to appear in the list of services available for users to request. When Self-Subscribing is enabled, users can request access to this service through the User Console.

4. (Optional) Add one or more categories. Type a category name and click the up arrow to add it to the service.

   Categories add additional information to a service. You can use this additional information to facilitate service searches in environments that include a significant number of services.

5. Specify a Service Run-time User Data Screen if you want to collect additional user data at the time a user requests the service.

   Use a Service Run-time User Data Screen to help ensure that all user data necessary to fulfill the service exists in the system. For example, a valid email address is required to fulfill a service that creates an account in Google Apps. If an email address for a user does not exist in the CA CloudMinder user store, the user is required to provide it when requesting the service.

   a. Click Browse.

      A list of available profile screens appears. These screens are typically used to collect user data.

   b. Select a profile screen that contains the user data you want to collect. Choose one of the following options:

      - Click Select to collect all user data contained in that screen.

      OR

      - Click Copy to customize the user data you want to collect. Specify a name and unique tag for the new screen. Add, edit or remove user data elements, and click OK.

      OR

      - Click Edit to change the user data contained in that screen. Add, edit, or remove user data elements, and click OK.

      Important! If you edit a user data screen, your changes apply everywhere the screen is used in the User Console. Consider copying and customizing the profile screen instead.

   c. Click Select.

      The user data elements that you selected are collected at the time the user requests the service.

      Note: If the required data exists in the system when a user requests the service, the data is prepopulated in the profile screen.

6. Define Admin Policies for the Service (see page 157).

Define Admin Policies for the Service

On the Administrators tab, you define who can add or remove users as members and administrators of this service. Admin policies contain admin and scope rules and at least one administrator privilege (Manage Members or Manage Administrators).

Admin rules define who can administer this service. Scope rules limit which users can become administrators. For example, an admin rule can allow all members of the Sales group to administer a service. A scope rule can then limit those users to only members of the Sales group in Boston, MA.

Follow these steps:

1. On the Administrators tab, click Add.

   The Admin Policy screen appears.

2. Define an admin rule for which users can administer this service. For example, you can specify users who are members of the Sales group, or who have the specific job title profile attribute of Sales Manager.

   Click the left arrow to edit a previously specified portion of a rule.

3. Define a scope rule to limit which users can administer this service. For example, if you specified users who are members of the Sales group in your admin rule, you can then limit the scope of that rule to only users whose city is Boston, MA.

   Note: You can add several admin policies with different rules and different privileges for each service.

4. If you want to allow administrators to add or remove members of this service, click "Can manage members of this service."

5. Click OK.

6. To edit a policy further, click the Edit icon. To remove a policy, click the minus sign icon.

7. Define Owner Rules for the service. (see page 157)

Define Owner Rules for the Service

On the Owners tab, you define rules about who can be an owner of the service. An owner is a user who can modify the service.

Follow these steps:

1. On the Owners tab, click Add.

   The Owner Rule screen appears.

2. Define an owner rule for which users can own this service. For example, you can specify users who are members of the Sales group, or who have the specific job title profile attribute of Sales Manager.

   Click the left arrow to edit a previously specified portion of a rule.

3. Click OK.

4. Define Prerequisites for the service. (see page 158)

Define Prerequisites for the Service

On the Prerequisites tab, you define services that users must have before requesting this service. A service only appears in the list of available services for a given user if that user is a member of all prerequisite services.

If a duration is set for a prerequisite service, that duration applies to the service you are defining. For example, Service A is a prerequisite for Service B. Service A has a duration of one week. Service B also expires in one week.

Follow these steps:

1. On the Prerequisites tab, click Add Service.

   A search screen appears.

2. Search for a service you want to designate as a prerequisite for this service.

   To display a list of all services over which you have administrative privileges, click Search without modifying the search criteria.

3. Select a service and click Select.

   An updated list of prerequisites for this service appears.

Configure Email Notification for Service Renewal

Some services expire after a certain period.

On the Email tab, you can configure an email notification that reminds service members to renew their membership before it expires. Members can then use the Renew Service task to renew their access.

CA CloudMinder provides a default email template that includes dynamic content. This content is automatically populated when the email is sent. Dynamic content, which appears in curly brackets ({ }) in the email notification editor, adds a specific user name, service name, and expiration date to the email.

You can modify the content of the email notification in the editor. For example, you can modify the body or subject text, change the font, or remove dynamic content.

Note the following items when configuring email notifications:

- If you are including dynamic content in the email notification, do not modify the text between the curly brackets ({ }).
- If the service has a prerequisite service that expires, email notifications are only sent for the prerequisite service, even when email notifications are configured for both services.

Follow these steps:

1. On the Email tab, select the Email notification to users before the service expires check box to enable notifications.
2. (Optional) Customize the email notification using the controls in the editor.

   The email notification editor supports HTML. You can add HTML content to the body of the email notification by clicking the Toggle HTML Source button (<>) in the toolbar.
3. Understand Fulfillment and Revocation actions. (see page 159)

Understand Fulfillment and Revocation Actions

On the Actions tab, you define the entitlements and information - tasks, roles, groups, and attributes - to be added, modified, or removed when a service is assigned or revoked. Put simply, service actions define the actions that a service carries out.

CA CloudMinder uses a Policy Xpress policy to define the circumstances under which Fulfillment and Revocation actions occur. CA CloudMinder preconfigures this policy so that whenever a user requests a service, the proper conditions and data exist. The service is automatically fulfilled or revoked.

An administrator must define the actions that the system takes in fulfilling or revoking a service. For example, when creating a service, an administrator can specify that service members receive the Sales Manager admin role, the Salesforce.com provisioning role, and the Sales group. Likewise, the administrator can specify that these entitlements are removed when the service is revoked.

Define Fulfillment and Revocation Actions for the Service

On the Actions tab, you define the entitlements and information that the system adds, modifies, or removes when a service is assigned to, or removed from, a user.

Follow these steps:

1. Click the Actions tab.

   The Fulfillment and Revocation Actions screen appears.

2. Click the Manage Fulfillment Actions or Manage Revocation Actions button.

   The Create Policy Xpress Policy screen appears.

   The following fields are pre-defined to create an action rule:

   Name

   Provides a friendly name for the action rule. This name must be unique.

   Description

   Defines the meaning of the action rule.

   Priority

   Defines which action rule executes, in the case of several action rules matching. This field is useful for defining default actions. For example, if you have multiple rules, each for a department name, it is possible to set a default by adding an additional rule with no conditions but a lower priority (such as 10 if all others are 5). If none of the department rules are matched, then the default is used.

3. Specify criteria to match under Action Rule Conditions.
4. Click the Add Action when Matched button under Add Actions

   The Add Action when Matched screen appears. On this screen, you define the actions that the system takes when the rule is matched.

5. Enter a friendly name that defines the purpose of the action.

   For example, enter "Add the Sales Manager Admin Role."

6. Select the category of action you want the system to take.

   For example, to add a role, select the Roles category.

7. Select the type of action you want the system to take.

   For example, to add or remove an admin role, select the Set Admin Role type.

8. Select the function that you want the system to perform.

   For example, to add an admin role, select the Add function.

   Note: When you select a function, a description of that function appears. This description can help you determine whether the selected function results in the system behavior you want.

9. Define the specific action that you want the system to take.

   For example, to add an admin role named "Sales Manager", enter the role name, or click the Browse button and select Sales Manager from the list of available admin roles.

10. Click OK.

    Repeat this procedure until you have added all the desired actions for this service.

11. Click OK.

    The system associates the designated fulfillment and revocation actions with the service. When a user receives the service, the associated entitlements and information are added, modified or removed.

12. You can now assign a service to a user (see page 161).

Assign a Service to a User

You can assign a service directly to an individual user. This user becomes a *member* of the service.

Follow these steps:

1. Navigate to Services, Request & View Access.

   A list of services you can administer appears.

2. Select the service that you want to assign to a user and click Select.

   A list of users that are assigned to the service appears.

3. Click Request Access.

4. Search for a user to whom you want to assign the service.

   To display a list of all users for whom you have administrative privileges, click Search without modifying the search criteria.

5. Select a user and click Select.

   An updated list of users that are assigned to the service appears.

6. Click Save Changes.

   The user receives the specified service. The user receives all applications, roles, groups, and attributes you included in the service.

Confirm Service Assignment

Once you have assigned a service to a user, confirm that all tasks associated with the service completed successfully.

Follow these steps:

1. Navigate to Services, View Service Access Request History.

   A search screen appears.

2. Search for the service you assigned to a user.

   To display a list of all services over which you have administrative privileges, click Search without modifying the search criteria.

   A list of services you can administer appears.

3. Select the service that you assigned, and click Select.

   A history of actions that is associated with the service appears.

4. Click Last Changed to see the most recent actions first.
5. Confirm that the user in question received the service successfully.
6. Click Close.

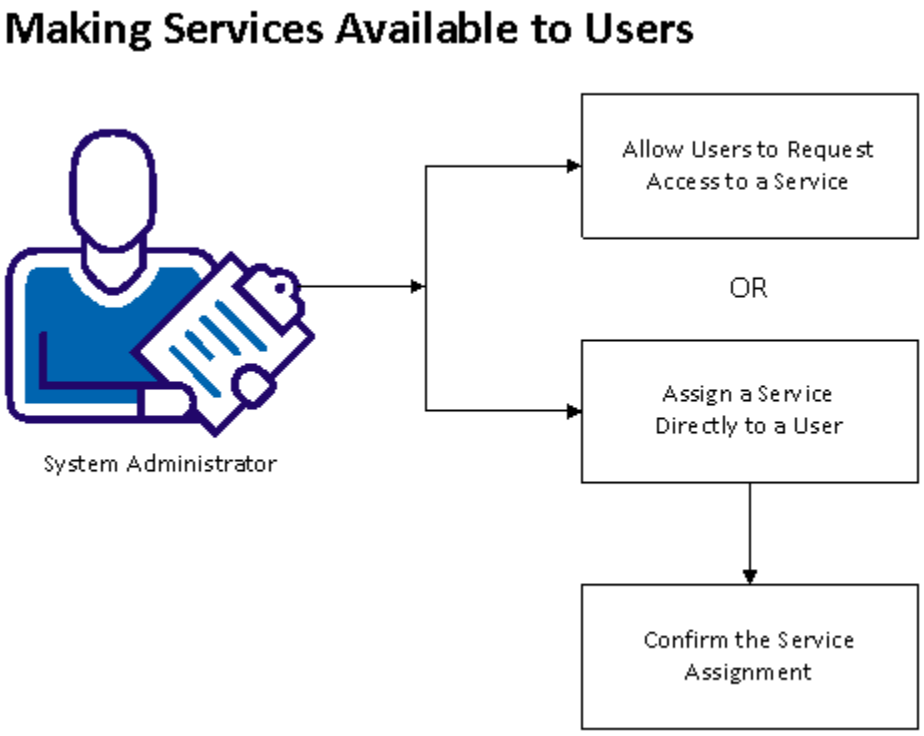
Making Services Available to Users

Services simplify entitlement management. A Service bundles together all the entitlements a user needs for a given business role. Services are available to the user through Access Request tasks in the User Console. Access Request tasks enable a user or administrator to request, assign, revoke and renew a Service through the user interface.

Services allow a system administrator to combine user activities and information - tasks, roles, groups, and attributes - into a single package, which are managed as a set. For example, all new Sales employees need access to a defined set of tasks, accounts on specific endpoint systems, and specific information added to their user account profiles. A system administrator creates a service named Sales Administration, containing all the required tasks, roles, groups, and profile attribute information for a new Sales employee. When an administrator assigns the Sales Administration service to a user, that user receives the entire set of roles, tasks, groups and account attributes that are defined by the service.

Another way users can access services is to request access themselves. In the User Console, each user has a list of services available for their request. This list is populated with services marked as "Self Subscribing" by a system administrator with the appropriate privileges, typically during service creation. From the list of available services, users can request access to the services they need. When the user requests access to a service, the request is fulfilled automatically. The associated tasks, roles, groups and attributes are assigned to the user immediately. A CA CloudMinder administrator with the appropriate privileges can also configure service fulfillment to require workflow approval, or to generate email notifications.

The following diagram shows the information to understand, and the steps to perform, to make services available to users.



You can make services available to users using the following methods:

1. Allow users to request access themselves.

   In the CA CloudMinder User Console, when the user clicks My Access, then Request & View Access, the user sees a list of services available for their request. The services that appear in this list are those marked "Self Subscribing" by a CA CloudMinder administrator with the appropriate privileges, typically during service creation.

   When the user requests access, the system assigns the service to the user. The user receives all applications, roles, groups and attributes associated with the service. If the service includes a Launch Role for an application, an icon and a link to the application appear in the User Console Home page.

2. Assign a Service Directly to a User (see page 161).

3. If you assign a service directly to a user, Confirm the Service Assignment (see page 165).

Assign a Service to a User

You can assign a service directly to an individual user. This user becomes a *member* of the service.

Follow these steps:

1. Navigate to Services, Request & View Access.

   A list of services you can administer appears.

2. Select the service that you want to assign to a user and click Select.

   A list of users that are assigned to the service appears.

3. Click Request Access.

4. Search for a user to whom you want to assign the service.

   To display a list of all users for whom you have administrative privileges, click Search without modifying the search criteria.

5. Select a user and click Select.

   An updated list of users that are assigned to the service appears.

6. Click Save Changes.

   The user receives the specified service. The user receives all applications, roles, groups, and attributes you included in the service.

Confirm Service Assignment

Once you have assigned a service to a user, confirm that all tasks associated with the service completed successfully.

Follow these steps:

1. Navigate to Services, View Service Access Request History.

   A search screen appears.

2. Search for the service you assigned to a user.

   To display a list of all services over which you have administrative privileges, click Search without modifying the search criteria.

   A list of services you can administer appears.

3. Select the service that you assigned, and click Select.

   A history of actions that is associated with the service appears.

4. Click Last Changed to see the most recent actions first.
5. Confirm that the user in question received the service successfully.
6. Click Close.

Modifying a Service

As a system administrator, you can modify a service that you previously created. For example, you can change the entitlements the service grants to service members by adding a role to the service. You can also adjust admin and owner rules for the service, service prerequisites, and other administrative details.

If CA CloudMinder has fulfilled a service for a given user, any changes that are made to the service do not propagate to that user. If you decide to modify a service, users who received the service before you changed it have the original entitlements. Users who receive the service after you change it have the entitlements that the modified service grants. For example, consider the following scenario:

As a system administrator, you create a Sales Manager service that grants the Sales Manager role and the Sales group to service members. Users request the Sales Manager service, and CA CloudMinder fulfills the service by granting the appropriate role and group to the users. You decide to modify the Sales Manager service to include the Employee Manager role. Existing members of the service do not then receive the Employee Manager role. Only new members of the Sales Manager service receive the Employee Manager role, in addition to the Sales Manager role and the Sales group.

Thus, consider modifying a service only if the service has no members. That is, modify a service only if no user has requested and received the service, and no administrator has assigned the service to a user.

You can modify administrative information, admin and owner rules, service prerequisites, and entitlements - tasks, roles, groups, and attributes - for the service.

Follow these steps:

1. Log in to a CA CloudMinder account that has service management privileges.

   For example, the first user of an environment has the System Manager role, which has the Modify Service task.

2. From the navigation menu, select Tasks, Services.
3. Click Manage Services, then Modify Service.

   A search screen appears.

4. Search for a service you want to modify.

   To display a list of all services for which you have administrative privileges, click Search without modifying the search criteria.

5. Select a service and click Select.

   A confirmation message appears.

6. Click Yes.
7. Click Submit.

   CA CloudMinder applies your changes to the service.

Adding a Search to Request and View Access

The Request and View Access task displays a list of services; however, no field exists to search for more services. To add a search field:

1. Select Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for Request and View Access.
3. Select the task in the Service category.
4. Click Tabs.
5. Under tab, click the edit icon to the left of Manage Access.
6. Click Browse on the List Screen line.
7. Configure the option that applies to add the correct search.
8. Select the required Screen and click on Edit button to edit the screen.
9. In Configure Standard List Screen, navigate to 'Select the fields that a user can search on:' section.
10. Select the search fields and configure search field names.
11. Click OK to save the changes.

Information about the Service Request, such as the Service Request Duration and User Data, appears in the Service Request approval workflow item. Also, this information is emailed if you assign the AddServiceToUserEvent policy-based workflow to the Request and View Access task.

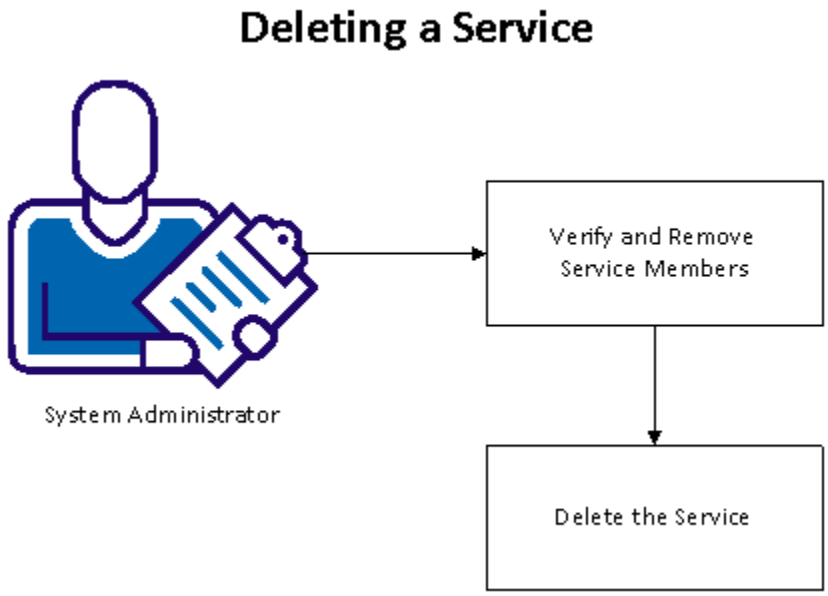
Deleting a Service

As a system administrator, you can delete a service. A deleted service is removed entirely from the system.

If users are assigned to a service, you cannot delete the service. Before you delete a service, first check for and remove all assigned users, or *members*.

Note: Similarly, if a user is a member of a service, you cannot delete the user. First, remove the user as a member of the service, then delete the user.

The following diagram shows the information to understand, and the steps you to perform, to delete a service.



The following topics explain how to delete a service:

1. Verify and Remove Service Members (see page 169)
2. Delete the Service (see page 169)

Verifying and Removing Service Members

Before you delete a service, first check for and remove existing members.

Follow these steps:

1. Log in to a CA CloudMinder account that has service management privileges.

   For example, the first user of an environment has the System Manager role, which has the Modify Service task.

2. Select Tasks, Services, Request & View Access.

   A list of services you can administer appears.

3. Select the service that you want to delete and click Select.

   A list of users that are assigned to the service appears.

4. If the service has members, clear the check boxes next to all users.

5. Click Save Changes.

   A confirmation message appears.

6. Click Yes.

   CA CloudMinder removes the members from the service.

Deleting a Service

You can delete a service that has no service members.

To delete a service

1. Log in to CA CloudMinder with an account that has service management privileges.

   For example, the first user of an environment has the System Manager role, which has the Delete Service task.

2. Navigate to Services from the left pane or by selecting Tasks..

3. Click Manage Services, Delete Service.

   A search screen appears.

4. Search for the service you want to delete.

   To display a list of all services for which you have administrative privileges, click Search without modifying the search criteria.

5. Select the service and click Select.

   A confirmation message appears.

6. Click Yes.

   The service is deleted.

Renewing Access to a Service

Some services expire after a certain period of time. Administrators can renew a service for users to prevent an interruption in their access.

You can renew a service using one of the following methods:

- Select the service, then select the user access to renew
- Select the user, then select the service to renew

Note: Depending on how an environment is configured, end users can also renew their access by using the Renew Access task.

The following procedure describes how to renew access by selecting the service first. If you want to select the user first, use the User Access Requests, Manage User Renew Requests task in the Users category.

Follow these steps:

1. Click Services, Renew Access in the User Console.
2. Search for and select the service that you want to renew.

   The User Console displays a list of users who currently have access to the service you selected, and the date their access expires.

3. Select the duration for the renewal in the Access Request column, then click OK.

   The options in the Duration field are determined when the service is created.

4. Click Save Changes.

You can view the status of the service renewal by using the View Access Request History in the User Console.

Chapter 8: Synchronization

This section contains the following topics:

User Synchronization between Servers

You configure synchronization in Identity Management to make sure that the users for the Identity Management user store and provisioning directory have matching data. To handle changes from either directory or user store, you configure inbound and outbound synchronization.

Inbound Synchronization

Inbound synchronization keeps Identity Management users up to date with changes that occur in the provisioning directory. Changes in the provisioning directory include those made using systems with connectors to the Provisioning Server. The synchronization uses the mappings defined on the Provisioning screen of the Management Console.

Failover for Inbound Synchronization

Fail over to an alternate Identity Management Server URL occurs only if the application server named by a URL is not running. If the application server is running and accepts the notification but then encounters a configuration error, such as unknown environment or environment not started, these errors block the delivery of notifications. These problems must be resolved before inbound notifications functions properly.

Outbound Synchronization

Outbound synchronization involves using Identity Management to create and update users in the provisioning directory.

Creating Provisioning Directory Users

User creation in the provisioning directory occurs only for provisioning related events, such as assigning a provisioning role to a user. A user is created in the provisioning directory *only* when you use an admin task that assigns a role to create the user.

Note: A Provisioning Directory user is also called a global user. A global user is the single user that connects endpoint accounts.

When user creation in Identity Management triggers user creation in the provisioning directory, Identity Management sends an email with a temporary password to the new user's email address as it is defined in the provisioning directory. The user can log in to the User Console with that password, however, the user must then change the password. As a result, the password is synchronized between the user store and provisioning directory.

If the user has no email address, the user cannot access the User Console until changing the password in the user store.

Note: To email a temporary password, email notifications must be enabled for the Environment, and the CreateProvisioningUserNotificationEvent must be configured for email notification. (See the *Configuration Guide*.)

Update Global Users using Identity Management

Updates to users in the provisioning directory occur when you use an admin task that modifies users. If no global user exists, no synchronization occurs.

Outbound mappings match the Identity Management user events to an outbound event that affects the provisioning directory.

Identity Manager User Event	Outbound Event
☐ DeleteUserEvent	POST_DELETE_GLOBAL_USER
☐ DisableUserEvent	POST_DISABLE_GLOBAL_USER
☐ EnableUserEvent	POST_ENABLE_GLOBAL_USER
☐ ModifyUserEvent	POST_MODIFY_GLOBAL_USER
☐ ResetPasswordEvent	POST_CHANGE_GLOBAL_USER_PWD

If a user exists in the provisioning directory but not in Identity Management, you can create that user in the User Console. If you have mapped attributes for the create task and the users have the same user ID, the attributes for the provisioning user are updated in the provisioning directory. Now you can manage that user from the User Console.

Note: If an event updates user attributes and you want the values to be synchronized to Identity Management, then you need to map the events to the Outbound Event: POST_MODIFY_GLOBAL_USER.

Delete Global Users using Identity Management

By default, outbound synchronization is configured for the Delete User event. When you delete a user in Identity Management, the user is also deleted in the provisioning directory and all endpoint accounts.

If Identity Management cannot delete a user's account in a managed endpoint, it deletes the user from the remaining accounts, but does not delete the user from the provisioning directory.

For example, suppose User A has a UNIX account and an Exchange account, which are managed in the Provisioning Server. When user A is deleted in Identity Management, the Provisioning Server attempts to delete the user's accounts. If the Provisioning Server cannot delete the Exchange account due to a communication error, it deletes user A's UNIX account, but does not delete the user from the provisioning directory. However, User A is not restored in the user store.

Enable Password Synchronization

The Provisioning Server allows password synchronization between Identity Management users and associated endpoint user accounts. Two configurations are required to enable endpoint initiated changes:

- Endpoints must be configured to capture endpoint-initiated changes and forward the changes to the Provisioning Server.
- The Enable Password Synchronization Agent attribute should be activated for the Global User.

Follow these steps:

1. In the Management Console, choose Advanced Settings, Provisioning.
2. Check Enable Password Changes from Endpoint Accounts.
3. Click Save.
4. Restart the Application Server.

Synchronize Users in Create or Modify User Tasks

On the profile tab of a task that creates or modifies users, synchronization controls ensure that changes to the Identity Management are also made to the global user. If you create admin tasks that create or modify users and you have Identity Policies, set the synchronization controls as follows:

- Set User Synchronization to On Task Completion.
- Set Account Synchronization to On Task Completion.

Note: For best performance, select the On Task Completion option. However, if you select the On Task Completion option for a task that includes multiple events, Identity Management does not synchronize until all of the events in the task complete. If one or more of those events require workflow approval, this may take several days. To prevent Identity Management from waiting to apply identity policies or synchronize accounts until all events complete, select the On Every Event option.

If you add attributes to admin tasks that manage users, you need to update the Attribute Mappings in the Provisioning screen in the Management Console. For each user attribute in Identity Management, a default provisioning attribute exists.

User Attribute	Provisioning Attribute
%ADMIN_ROLE_CONSTRAINT%	%ADMIN_ROLE_CONSTRAINT%
%EMAIL%	%EMAIL%
%ENABLED_STATE%	%ENABLED_STATE%
%FIRST_NAME%	%FIRST_NAME%
%FULL_NAME%	%FULL_NAME%
%IDENTITY_POLICY%	%IDENTITY_POLICY%
%LAST_NAME%	%LAST_NAME%
%PASSWORD%	%PASSWORD%
%PASSWORD_DATA%	%PASSWORD_DATA%
%USER_ID%	%USER_ID%

Synchronization Tasks

You can perform the following types of synchronization:

User Synchronization

Ensures that each user has the necessary accounts on the appropriate managed endpoints, and that each account is assigned to the appropriate account templates as called out by the user's provisioning roles.

Account Synchronization

Ensures that the capability attribute values on accounts are the appropriate values as indicated by the account's assigned account templates. Account synchronization can be strong or weak. Weak synchronization ensures that accounts capability attributes have at least the minimum capability required by its account templates. Strong synchronization ensures that account capability attributes have the exact capability required by its account templates. Account synchronization is strong if the account belongs to at least one account template whose Strong Synchronization check box is selected.

No corresponding Strong Synchronization check box governs User Synchronization, but a similar concept exists. When you issue the Synchronize User with Roles menu item on a user, you are presented with two synchronization options:

- Add missing accounts and account template assignments.
- Delete extra accounts and account template assignments.
- By selecting only the Add check box, which is similar to Weak Account Synchronization, you want global users to have at a minimum all accounts required by their assigned provisioning roles, but you allow users to have additional accounts not prescribed by current provisioning roles.

Select both the Add and Delete check boxes, which is similar to Strong Account Synchronization, to have the provisioning roles define exactly which accounts the user should have. Any additional accounts are deleted.

Choose Weak/Strong Account Synchronization or Weak/Strong User Synchronization based on how precisely provisioning roles are defined. If your users fit into clearly-defined provisioning roles where account access is tied to those roles, you would use Strong Synchronization.

Note: Some endpoint types set strong synchronization as the default. For more information, see the CA Identity Management and Governance Connectors wiki.

User synchronization and account synchronization are separate tasks that you must perform individually. Typically, you perform user synchronization first to ensure that all necessary accounts are created, then perform account synchronization later so the Provisioning Server assigns or changes the values of the account attributes.

The Provisioning Server provides two sets of synchronization menu options for objects:

- Check synchronization menu options verify the synchronization and return a list of the accounts that do not comply with the provisioning roles or account templates.
- Synchronize menu options synchronize global users with their provisioning roles or accounts with their account templates.

If you perform the check synchronization functions first, the Provisioning Server tells you what corrections the synchronize functions will perform. If the check synchronization functions find no problem, the synchronize functions do not run.

User Synchronization

User synchronization creates, updates, or deletes accounts so they comply with the provisioning role assigned to a user. So if administrators add or delete accounts on your managed endpoint by using native tools, and you have not performed a recent re-exploration of your endpoint to update the provisioning directory, User Synchronization may indicate no problems exists when actually a user may have extra or missing accounts.

Why Users Become Out of Sync

The following are some reasons why users become out of sync with their provisioning roles or account templates:

- Earlier attempts to create the necessary accounts failed due to hardware or software problems in your network, thereby causing missing accounts.
- Provisioning roles and account templates may have changed, thereby creating extra or missing accounts.
- Accounts were assigned to account templates after they were created, so accounts exist that have not been synchronized with their account templates.
- The creation of a new account is delayed because the account was specified to be created later.
- A new endpoint was acquired. During exploration and correlation, the Provisioning Server does not assign provisioning roles to the users automatically, so you must update the role to indicate which users should have accounts on the new endpoint. Any account that was correlated to a user is listed as an extra account when the user is synchronized.
- An existing account was assigned to a user by copying the account to the user, thereby performing a manual correlation and establishing an extra account.
- An account was created for a user other than by assigning the user to a role. For example, if you copy a user to an account template that is not in any of the user's provisioning roles, the account is listed as an extra account or as an account with an extra account template. If you copy the user to an endpoint to create an account using the endpoint's default account template, that account could be an extra account.

User with Roles Synchronization

You can check synchronization on users to list extra accounts or account templates and accounts that are missing. When you request to synchronize user with roles, the Provisioning Server ensures that the user has all the accounts required by the person's provisioning roles and ensures each account belongs to the correct account templates.

- With this task, you can select a checkbox to create the account on the endpoint. If more than one account template in the user's provisioning roles prescribes the same account, the account is created by merging all relevant account templates.
- During user with roles synchronization, you have the option to delete extra accounts. You may determine that your users have legitimate reasons for having accounts other than those required by their provisioning roles. If that is the case, you should not select this delete option.

  If an account being deleted resides in a managed endpoint for which account deletions have been disabled, the account is not actually deleted.

Create Accounts

Because provisioning roles contain account templates, and account templates are associated to endpoints, a user should have accounts listed on each endpoint with the correct account attributes.

With this task, you can select a checkbox to create the account on the endpoint. If more than one account template in the user's provisioning roles prescribes the same account, the account is created by merging all relevant account templates.

This account is assigned to those account templates, which are currently not synchronized with the account. Account synchronization is not necessary on newly created accounts.

Delete Accounts

During user with roles synchronization, you have the option to delete extra accounts. You may determine that your users have legitimate reasons for having accounts other than those required by their provisioning roles. If that is the case, you should not select this delete option.

If an account being deleted resides in a managed endpoint for which account deletions have been disabled, the account is not actually deleted.

Add Account Templates to Accounts

If an account is missing one or more account template assignments, user with account templates synchronization assigns an existing account to those Account Templates. When an account is assigned to one or more new Account Templates, account synchronization is run automatically to update the capability attributes of the account to capabilities specified by the Account Templates.

After account update from user with account templates synchronization, the account may or may not be in sync with its Account Templates. If one of the Account Templates added was a strong synchronization account template or if two or more Account Templates were added to an account, user with roles synchronization will start a full account synchronization on the account. However, if only one weak synchronization account template was added, user synchronization with account templates synchronization starts an account synchronization involving only this one account template. If the account was previously out of account synchronization with its other Account Templates before this update, it could still be out of account synchronization afterwards.

Removing Account Templates from Accounts

User with roles synchronization can also be used to remove extra account templates from an account. This is only done if you select the delete option. When user synchronization determines that an account needs to be updated to remove one or more extra account templates, account synchronization is run automatically on the account to synchronize its capability attributes with the account templates remaining on the account.

This account synchronization that occurs when removing account templates from an account will use strong synchronization if any of the remaining account templates is marked for strong synchronization and weak synchronization if all of the remaining account templates are marked for weak synchronization.

Whether weak or strong synchronization is used affects whether account capabilities granted earlier when an account template was assigned to an account are taken away when that that account template is later removed. With strong synchronization, a capability granted by an account template, such as a group membership or higher quota, will be taken away (group membership removed or quota lowered) if none of the account templates remaining on the account prescribe that capability. However, with weak synchronization, typically the account is unchanged because the Provisioning Server does not distinguish between on-demand extra capabilities and capabilities granted through account templates.

The exception to this rule is for certain multivalued capability attributes designated as SyncRemoveValues attributes. A simple multivalued attribute representing a collection of values assigned to the account (a group membership list, say), will typically be listed as a SyncRemoveValues attribute. For these attributes, the weak synchronization action that occurs while removing an account template from an account will remove values prescribed by the account template that is being removed - as long as that value is not also prescribed by one of the remaining account templates.

For example, if you create your account templates where each account template assigns a unique group membership to your account, this SyncRemoveValues feature will mean that when you change a global user's provisioning roles so as to no longer require a particular account template, the account will be updated to no longer belong to the group prescribed by that account template. You will note that this is not exactly the same as strong synchronization, as group memberships given to accounts beyond what is prescribed to account templates are retained.

For all single-valued attributes and certain multivalued attributes which are not designated as SyncRemoveValues attributes, the weak synchronization action while removing an account template from an account is the same as a normal weak synchronization action - capabilities are never removed.

If you want the capabilities never to be removed by weak synchronization, disable the SyncRemoveValues feature by setting the domain configuration parameter Synchronize/Remove Account Template Values from Accounts to No.

Account Template Synchronization

Changes that you make to account templates affect existing accounts as follows:

- If you change the value of a capability attribute, the corresponding account attribute is updated, if necessary, to be in synchronization with the account template attribute value. See the description of weak and strong synchronization.
- Certain account attributes are designated by the connector as not being updated on account template changes. Examples are certain attributes that the endpoint type only allows to be set during account creation, and the Password attribute.

Which Attributes are Updated

When you change capability attributes in an account template, the corresponding attribute on the accounts change. This change has an impact on the attributes for the account. The impact is based on the following factors:

- Whether the account template is defined to use weak or strong synchronization.
- Whether the account belongs to multiple account templates.

Weak Synchronization

Weak synchronization ensures that users have the minimum capability attributes for their accounts. Weak synchronization is the default in most endpoint types. If you update a template that uses weak synchronization, Identity Management updates capability attributes as follows:

- If a number field is updated in an account template and the new number is greater than the number in the account, Identity Management changes the value in the account to match the new number.
- If a check box was not selected in an account template and you subsequently select it, Identity Management updates the check box on any account where the check box is not selected.
- If a list is changed in an account template, Identity Management updates all accounts to include any value from the new list that was not included in the account's list of values.

If an account belongs to other account templates (whether those templates use weak or strong synchronization), Identity Management consults only the template that is changing. This action is more efficient than checking every account template. Because weak synchronization only adds capabilities to accounts, it generally is not necessary to consult those other account templates.

Note: When propagating from a weak synchronization account template, changes that would remove or lower capabilities could leave some accounts unsynchronized. Remember that with weak synchronization, capabilities are never removed or lowered. Without consulting other templates for an account, the propagation does not consider if weak synchronization is sufficient.

In this situation, use Synchronize Users with Account Templates to synchronize the account with its account templates.

Strong Synchronization

Strong synchronization ensures that accounts have the exact account attributes that are specified in the account template.

For example, suppose that you add a group to an existing UNIX account template. Originally, the account template made accounts members of the Staff group. Now, you want to make the accounts members of both the Staff and System groups. All accounts that are associated with the account template are considered synchronized when each account is a member of the Staff and System groups (and no other groups). Any account not in the Staff group is added to both groups.

Some other factors to consider include the following situations:

- If the account template uses strong synchronization, any account belonging to groups, other than Staff and System, are removed from those extra groups.
- If the account template uses weak synchronization, the accounts are added to the Staff and System groups. Any account that has additional groups that are defined to it remains a member of these groups.

Note: Synchronize accounts with their templates regularly to ensure that the accounts stay synchronized with their account templates.

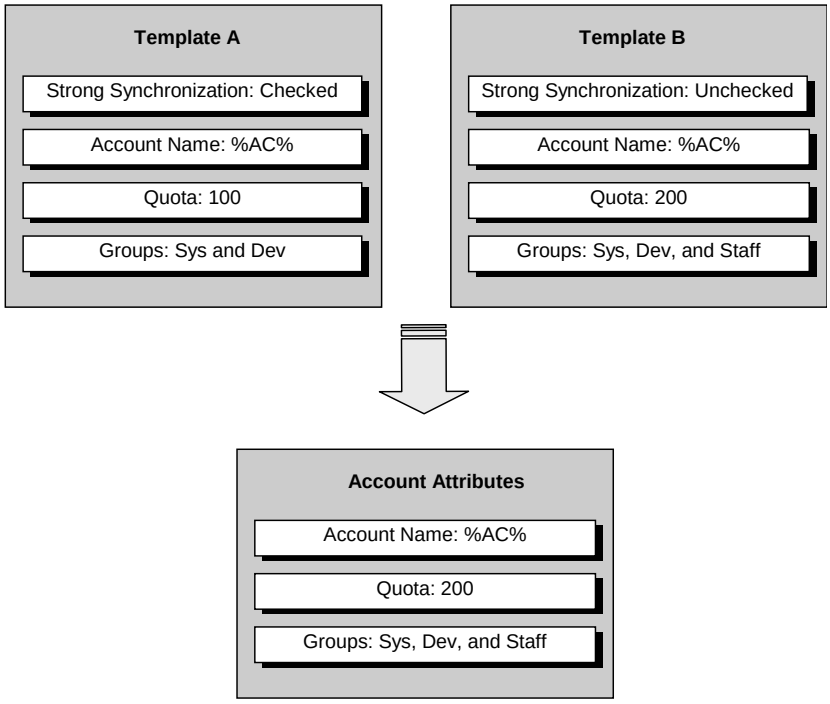
Accounts with Multiple Templates

Synchronization also depends on whether the account belongs to more than one account template. If an account has only one account template and that template uses strong synchronization, each attribute is updated to exactly match what the account template attribute value evaluates to. The result is the same as if the attribute were an initial attribute.

An account may belong to multiple Account Templates, as would be the case if a user belonged to multiple provisioning roles each of which prescribed some level of access on the same managed endpoint. When this happens, Identity Management combines those account templates into one effective account template that prescribes the superset of the capabilities from the individual account templates. This account template is itself considered to use weak synchronization if all its individual account templates are weak or strong synchronization if any of the individual account templates is strong.

Note: Often you use only weak synchronization or only strong synchronization for the account templates controlling one account, depending on whether your company's roles completely define the accesses your users need. If your users do not fit into clear roles and you need the flexibility to grant additional capabilities to your user's accounts, use weak synchronization. If you can define roles to exactly specify the accesses your users need, use strong synchronization.

The following example demonstrates how multiple account templates are combined into a single effective account template. In this example, one account template is marked for weak synchronization and the other for strong synchronization. Therefore, the effective account template created by combining the two account templates is treated as a strong synchronization account template. The integer Quota attribute takes on the larger value from the two account templates, and the multivalued Groups attribute takes on the union of values from the two polices.



Attributes Only for New Accounts

In an account template, certain attributes are only applied when creating the account. For example, the Password attribute is a rule expression that defines the password for new accounts. This rule expression never updates the password of an account. Changes to the password rule expression only affect accounts that are created after the rule expression was set.

Similarly, a template rule expression for a read-only account attribute affects only accounts that are created after the rule expression was set. Changing it has no effect on existing accounts.

Account Synchronization

Account synchronization updates capability attributes to ensure that the account has the capabilities specified by the account templates.This synchronization does not affect the account's initial attributes.

To synchronize capability attribute changes in an account template with its accounts, use one of the synchronization menu options discussed in this section.

Check Account Synchronization

You can check account synchronization for endpoints and users. This action returns a list of accounts that do not comply with account templates. The following table describes what happens when you check the synchronization of accounts on each object:

Object	Synchronizes
Endpoint	Account attributes for each account on an endpoint and ensures they comply with associated account templates.
Global user	Account attributes for each of a user's accounts and ensures they comply with associated account templates.

Synchronize Accounts

You can perform account synchronization on endpoints, users, and account templates. The following table lists the effect of account synchronization on each object:

Object	Synchronizes
Endpoint	Each account on an endpoint with its associated account templates.
Global user	Each account of a global user with each account template associated to it.

Chapter 9: Identity Policies

This section contains the following topics:

Identity Policies

An Identity policy is a set of business changes that occur when a user meets a certain condition or rule. You can use identity policy sets to:

- Automate certain identity management tasks, such as assigning roles and group membership, allocating resources, or modifying user profile attributes.
- Enforce segregation of duties. For example, you can create an identity policy set that prohibits members of the Check Signer role from having the Check Approver role, and restricts anyone in the company from writing a check over $10,000.
- Enforce compliance. For example, you can audit users who have a certain title and make more than $100,000.

  Identity policies that enforce compliance are called *compliance policies*.

The business changes associated with an identity policy include:

- Assigning or revoking roles, including provisioning roles (if you are using a provisioning directory only)
- Assigning or revoking group membership
- Updating attributes in a user profile

For example, a company may create an identity policy which states that all Vice Presidents belong to the Country Club Member group and have the role Salary Approver. When a user's title changes to Vice President and that user is synchronized with the identity policy, Identity Management adds the user to the appropriate group and role. When a Vice President is promoted to CEO, she no longer meets the condition in the Vice President identity policy so the changes applied by that policy are revoked, and new changes based on the CEO policy are applied.

The change actions that occur based on an identity policy contain events which can be placed under workflow-control and audited. In the previous example, the Salary Approver role grants significant privileges to its members. To protect the Salary Approver role, the company can create a workflow process that requires a set of approvals before the role is assigned, and they can configure Identity Management to audit the role assignment.

To simplify identity policy management, Identity policies are grouped in an identity policy set. For example, the Vice President and CEO policies may be part of the Executive Privileges identity policy set.

Note: Identity Management includes an additional type of identity policy, called a *preventative identity policy* (see page 207). These policies, which execute before a task is submitted, allow an administrator to check for policy violations before assigning privileges or changing profile attributes. If a violation exists, the administrator can clear the violation before submitting the task.

Identity Policy Set Planning Worksheet

An identity policy set contains one or more identity policies. Before you create an identity policy set, use the following worksheet to plan each identity policy in the set.

Question	Your Response
What name do you want to give the identity policy?	
Which users does the identity policy apply to?	
When an identity policy is applied to a user, what actions should Identity Management perform?	
When an identity policy that once applied to a user no longer applies, what actions should Identity Management perform?	
Should Identity Management apply the changes in an identity policy multiple times or only the first time a user meets the conditions in the policy?	

After you complete this worksheet for each identity policy in a policy set, verify that the policies do not conflict with other policies. For example, make sure that a policy does not grant a privilege that another policy revokes.

Create an Identity Policy Set

To create an identity policy set, you must have the System Manager role, or a role that includes the Create Identity Policy Set task.

To create an identity policy set, complete the following steps:

1. Define the Profile for the Identity Policy Set (see page 187)
2. Create a Policy Set Member Rule (see page 188)
3. Create an Identity Policy (see page 188)
4. Specify Owners for the Identity Policy Set (see page 197)

Note: To use policies for Identity Management environment, enable identity policies in the Identity Management Management Console. See the *Configuration Guide* for more information.

Define the Profile for the Identity Policy Set

The Profile tab allows you to define basic properties for an identity policy set.

To define an identity policy set profile

1. Select Policies, Manage Identity Policies, Create Identity Policy Set from the User Console.

   You must be logged in to Identity Management as a user with privileges to manage identity policies. The default System Manager role includes these privileges.
2. Choose to create a new identity policy set or create a copy of an existing identity policy set.
3. Enter a name for the identity policy set.
4. Enter a category for the identity policy set.

   The category groups identity policy sets with similar purposes for reporting. The Category field is required.
5. Optionally, enter a description for the identity policy set.
6. If you do not want to make the identity policy set available for use, clear the Enabled check box.
7. When you have completed the Profile tab, select the Policies tab to create the identity policies for the identity policy set.

More information:

Create an Identity Policy (see page 188)
Create a Policy Set Member Rule (see page 188)

Create a Policy Set Member Rule

You can create a member rule for a policy set, so that the policy set applies only to certain users. The rule is evaluated before evaluating identity policies in the set, which can save significant time. For example, if a member rule limits the identity policy evaluation to 10 percent of users, it saves 90 percent of the evaluation time.

To create a Policy Set Member Rule

1. Select the Policies tab.
2. Click the Edit symbol under Policy Set Member Rule.
3. Enter a rule to apply the policy to only certain users.
4. Click OK.

More Information:

Create an Identity Policy (see page 188)

Create an Identity Policy

After you define the profile and member rule for the Identity Policy Set, you can define the identity policies in that policy set.

Note: In large implementations, it may take significant time to evaluate identity policy rules. To reduce the evaluation time for rules that include user-attributes, you can enable the in-memory evaluation option. For more information, see the *Configuration Guide*.

To create an identity policy

1. Select the Policies tab.
2. Click Add.
3. Enter a name for the identity policy.
4. Select the Apply Once check box if you want to apply the policy only when a user first meets the policy.
5. Select the Compliance check box to flag this policy as a compliance policy.

   If this check box is selected:

   - Identity Management can generate reports for users that are not synchronized with compliance policies.
   - The Compliance Violation action is visible in the Action on Apply/Remove Policy list box.
6. Identify the users to which the policy applies in the Policy Condition section.

7. In the Action on Apply Policy section, define the actions that Identity Management takes when the identity policy is applied to a user.
8. In the Action on Remove Policy section, define the actions that Identity Management takes when a user no longer meets the conditions for the identity policy.
9. Click OK.

Note: Before you can use the identity policy set that you created, enable identity policies in the Management Console. See the *Configuration Guide* for more information.

The Apply Once Setting

Identity Management applies an identity policy differently, based on the Apply Once setting.

Enabling the Apply Once Setting

If the Apply Once setting is enabled, Identity Management applies the changes associated with the identity policy when a user *first* meets the condition defined in the policy. The change actions associated with the policy occur only once. Therefore, Identity Management does not apply policy updates to users, if the policy was previously applied.

When a user no longer meets the condition defined in the policy, Identity Management executes the policy's remove actions.

The Apply Once setting is typically used when provisioning resources. For example, you may have a policy that assigns a cell phone to managers. When a user first becomes a manager, that user is assigned a cell phone. Identity Management only issues the cell phone once, not each time the policy is evaluated. If the cell phone policy is updated to include a newer cell phone model, Identity Management does not issue new cell phones to existing managers.

Note: Resource provisioning is available when Identity Management integrates with a Provisioning Server.

Disabling the Apply Once Setting

If the Apply Once setting is not enabled, the change actions associated with the identity policy are applied each time an identity policy is evaluated. This means that Identity Management applies change actions for every user who meets the condition in the policy, regardless of whether the change actions were applied previously.

Typically, you disable the Apply Once setting in an identity policy that enforces compliance. For example, you can create an identity policy that restricts managers' spending authority to $5,000. If Identity Management encounters a manager whose spending authority is set to $10,000, it resets the spending authority to $5,000. Each time a manager is synchronized with the identity policy, Identity Management checks to make sure the spending authority is set correctly.

If a manual change that conflicts with a change action is made to a user profile, Identity Management overwrites the change when the user is synchronized with the policy.

In the previous example, if someone manually increases a manager's spending authority to $10,000, Identity Management resets the spending authority to $5,000 when the manager is synchronized with the policy.

The following table summarizes the effects of enabling or disabling the Apply Once setting.

If Apply Once is...	**Then...**
Enabled	■ Change Actions associated with the identity policy are applied only once ■ Manual changes made after the identity policy is applied are preserved ■ Updates are not applied to users who meet the condition in an identity policy, if Identity Management applied the policy previously ■ When a user no longer meets the condition in an identity policy, Identity Management executes the remove actions
Disabled	■ Change actions associated with the identity policy are applied every time a user is synchronized with the policy ■ Manual changes are overwritten when the identity policy is applied ■ Updates to the policy are applied when a user is synchronized ■ When a user no longer meets the condition in an identity policy, Identity Management executes the remove actions

Policy Conditions

Policy conditions are the rules that determine the set of users to which an identity policy applies.

The following table describes the available options.

Syntax	Condition	Example
(all)	The identity policy applies to all users.	
where <user-filter>	The user must match one or more attribute values.	Users where title=manager and locality=east
in <org-rule>	The user must belong to named organizations. **Note:** When you select this option, Identity Management displays a new list box where you can select the following options: ■ organization <organization> [and lower]-- Use an organization search screen to select an organization and, optionally, include the organization's child organizations. ■ Organizations where <org-filter> [and lower]--Specify a filter that selects one or more organizations.	Users in organization sales and lower
where <user-filter> and who are in <org-rule>	The user must match specific user attributes and belong to a specific organization.	title=manager and organization=Sales*

Syntax	Condition	Example
who are members of <group-member-rule>	The user must belong to a group which meets a condition specified by attributes on the group. **Note:** When you select this option, Identity Management displays a new list box where you can select the following options: ■ group <group>--Use a group search screen to select a group. ■ group where <group-filter>--Specify a filter that selects one or more groups.	Users who are members of groups where owner=CIO
who are members of <role-rule>	The user must be a member of a role. The role can be an: ■ access role ■ admin role ■ provisioning role **Note:** To use provisioning roles, Identity Management must integrate with a Provisioning Server. See the *Installation Guide* for more information.	Users who are members of the Help Desk role
who are administrators of <role-rule>	The user must an administrator for a role. The role can be an: ■ access role ■ admin role ■ provisioning role **Note:** To use provisioning roles, Identity Management must integrate with a Provisioning Server. See the *Installation Guide* for more information.	Users who are administrators of the Sales Manager role

Syntax	Condition	Example
who are owners of <role-rule>	The user must be an owner for a role. The role can be an: ■ access role ■ admin role ■ provisioning role **Note:** To use provisioning roles, Identity Management must integrate with a Provisioning Server. See the *Installation Guide* for more information.	Users who are owners of the User Manager role
returned by the query <LDAP-query>	The user must meet a condition based on an LDAP query.	User who meet the conditions of an LDAP query. For example: (departmentNumber= Accounts)
in <administrative-union-constraint>	The user must meet at least one of the conditions in a list of conditions. You can include the following types of filters in an administrative union constraint: ■ Member of access/admin/provisioning role ■ Administrator of access/admin/provisioning role ■ owner of access/admin/provisioning role ■ member of a group	Users who are a member of the Certify Manager role, *or* who are an owner of the Certify Manager role.

Syntax	Condition	Example
in <administrative-intersection-constraint>	The user must all of the conditions in a list of conditions. You can include the following types of filters in an administrative union constraint: ■ Member of access/admin/provisioning role ■ Administrator of access/admin/provisioning role ■ owner of access/admin/provisioning role ■ member of a group	Users who are members of the Contract Initiator role *and* the Contract Approver role.

Actions on Apply/Remove Policies

You can define change actions that Identity Management performs when it evaluates the identity policy. The actions include:

Actions on Apply Policy

A set of actions that Identity Management performs when a user meets the conditions in the policy conditions.

Actions on Remove Policy

A set of actions that Identity Management performs when a user no longer meets the conditions in the policy conditions.

The actions that Identity Management can perform when identity policies are applied or removed are the same. See the following table for more information.

Change Action	Description
Add to group <group-name> [...]	Adds users to a group. When you select this option, Identity Management presents a screen where you can search for the group you want.
Add to <group-name> in user's organization	Adds users to a local group. When you select this option, Identity Management presents a text box where you can enter the name of the group that you want.

Change Action	Description
Set <single-value-user-attribute> to value	Sets the value of an attribute in a user profile. If there is an existing value, Identity Management overwrites it with the value specified in the change action.
Add <value> to <multi-value-user-attribute>	Adds a value to a multi-value user attribute. This option does not overwrite existing values.
Make member of access role	Assigns users to an access role.
Make administrator of access role	Make users administrators of an access role
Make member of admin role	Makes users members of an admin role
Make administrator of admin role	Makes users administrators of an admin role
Make member of provisioning role	Makes users members of a provisioning role, which creates associated endpoint accounts. **Note:** To use provisioning roles, Identity Management must integrate with a Provisioning Server. See the *Installation Guide* for your application server.
Make administrator of provisioning role	Makes users administrators of a provisioning role. **Note:** To use provisioning roles, Identity Management must integrate with a Provisioning Server. See the *Installation Guide* for your application server.
Remove from group <group-name> [...]	Removes users from a group. When you select this option, Identity Management presents a screen where you can search for the group you want.
Remove from <group-name> in user's organization	Removes users from a local group. When you select this option, Identity Management presents a text box where you can enter the name of the group that you want.
Remove <value> from <multi-value-user-attribute>	Removes a value from a multi-value user attribute.
Remove member from access role	Revokes an access role.

Change Action	Description
Remove administrator from access role	Revokes administrator privileges for a specific access role
Remove member from admin role	Revokes an admin role.
Remove administrator from admin role	Revokes administrator privileges for a specific admin role
Remove member from provisioning role	Revokes a provisioning role.
Remove administrator from provisioning role	Revokes administrator privileges for a specific provisioning role.
Send audit message	Sends a message that you create to the audit database. This message may appear in a report that you create.
Compliance violation	Sends a message that you create to the audit database. If you create a compliance report, the message appears each time the identity policy is applied/removed from a user. See the *Configuration Guide* for more information about auditing. **Note:** You must enable the Compliance check box on the Profile tab for the Identity Policy Set to use the Compliance Violation option.
Accept (Action on Apply Policies only)	Allows the task to submit when there is a preventative identity policy violation. When you select this action, you provide a message that Identity Management writes in the audit database and displays in View Submitted Tasks when a violation occurs.
Reject (Action on Apply Policies only)	Prevents a task from submitting when an identity policy violation occurs. This action is used with preventative identity policies to prevent users from receiving privileges that may result in a conflict of interest or fraud. When you select this action, you also provide a message that Identity Management displays when a violation occurs. The message is stored in the audit database and displayed in the User Console.

Change Action	Description
Warning (Action on Apply Policies only)	Triggers a workflow process when a preventive identity policy violation occurs, if you associate that violation with a workflow approval policy. Identity Management allows the task to submit regardless of whether workflow is configured. **Note:** For information about associating a workflow process with a preventative identity policy, see Workflow and Preventative Identity Policies. (see page 212) When you select this action, you also provide a message that Identity Management displays when a violation occurs. The message is stored in the audit database and displayed in View Submitted Tasks.

More information:

Preventative Identity Policies (see page 207)
Workflow and Preventative Identity Policies (see page 212)

Specify Owners for the Identity Policy Set

On the Owners tab, you define rules about who can be an owner of the identity policy set. An identity policy set owner can modify the basic information about the policy set, and can add, change, or remove identity policies in the set.

Follow these steps:

1. Select the Owners tab.
2. Click Add and define owner rules.
3. Click Submit.

Manage an Identity Policy Set

Identity Management includes the following tasks for managing an identity policy set:

- View Identity Policy Set
- Modify Identity Policy Set
- Delete Identity Policy Set

By default, when an administrator uses one of these tasks, Identity Management displays a list of all identity policy sets for which that administrator is an owner. The administrator can then choose the policy set he needs from the list.

In a Identity Management environment that includes many identity policy sets, you may want to customize the View, Modify, and Delete Identity Policy Set tasks to allow administrators to search for an identity policy set, instead of displaying them in a list.

To customize these tasks:

1. In the User Console, select Roles and Tasks, Admin Roles, Modify Admin Task.

   The Modify Admin Task screen opens.

2. Search for and select the task that you want to customize.

3. On the Scope tab, select All Identity Policy Sets.

   When you select this option, Identity Management uses the Default Identity Policy Set Search screen definition.

4. Click Submit.

How Users and Identity Policies Are Synchronized

When using identity policies, it is important to understand how Identity Management evaluates and applies the policies to users. Without a thorough understanding of the user synchronization process, you may configure identity policy sets that yield unexpected results.

The following procedure describes how Identity Management evaluates and applies identity policies:

1. The user synchronization process begins:

   - **Automatically**—You can configure Identity Managementtasks to automatically trigger user synchronization
   - **Manually**—Use the Synchronize User task in the User Console to synchronize a user.

2. Identity Management determines the set of identity policies that apply to a user.

3. Identity Management compares the set of identity policies that apply to a user with the list of policies that have already been applied to that user.

   Note: The list of policies that have been applied to a user is stored in the %IDENTITY_POLICY% well-known attribute in the user profile. For information on configuring this attribute, see the *Configuration Guide*.

   - If an identity policy is on the list of applicable policies, *and* the policy has *not* been applied to the user previously, then Identity Management adds the policy to an allocation list.

- If an identity policy is on the list of applicable policies, the policy has been previously applied to the user, and the Apply Once setting for the policy is disabled, Identity Management adds the policy to a reallocation list.
- An identity policy is not on the list of applicable policies, and the policy has been applied to the user, the user no longer matches the policy condition. Identity Management adds these policies to a deallocation list.

4. After Identity Management evaluates all of the policies for a user, it applies policies in the following order:
   a. Identity policies from the deallocation list
   b. Identity policies from the allocation list
   c. Identity policies from the reallocation list
5. After the identity policies have been applied, Identity Management reevaluates the policies to see if any additional changes are needed based on changes that occurred in the first synchronization process (steps 2-4).

   This is to ensure that changes made by applying identity policies did not trigger other identity policies.
6. Identity Management continues to reevaluate and apply identity policies until the user is synchronized with all applicable policies, or until Identity Management reaches the maximum recursion level, which is defined in the Management Console.

   For example, an identity policy may change a user's department when the user is assigned a role. The new department triggers another identity policy. However, if the recursion level is set to 1, the subsequent change is not made until the user is synchronized again.

   For more information about setting the recursion level, see the Management Console Online Help.

Configure Automatic User Synchronization

Identity Management can automatically synchronize user accounts with identity policies at different points during a task's lifecycle.

A Identity Management task generates *events*, detectable activities that occur during task processing. For example, the default Create User task generates the CreateUserEvent, AddUserToGroupEvent, and the AssignAccessRoleEvent. You can configure Identity Management to synchronize users after a task completes, or when each event completes.

Note: The section Synchronize Users with Identity Policies (see page 198) provides more information on the user synchronization process.

To configure a task to trigger user synchronization

1. Log in to Identity Management as a user who can modify admin tasks.
2. Select Roles and Tasks, Admin Tasks, Modify Admin Task.

   Identity Management displays a search screen.
3. Search for and select the admin task that will trigger user synchronization.
4. Select one of the following options in the User Synchronization field on the Profile tab for the task:
   - **Off**—This task will not trigger user synchronization.
   - **On Task Completion**—Identity Management starts the user synchronization process after all of the events have completed. This setting is the default synchronization option for the Create User, Modify User, and Delete User tasks. The default setting for all other tasks is Off.

     Note: If you select the On Task Completion option for a task that includes multiple events, Identity Management does not synchronize users until all of the events in the task complete. If one or more of those events require workflow approval, this may take several days. To prevent Identity Management from waiting to apply identity policies until all events complete, select the On Every Event option.
   - **On Every Event—**Identity Management starts the user synchronization process when each event in a task completes.

     For tasks with a primary and secondary event for the same user, setting user synchronization to On Every Event may result in more evaluations for which policies apply to a user than if the On Task Completion option is selected.

Synchronize Users Manually

You may want to manually synchronize a user with an identity policy set to ensure that a user account has the right privileges, or complies with a compliance policy.

You can manually synchronize a user by using the Synchronize User task in the Identity Management User Console.

Note: For the Synchronize User task to work properly, the User Synchronization option must be set to Off, and the Account Synchronization option must be set to On Task Completion or On Every Event. For better performance, choose the On Task Completion option. These options are set in the Profile tab for the Synchronize User task.

The Synchronize User task includes the following tabs:

- **Currently Matched Policies**—Displays a list of identity policies that Identity Management will apply to the user when the Synchronize User task is submitted.

  Note: The Currently Matched Policies tab displays only the identity policies that apply to the user at the time you access the Synchronize User task. When the user is synchronized with those policies, changes may occur that trigger other identity policies. To prevent Identity Management from applying the new policies until you have reviewed them, set the recursion level for identity policy sets to 1 in the Identity Management Management Console. After submitting the Synchronize User task, access it again to review the policies.

- **Policies Already Applied**—Displays a list of identity policies that have already been applied to the user.

- **Synchronization Summary**—Displays all of the identity policies that apply to the user and the change actions for those policies.

To synchronize a user account

1. Log in to Identity Management as a user who can use the Synchronize User task. (By default, users with the System Manager role can use this task.)
2. Select Policies, Synchronize User.

   The Synchronize User task opens.
3. Select the Synchronization Summary tab.
4. Review the policies and associated actions that Identity Management will apply to the user, then click Submit.

Verify User Synchronization

To verify that the appropriate changes take place when a user is synchronized with identity policies, check the Policies Already Applied tab in the Synchronize User task.

1. Log in to Identity Management as a user who can use the Synchronize User task. (By default, users with the System Manager role can use this task.)
2. Select Policies, Synchronize User.

   The Synchronize User task opens.
3. Select the Policies Already Applied tab.
4. Review the policies and associated actions that Identity Management applied to the user.

Identity Policy Sets in a Identity Management Environment

The following sections describe different ways to use identity policies:

- Example: Automatically Populating User Attributes (see page 202)
- Example: Allocating Resources and Entitlements (see page 203)
- Example: Enforcing Compliance (see page 204)
- Example: Enforcing Segregation of Duties (see page 205)

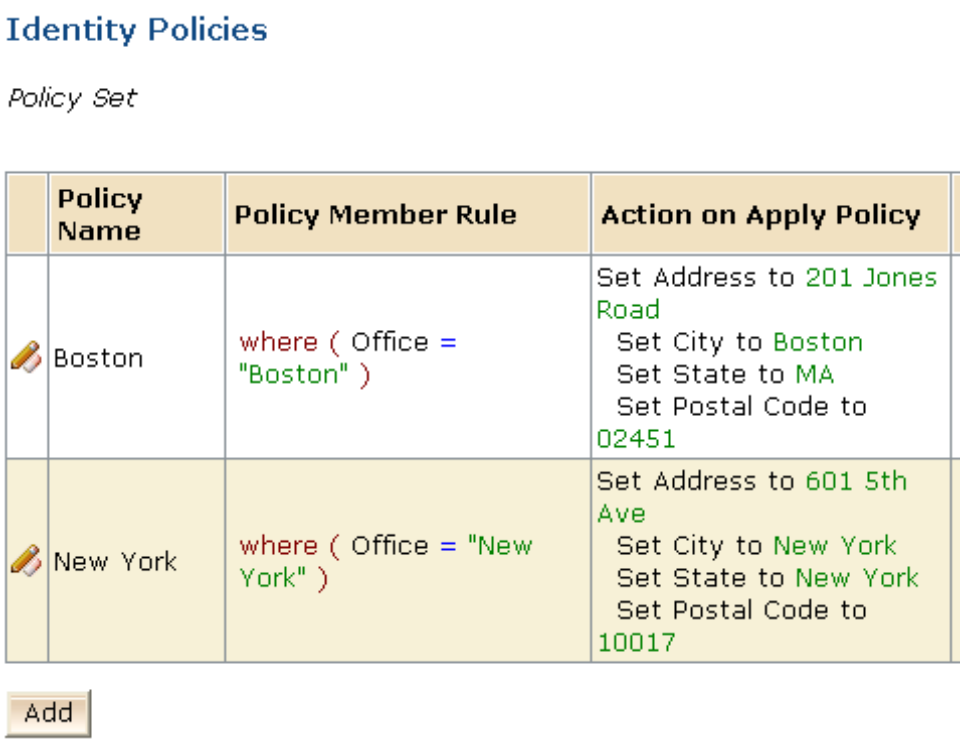
Example: Automatically Populating User Attributes

You can use an identity policy set to automatically assign user attribute values based on another attribute value or user entitlement. For example, you can create an identity policy set that automatically fills in a user's mailing address based on the user's home office.

To configure an identity policy set for employee addresses, create an identity policy with the following settings for each office location:

Setting	Value
Policy Condition	office = <office_location>
Action on Apply Policy	set Street Address = <some street_address> set City = <some city> Set State/Province = <some state or province> Set Postal Code = <some postal code>

The following figure shows sample policies in the Employee Addresses identity policy set.



Example: Allocating Resources and Entitlements

Identity policies can automatically assign resources, such as domain accounts, or grant entitlements, such as making a user a member of a role, when users meet the policy condition. For example, you can create a set of identity policies that assign resources and roles based on a user's title.

To create an identity policy set for allocating resources and roles, create an identity policy with the following settings for each of the titles in your organization:

Setting	Value
Policy Condition	title = <some_title>

Setting	Value
Action on Apply Policy	Any actions that allocate resources or entitlements to users who meet the policy condition, for example: ■ make member of <some_group> ■ make member of admin role <some_admin_role> ■ make member of provisioning role <some_provisioning role>
Action on Remove Policy	Any actions that remove resources or entitlements when a user no longer meets the policy condition. For example, if Identity Management made the user a member of a role when the identity policy was applied, you may want to configure Identity Management to revoke the role when the user no longer meets the policy condition.

The following figure illustrates sample policies in the Employee Resources identity policy set:

Identity Policies

Policy Set

	Policy Name	Policy Member Rule	Action on Apply Policy	Action on Remove Policy
	Managers	where (Title = "Manager")	Make member of admin role User Manager Make member of provisioning role Corporate NT Domain Role	Remove member from admin role User Manager
	Human Resources	where (Title = "HR Administrator")	Make member of admin role User Manager Add to group HR Department Make member of provisioning role Corporate NT Domain Role	Remove from group HR Department Remove member from admin role User Manager Remove member from provisioning role Corporate NT Domain Role

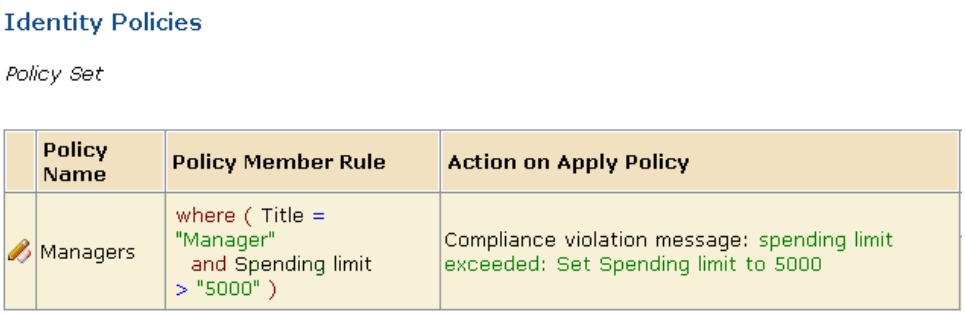
Example: Enforcing Compliance

You can configure identity policies to define conditions that must or must not exist, and to take certain actions based on the evaluation of those conditions. For example, you can define a compliance policy that states that managers must have a spending limit of $5,000. If a manager has a spending limit of $10,000, Identity Management can reset the manager's spending limit, and record a compliance violation for auditing purposes.

To create a compliance policy set for enforcing spending limits, create an identity policy with the following settings:

Setting	Value
Apply Once	Not enabled
Compliance	Enabled
Policy Condition	Any conditions that define compliance or a compliance violation--for example: title=<some_title> AND Spending Limit > <some spending limit>
Action on Apply Policy	The actions that Identity Management should take when the policy condition applies--for example: ■ Compliance violation message: Spending limit exceeded ■ Set spending limit to <some_value>

The following figure shows the sample compliance policy described in this example.



Example: Enforcing Segregation of Duties

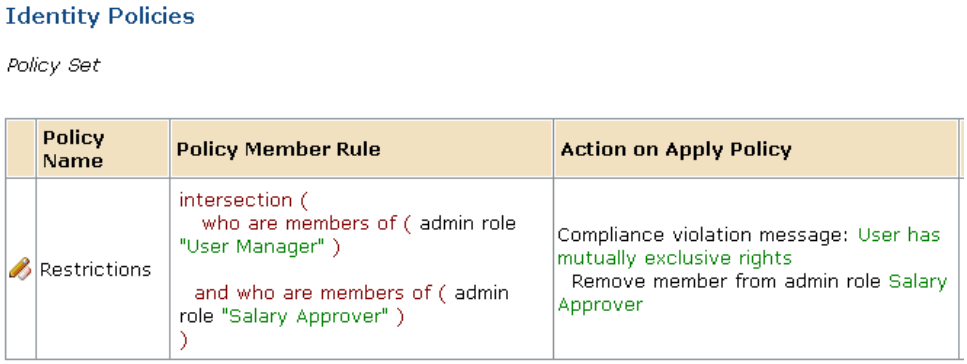
Identity policies can define roles that are mutually exclusive and cannot be granted to the same user concurrently. For example, you can prevent a user manager who can grant raises from also being a salary approver.

To create an identity policy set that enforces segregation of duties, create an identity policy with the following settings:

Setting	Value
Apply Once	Not enabled

Setting	Value
Compliance	Enabled
Policy Condition	Use the "in <administrative-intersection-constraint>" option to define a set of conditions that violate a business policy. If a user meets all of the conditions, Identity Management takes the actions in the Action on Apply Policy field. For example, set the policy condition as follows: intersection (who are members of <some_role>) and who are members of <some_other_role>)
Action on Apply Policy	The actions that Identity Management should take when the policy condition applies--for example: ■ Compliance violation message: User has mutually exclusive roles ■ Remove member from <some_role>

The following figure illustrates the identity policy in this example.



Preventative Identity Policies

A *preventative identity policy* is a type of identity policy that prevents users from receiving privileges that may result in a conflict of interest or fraud. These policies support a company's Segregation of Duties (SOD) requirements.

Preventative identity policies, which execute before a task is submitted, allow an administrator to check for policy violations before assigning privileges or changing profile attributes. If a violation exists, the administrator can clear the violation before submitting the task.

For example, a company can create a preventative identity policy that prohibits users who have the User Manager role from also having the User Approver role. If an administrator uses the Modify User task to give a User Manager the User Approver role, Identity Management displays a message about the violation. The administrator can change the role assignments to clear the violation before submitting the task.

You can create preventative identity policies for the following changes:

- **Role membership**

  Prevents users from having certain roles at the same time.

  For example, users cannot have the User Manager and User Approver roles at the same time.

- **Role administrators**

  Prevents users from being administrators of certain roles if they are administrators of other roles.

  For example, users cannot be administrators for the User Manager and User Approver roles at the same time.

- **User attributes**

  Prevents users from having certain profile attributes at the same time.

  For example, users cannot have the title Senior Account and belong to the IT department.

- **Organization attributes**

  Prevents user profiles from being created in a certain organization.

  For example, administrators cannot create employee profiles in the Suppliers organization.

- **Group attributes**

  Prevents users from being members in certain groups.

  For example, users cannot be members of the Project Team group and the Accounting Group.

More information:

Actions for Preventative Identity Policy Violations (see page 208)

Actions for Preventative Identity Policy Violations

When a preventative identity policy applies to a business change, CA performs certain actions to address the violation.

When you specify one of these actions in an identity policy, you specify a message that describes the violation. This message is recorded in the audit database. Depending on the type of action, the message may also be displayed to users in the User Console and recorded in View Submitted Tasks.

You can configure the following actions for a preventative identity policy:

Accept

Identity Management displays a message in View Submitted Tasks that describes the violation, but allows the task to be submitted.

Reject

Identity Management displays a message in the User Console and prohibits the task from submitting.

Warning

Identity Management displays a message in the User Console and in View Submitted Tasks. This action can optionally trigger a workflow process that requires an approval from an appropriate user before Identity Management executes the task.

To trigger a workflow process, you associate the preventative identity policy with a policy-based workflow process (see page 214) in tasks that may cause the violation.

For example, if the violation occurs when a user receives certain roles at the same time, configure the workflow process for all tasks that assign those roles to users.

Note: When you configure the policy-based workflow process for the task, the approval rule must reference the name of the preventative identity policy.

How Preventative Identity Policies Work

The following sample process illustrates how preventative identity policies work:

1. An identity policy administrator creates a preventative identity policy that prohibits users who have the title Senior Accountant from being in the IT department.

   When defining this identity policy, the administrator specifies that Identity Management should reject any changes that violate this policy.

2. An HR administrator uses the Create User task to create a user profile for a new Senior Accountant. The HR administrator correctly selects the user's title, but accidentally selects the IT department.

3. The HR administrator completes the remaining fields in the Create User task and clicks Submit.

4. Identity Management detects that the task involves changes that are defined in an identity policy and evaluates the changes for violations.

5. Identity Management detects the violation, displays a message to the HR administrator, and prevents the task from submitting.

   Identity Management also records the message in the audit database.

6. The HR administrator views the details of the violation in the message and changes the user's department to Finance. Then, the administrator resubmits the task.

7. Identity Management evaluates the proposed changes against all applicable identity policies, and then allows the Create User task to submit.

Important Notes about Preventative Identity Policies

Before you implement preventative identity policies, note the following:

- Preventative identity policies only prevent violations that would occur because of proposed changes in the current task. They do not prevent existing violations.

  For example, a company creates a preventative identity policy that prohibits users from having the User Manager and User Approver roles at the same time. An administrator assigns the Group Manager role to a user who already has the User Manager and User Approver roles. Identity Management allows the new assignment to succeed because that change does not directly cause a violation of the policy.

- If multiple preventative identity policies apply to a set of proposed changes, Identity Management applies policies with Reject actions first.

- Do not specify dynamic groups in preventative identity policy conditions. (Policy conditions determine the set of users that the preventative identity policy applies to.)

  For example, a company has a dynamic group that includes all users who have the title Manager. That company also creates a preventative identity policy that prohibits members of the Managers group from having the Contractors role.

  An administrator changes the title of a user who has the Contractors role to Manager. This change will make the user a member of the Managers group *after* the task submits successfully. However, the user's title is not Manager at the time that Identity Management evaluates the policy, so no violation is detected.
- The role owner filter and the LDAP query filter are not supported in policy conditions for preventative identity policies.

Create a Preventative Identity Policy

Before you create a preventative identity policy, you create an identity policy set, which logically groups a set of identity policies.

Note: See Important Notes about Preventative Identity Policies (see page 209) before you begin.

To create a preventative identity policy set

1. Open Policies, Create Identity Policy Set in the User Console.

   Create a new identity policy set or use an existing identity policy set as a template.
2. Define the profile for the identity policy set (see page 187) on the Profile tab.
3. Create a policy set member rule (see page 188) on the Policies tab.
4. Create a preventative identity policy as follows:
   a. Click Add.
   b. Enter a name for the identity policy.

      Note: The Apply Once and Compliance settings do not apply to preventative identity policies.

   c. Identify the users to which the policy applies in the Policy Condition section.

      Note: The role owner filter and the LDAP query filter are not supported for preventative identity policies.

d. In the Action on Apply Policy field, define the actions that Identity Management takes when Identity Management detects a policy violation:

Accept

Identity Management displays a message in View Submitted Tasks that describes the violation, but allows the task to be submitted.

Reject

Identity Management displays a message in the User Console and prohibits the task from submitting.

Warning

Identity Management displays a message in the User Console and in View Submitted Tasks. This action can optionally trigger a workflow process (see page 212).

When you select one of these actions, Identity Management displays a text box where you can specify the message that appears when a violation occurs.

e. Specify the message in the text box.

Note: If you are localizing the User Console, you can specify a resource key instead of text in the message field. See the *User Console Design Guide* for more information about resource keys.

f. Add additional actions if necessary and click OK.

5. Specify owners for the Identity Policy set (see page 197).

Note: Before you use the identity policy set that you created, make sure that identity policies are enabled in the Management Console. See the *Configuration Guide* for more information.

Use Case: Preventing Users from Having Conflicting Roles

Forward, Inc. wants to prevent its employees from having the User Manager role and the User Approver role at the same time. Employees who have both of these roles can modify user attributes, such as salary, and approve them inappropriately.

To prevent this situation, Forward, Inc. creates a preventative identity policy that applies to users who have the User Manager and User Approver Roles. If an administrator attempts to give these roles to a user, Identity Management rejects the task submission and displays a message that explains the violation.

You configure a preventative identity policy to support this use case as follows:

- Create an identity policy set for the policy that you want to create.
- Create a preventative identity policy with the following settings:
  - Policy Condition:

  - Action on Apply Policy:
    - Reject with message: User cannot be a member of User Approver and User Manager roles

Workflow and Preventative Identity Policies

When a preventative identity policy is configured to issue a warning, you can define a task level policy-based workflow process, which is associated with the identity policy, for tasks that may trigger a violation. For example, if an identity policy prohibits Senior Accountants from being members of the IT department, you define a task level policy-based workflow process on the Create User and Modify User tasks.

All work items that are generated as a result of task level policy-based workflow must be approved before Identity Management executes the task. Approvers see a work list item when they log into the User Console. When the approver clicks the work list item, an approval task, which includes the warning message that describes the violation, appears. The approver can choose to approve or reject the task, based on the violation.

Policy-based workflow processes are associated with preventative identity policies by the policy name.

More information:

Policy-Based Workflow (see page 328)

Identity Policy Violations in Approval Tasks

When a preventative identity policy is associated with a workflow process for a task, Identity Management generates a work list item for the appropriate approvers. These approvers use an Approval task to approve or reject the change that triggered the policy violation.

The default Approval task includes a section that lists identity policy violations. There may be more than one violation if the proposed changes trigger multiple preventative identity policies.

Each violation can have of the following status:

- **Pending Evaluation**

  Identity Management has not started evaluating the approval rules for the task yet. This is the initial state.

- **Awaiting Approval**

  Identity Management located a match for the identity policy defined in the approval rules and triggered the associated workflow process.

- **Approved**

  An approver approved the proposed changes. Identity Management makes the changes that triggered the preventative identity policy violations.

- **Rejected**

  An approver rejected the proposed change. The task is rejected.

- **No Workflow Configured**

  There is no workflow process configured for this violation. The task executes without any approval required.

How to Configure Workflow for Preventative Identity Policies

You configure workflow for preventative identity policies in the admin tasks that include changes that may trigger an identity policy violation.

For example, if the preventative identity policy prohibits users from having certain admin roles at the same time, configure tasks that assign admin roles to support workflow for preventative identity policies.

Note: Before you configure workflow, create a preventative identity policy with the following settings:

- A unique policy name

  The policy name must be unique across all identity policy sets because workflow processes are associated with preventative identity policies by the policy name.

  If multiple preventative identity policies have the same name, multiple workflow processes may apply.

- Warning in the Action on Apply Policy field

  Warning is the only action that can trigger a workflow process.

After you configure the preventative identity policy, determine the tasks that may trigger the policy violation. Then, create a workflow approval policy (see page 214) for those tasks.

Create a Workflow Approval Policy for Preventative Identity Policies

You can configure a task level policy-based workflow process for an admin task. This workflow process includes one or more approval policies that can associate a preventative identity policy with a workflow. Identity Management executes the workflow when a violation of the associated preventative identity policy occurs.

Note: For more information about task level policy-based workflow processes, see Policy-Based Workflow (see page 328).

To create a workflow approval policy for preventative identity policies

1. Modify the admin tasks that allow changes that might trigger a violation of a preventative identity policy.

   For example, if an identity policy violation occurs because a user has the User Manager and User Approver roles, modify the admin tasks that allow administrators to assign roles, such as Create User, Modify User, and Modify Admin Role Members/Administrators.

2. Click the edit icon next to the Workflow Process field on the Profile tab for the task to add a workflow process.

   Identity Management displays the Task Level Workflow Configuration screen.

3. Select Policy Based, then click Add.
4. In the Approval Rule section, select the Identity Policy Violation object.
5. In the Identity Policy field select a filter that determines which identity policies trigger the workflow associated with the approval policy.

   In the filter, include the identity policy name, *not* the identity policy set name.
6. Configure the Rule Evaluation, Policy Order, and Policy Description fields as needed.
7. Select a workflow process, then click OK.

   When you select a workflow process, Identity Management displays additional fields.
8. Specify approval tasks and approvers as needed.

   Identity Management associates the workflow process with the preventative identity policy.

Use Case: Approving Titles

Forward, Inc has a company policy that states that all managers must be full-time employees. However, Forward, Inc has recently hired many contractors for special projects. To run these special projects efficiently, some of the contractors will be given the Manager title. Forward, Inc wants to require approvals from the Human Resources Director before allowing administrators to assign the Manager title to a contractor.

To automate the approval process in these situations, Forward, Inc creates a preventative identity policy, named Manager Titles for Contractors, that detects when a user title is Manager and user organization is Contractor. Forward, Inc also configures a policy-based approval process on the Modify User task. This approval process is triggered when the Manager Titles for Contractors policy is violated.

When an administrator changes a contractor's title to Manager, Identity Management displays a warning message, and sends a work item to the Human Resources Director for approval. Identity Management does not change the contractor's title until the work item is approved.

To configure support for this use case, you complete the following in Identity Management:

- Create a preventative identity policy called Manager Titles for Contractors with the following settings:
  - Policy Condition: Users where (Title = "Manager" and Organization = "Contractor")
  - Action on Apply Policy: Warning with message "Managers must be full-time employees"
- Modify the Modify User task to include a workflow process with the following settings:
  - Workflow Process: Policy Based
  - Approval Rule Object: Identity Policy Violation
  - Identity Policy: where (Name = "Manager Title for Contractors")
  - Workflow Process: SingleStepApproval

Combining Identity Policies and Preventative Identity Policies

You can combine identity policies and preventative identity policies to address Segregation of Duties (SOD) requirements. In this case, identity policies address existing SOD violations and preventative identity policies prohibit new violations.

To support this use case, you configure an identity policy set with two types of actions:

- Actions that occur during user synchronization

  These actions result in changes to user attributes, group and role members, administrators, or owners. For example, an action of this type may remove a user from a role when a violation is detected.

  These actions differ from preventative actions in that they are not applied when a task is submitted. They are applied only during user synchronization (see page 198).

- Preventative actions

  These actions determine what Identity Management does when a preventative identity policy violation occurs *before* a task is submitted. Identity Management can allow the task to submit, issue a warning and trigger a workflow process, or prevent the task from submitting.

  In each of these cases, the violation is recorded in the audit database.

Consider a company that wants to prevent users from having the HR Administrator and Salary Approver roles at the same time. That company creates an identity policy with two Action on Apply Policy actions:

- Remove the user from role Salary Approver

  This action occurs when Identity Management synchronizes users with identity policies.

  In this case, this company configured user synchronization for the Modify User task. When an administrator modifies a user, Identity Management evaluates all applicable identity policies and applies the actions. In this example, Identity Management removes users who have the HR Administrator role and the Salary Approver Role from the Salary Approver role.

- Reject the task

  This preventative action prohibits administrators from assigning these two roles to a person by not allowing the administrator to submit the task.

Note: When you configure an identity policy with both of these types of actions, verify that the actions do not conflict. For example, you can configure an identity policy that prevents users from having the Manager and Contractor roles. In the policy, you specify two actions:

- A warning that triggers a workflow process, which requires an approval before assigning the roles, and
- An action that removes a user from the Manager role

An approver approves the role assignment for the Manager and Contractor roles, but the second action removes the user from the Manager role when user synchronization occurs.

Chapter 10: Policy Xpress

This section contains the following topics:

Policy Xpress Overview

Policy Xpress allows you to create complex business logic (policies) in Identity Management without the need to develop custom code. However, the concepts involved in creating Policy Xpress policies are complex and require careful thought and planning. An administrator using Identity Management portal screens can configure a policy within Policy Xpress to implement even the most sophisticated business logic required. As business policies change, an administrator can modify the policies using configuration screens within Identity Management without requiring a developer to make underlying code changes, or more importantly—with proper change management procedures—without restarting the Identity Management services.

Note: For more detailed information on Policy Xpress, see the Policy Xpress Wiki.

How to Create a Policy

To create a policy with Policy Xpress, define the following basic elements of a policy.

Profile

Defines the policy type and priority, and allows for grouping similar policies for easy management.

Events

Define when a policy runs.

Note: Be sure to set the Events parameter carefully. Business logic must run at specific times to prevent data corruption and to increase performance. For example, setting a user as enabled should occur when the user is created. Running this logic at all times may cause user accounts that should be disabled to become enabled again. Another example is giving the user a provisioning role that grants access to a certain system. This role should only be assigned to the user after a different role has been assigned and approved. Policy Xpress allows for the activation of its business logic during event and Business Logic Task Handler processing, much like custom adapters. Therefore, unlike identity policies, the logic can be triggered at any time, and not only at the beginning of a task.

Data (Data Elements)

Specify the data used by the policy. Every type of business logic requires some data to work with. That data can be used to make decisions or it can be used to construct more complex data. Policy Xpress provides many individual components to gather data. These components are referred to as *Data Elements*. An example of a data element is a user's attribute value. For example, Policy Xpress can gather the user's first name and store it as a data element for later use.

Entry Rules

Define the requirements that must be met before execution. Defining entry rules allows you to specify when Policy Xpress evaluates policies, which can simplify policies and improve performance. An example of an entry rule is to run a 'Set Full Name' policy *only* if the first name or the last name has changed.

Action Rules

Define the action taken based on the information gathered. For example, based on a user's department name, Policy Xpress can assign a user to different roles or specify different account values.

Actions

Specify the action to perform. At the end of the process, Policy Xpress performs the actions needed by the business logic. Policy Xpress works by having an action rule attached to multiple actions, so when the rule is met, the actions are performed. Actions can include assigning attribute values to a user or an account, executing a command line, running a SQL command, or generating a new event.

Profile

The profile tab for a Policy Xpress policy contains fields that manage policies and refine policy capabilities.

Note: A policy only applies to the environment it is created in. For example, if you create a policy while logged into the neteauto environment, the policy runs only for the neteauto environment.

Provide the following profile information when creating a policy:

Policy Name

Defines a unique friendly name for the policy.

Policy Type

Defines the listeners (see page 222) that trigger the policy. Each policy type has a different configuration.

Note: You cannot change this field once the policy is saved.

Category

Defines a group of related policies. This field allows you to group policies for easy management.

Description

Specifies a description of the policy.

Priority

If there are multiple policies that run at a single event, this field specifies when the policy runs. Policies are executed based on their priority. The lower the number, the higher the priority (priority 1 runs first, 10 runs second, 50 runs third, and so on). Setting priority is useful for policies which have a dependency on one another, or breaking a complex policy into two simple ones, that run one after the other.
For example, there are three policies which run if there is a specific value in the database. Instead of having each of the policies verify the value in the database, you can create a policy that runs before the other three policies and checks the value. If the new policy matches the required value, Policy Xpress can set a variable. The other three policies only run if that variable is set, which prevents redundant access to the database.

Enabled

Specifies if the policy is active in Identity Management. You can clear this check box if you want to disable a policy without deleting it.

Run Once

Specifies if the policy runs only once. Some policies may need to run every time they meet criteria, and others may need to run only once. This value determines if action rules that have already executed in the past should execute again.

For example, adding an SAP role to a user based on department is an action that should only occur the first time the user matches that department. Alternately, a policy that sets the user's salary level based on title would *not* be set to run once, to make sure that no unauthorized changes take place.

Note: The Run Once option applies to an object, it does not apply globally.

Listeners

Policy Xpress policies are triggered by something that happens in the system. To implement this functionality, listeners that integrate with the system notify Policy Xpress when an event happens, and provide details about the event that occurred.

The following listeners are available:

Event

Listens for every event in the system and all the states associated with the event (before, approved, rejected, and so on). This listener also reports the name of the event to Policy Xpress. The following states are available for the Event listener:

- Before
- Rejected
- Approved
- After
- Failed

UI

Listens for different tasks running in the system during the synchronized state, meaning while a user still has the user interface for the task open. The following states are available for the UI listener:

- Start—when the task starts
- Set subject—when the primary object is found
- Validate On Change (see page 223)—when an attribute set with the Validate on Change flag changes
- Validate On Submit—when clicking the submit button
- Submission—when the task is submitted

Workflow

Listens for workflow processes that have found approvers. This listener is useful for performing logic based on approvers, such as sending an email to the approver.

Submitted task

Listens for submitted tasks not running in the background. This listener is similar to the Event listener, however, it considers the task as a whole, instead of the task's events. The following states are available for the Submitted task listener:

- Task started
- Task completed
- Task failed

Reverse Sync

Listens for notifications in the system that relate to Identity Management's Explore functionality.

On-Screen Attribute Validation

In addition to the defined triggers (policy types), Policy Xpress can also listen to validation on attributes. This allows you to create policies that can run when an on-screen attribute that has been flagged as “validate on change” is updated.

This functionality can be used for creating dependant drop-down lists. For example, if there are two drop-down lists on the screen, Policy Xpress runs when the first drop-down option is selected, then sets the values for the second drop-down list based on the option selected in the first. An unlimited number of drop-down lists and other screen refreshes can be done. This differs from Select Box Data by allowing the drop-down options to be populated using any logic, rather than importing an XML file of static options.

Another use is populating other attributes based on the value of one attribute. For example, when an administrator selects a department, Policy Xpress can automatically populate other attributes, such as a department manager, a department number, and a HR Dept Code. This replaces the need to write logical attribute handler custom code.

To configure validation with a Policy Xpress policy

1. In the User Console, modify a task's profile screen and select the field you want to listen for.
2. Access the field's properties and select Yes in the drop-down list for Validate on Change.
3. In Policy Xpress, create a policy of type 'UI (see page 222)' .
4. Under the Run at Events tab, select the State 'Validate on Change' and the task you modified in Step 1.

Use Case: Checking for Offensive Names

When a new user is created, you may want to check if the username is offensive. The following process outlines how to check for offensive names using a Policy Xpress policy.

1. Be sure that the appropriate fields in the Create User task's profile screen are set to Validate on Change = Yes.
2. In Policy Xpress, create a policy of type 'UI'.
3. Under the Run at Events tab, select the State 'Validate on Change' and the Create User task.
4. Create the following data elements to check the first name:
   - Get the first name attribute (Attributes, User attribute, Get)
   - Parse the first name to all lower case letters (General, String parser, To lower)
   - Check the first name against offensive words in a database table (Data sources, SQL query data).
5. Create similar data elements as in Step 4 to check the last name.
6. Create an action rule as follows:
   - Condition—first name is not equal to "" (this occurs if the query returns a message that the name is offensive)
   - Action—message that displays (Messages, On screen message) indicating the offensive name.

   This rule will force the user to change the name before submitting the Create User task again.
7. Create a similar action rule as in Step 6 for the last name.

Events

Depending on the policy type selected on the profile tab, you can configure activation times to establish when the policy is evaluated. For example, a policy of type Event can be set for evaluation Before a CreateUserEvent. A policy of type Task can be set for evaluation at Set Subject for DisableUserEvent.

To configure an activation time, select the following fields:

State

Specifies the time frame or action related to the event that activates the policy. For example, a policy can be set to run "Before" an event occurs.

Event Name

Specifies the event that activates the policy, such as a CreateUserEvent.

A policy can have more than one activation time. Every time a specified activation time (a state and an event) occurs in the system, Policy Xpress searches for all policies with that activation time, and evaluates each policy based on its order.

Note: If a policy matches an activation time that occurs in the system, it does not mean that the policy is automatically run. Rules criteria evaluated later in the process determine if the policy is completed.

Data Elements

Data elements are used for creating policy data. A policy can contain multiple data elements that represent the information used by the policy.

Policy Xpress uses flexible plug-ins for gathering the data element information. Each plug-in can perform a small, dedicated task. However, several plug-ins can be used together to build more complex policies. An example of a data element plug-in is a user attribute element. The goal of the element is to gather information about a certain attribute which is a part of the user's profile.

Data elements are calculated when they are called, meaning either a rule is using the data element, or another element needing calculation is using the data element as a parameter.
For example, an SQL query data element can retrieve a value from a table, but it needs the user's department to build the query. In this case, the department data element must run before the SQL query data element, and then the value can be used as a parameter (see page 227).

The following fields define a data element:

Name

Defines a friendly name that describes the data element. Some data elements are complex (such as getting variables or retrieving information from the database). Be sure to select a meaningful name to simplify data element management.

Category

Provides a grouping of data elements. This field sorts the data elements and makes selection easier.

Type

Specifies the data element type, each with its own dedicated use. This field is based on the category selected.

Function

Defines possible variations of the same data. Most data elements only support the Get function.

For example, the user attribute data element has the following functions:

- Get—returns the values of the attribute
- is multi valued—returns true if the value is multi-valued
- is logical—returns true if the value is logical

Function Description

Provides a prepopulated description of the function. Each function selected provides a different description to help in understanding its use and what the expected values are.

Parameters

Defines the parameters passed to the data element. Data elements are dynamic and can do different things based on the parameters. A user attribute data element returns different results based on the attribute selected. The sub type option also defines the number of parameters, their names, and the optional values when available.

You can add additional parameters if necessary. The SQL query example accepts two required parameters, the data source and the query itself. The query can use the "?" to be replaced with values (much like a prepared statement). Adding additional parameters allows you to set those values.

Note: When viewing data elements in Policy Xpress, there is a column titled 'In Use'. A checkmark in this column means that the data element is used by a rule, an action parameter, or as a parameter to other data elements.

Use Dynamic Values in Data or Action Elements

Dynamic values are the result of calculated data elements, and their values are only decided at run time. These values can then be used as parameters of other data elements (that are subsequently calculated, based on priority).

To use a dynamic value as a parameter for a data element

1. In the Policy Data tab, locate the parameter you want to set a dynamic value in.
2. In the empty text field, enter any regular text or select the dynamic value from the right drop-down list.
3. Click OK.

Variables

Policy Xpress has variables that are set with actions and saved as data elements (Variables category). Variables are shared across all policies that run at the same time, so a variable that has been set can be used by other policies of lower priority.

For example, a variable can contain a value calculated once by a policy, and then shared across other policies that no longer need to recalculate the value. The initial policy sets a value to the variable, and policies that run later read that value by using a data element that has the variable name as a parameter.

A variable can also be a trigger for other policies. In this case, the policies only run if the policy before them has run.

Entry Rules

Entry Rules define the conditions for when a policy should run. These conditions use the values gathered by the data elements in the policy.

There can be multiple entry rules in a policy, and an entry rule can have multiple conditions. At least one entry rule must be matched, meaning *all* conditions in that entry rule must be met for a policy to move on to the action rules.

The following fields define an entry rule:

Name

Provides a friendly name for the entry rule.

Description

Defines the meaning of the entry rule.

Conditions

Specifies the criteria to match.

Note: Conditions in an entry rule always have an AND operator between them.

More Information:

Conditions (see page 228)

Conditions

A condition is used in entry and action rules and is comprised of the following components:

- Policy Data
- Operator
- Value

For example, you want to create a condition that checks if a user's department was changed. First, define a Department Changed data element, then, in the condition, select the Department Changed data element, set the operator to Equals, and set the value to True.

More Information:

Entry Rules (see page 227)
Action Rules (see page 228)

Action Rules

Action rules are similar to entry rules in structure, but differ in functionality. Action rules define when action should be taken. For example, if you want a policy to perform an action when a user's department has changed to Sales, create an action rule that defines when 'Department = Sales'.

Also, instead of having to match one entry rule, several action rules may be matched. The single action rule with the highest priority (0 being the highest) is the *only* one used.

Action rules also contain one or more actions, and the actions are divided into Add Actions and Remove Actions.

The following fields define an action rule:

Name

Provides a friendly name for the action rule. This name must be unique.

Description

Defines the meaning of the action rule.

Conditions

Specifies the criteria to match.

Priority

Defines which action rule executes, in the case of several action rules matching. This field is useful for defining default actions. For example, if you have multiple rules, each for a department name, it is possible to set a default by adding an additional rule with no conditions but a lower priority (such as 10 if all others are 5). If none of the department rules are matched, then the default is used.

Add Actions

Defines a list of actions taken when the rule is matched. For example, you can configure a rule that states if the user's department matches the one configured in the condition, add a specific Active Directory group. Action rules behave differently, based on the Run Once setting. If the policy is set to run once, the associated actions are performed the first time the rule matches. The actions are not performed again for each subsequent rule match. In the example above, the Active Directory group is added to the user only once. If Run Once is not set, then the actions run again as long as the rule is matched. This field is important for enforcing values.

Remove Actions

Defines a list of actions to perform when the rule no longer matches. For example, the previous example added an Active Directory group to the user, based on the department. If the department changes, then the remove action removes the Active Directory group.

More Information:

Conditions (see page 228)

Actions

Actions perform the business logic after all the decision-making is done. An action works in a similar way to data elements except at the end. When it runs, it performs a task instead of returning a value.

Note: Actions are run in the order they appear in the User Console.

The following fields define an action:

Action Name

Defines the purpose of the action.

Category

Provides a grouping of actions. This field sorts the actions and makes selection easier.

Type and Function

Defines the type and function of the action taken.

Note: For more information about Type and Function, see Data.

Function Description

Provides a prepopulated description of the function. Each function selected provides a different description to help in understanding its use and what the expected values are.

Parameters

Defines the parameters passed to the action.

Flow Control

By default, policies are sorted by priority and then evaluated one by one. While this flow almost always applies, you can change the flow, if necessary.

This flow-changing functionality is represented by an action that can be attached to any action rule. Flow-changing functions are located under the System category of the action.

Important! Use caution when changing process flows. Using these actions may result in an infinite loop. For example, if you set 'Redo the current policy' on an action rule with no conditions, the rule will always be true, and the policy will always restart and never exit.

The following four flow-changing functions can be used:

Stop processing

Causes all policies after the current policy to be ignored, and causes Policy Xpress to exit.
Note: Only Policy Xpress exits. If you want to force Identity Management to stop also, you can use the Exception type action plug-in.

Restart all policies

Stops processing the rest of the policies and goes back to the start of the list. This option is useful in cases where the action of one policy causes another policy, which ran before it and did not execute, to now meet the entry criteria. That policy is now reevaluated.

Redo the current policy

Causes a policy to run again. This option is useful for iteration. For example, creating a unique username requires a policy to run over and over again until it finds a unique name.

Go to a specific policy

This action requires selecting an existing policy. If that policy is running at the same time as the current policy (can be before or after) then Policy Xpress jumps to the selected policy. If the new policy is of lower priority, all policies between the current policy and the selected policy are ignored. If the new policy priority is higher, the process goes back.

Note: Because the action may cause Policy Xpress to skip certain policies, use this action type with caution.

Set Objects Associated with Accounts

When creating an Add Action to set an object that is associated with an account, such as Member Of, a specific relationship format is used to represent the object. The following two types of formats can represent the object in Identity Management:

- To represent simple relationships between the object and the account, for example, Active Directory Groups:
  `NativeGroup=Administrators,Container=Builtin,EndPoint=LocalAD,Namespace=ActiveDirectory,Domain=im,Server=Server`
- To represent binding relationships between the object and the account, for example, SAP Roles:
  `{"validFromDate":"2009\/12\/01","roleName":"SAPRole=SAP_AUDITOR_ADMIN,EndPoint=sap endpoint,Namespace=SAP R3,Domain=im,Server=Server","validToDate":"2009\/12\/31"}`

  A binding relationship differs from a simple relationship in that the association between the object and the account has additional data on it. In the previous example, the parameters validFromDate and validToDate only contain data related to the association between the account and the SAP role. The validFromDate and validToDate data does not exist on the account, or the role object.

To discern the format for the relationship, create a data element that Gets the value of the object. The value returned is the format you use in the Add Action to set that object.

Example: Active Directory Groups

1. Create a Policy Xpress policy with the following settings:
   - Policy Type: Event
   - Events: After – Modify User
2. In the Action Rule, configure the following Add Action:
   - Category: Attributes
   - Type: Set Account Data
   - Function: Set
   - Endpoint Type: Active Directory
   - Endpoint: *endpoint_name*
   - Account Name: *account*
   - Attribute: Member Of (groupMembership)
   - Value: NativeGroup=Administrators,Container=Builtin,Endpoint=*endpoint_name*,Namespace=ActiveDirectory,Domain=im,Server=Server

Advanced

Policy Xpress allows for many configuration variations and also interacts with external components. Due to this flexibility, errors may occur that are not necessarily bugs, such as an incorrectly configured data source, a missing value returned from a dynamic data element, or an endpoint which is not responding.

Usually when an error occurs, the system will stop the calculation of policies for the current step. However, you can change the default error response, based on the error category. For example, if you have a policy that is non-critical, you can define that the processing should continue in the event of an error.

The Advanced tab allows you to change the default error responses if necessary.

Note: We recommend that these error responses be left to their defaults, but for advanced use cases, these settings can be changed per policy. For example, if you have a policy that is non-critical, you can define that processing should continue even if the policy failed.

The following error categories can be configured on the tab:

- Validation—caused by providing incorrect information to a plug-in. This type of error is reported before the action is attempted.
- Environment—caused by problems in the environment, such as a failing database server for the SQL plug-in.
- Allowed—a non-critical error. The default behavior for this error type is to continue processing the request, such as when sending an email fails.

For each of the previous errors, the following options can be set:

- Fail event—stops the current action. This is the default for most error types.
- Fail policy—stops the current policy and all actions associated with it. The rest of the policies continue.
- Ignore—logs any failure but does not stop the actions or policies.

Chapter 11: Reporting

This section contains the following topics:

Configuration Overview

Within Identity Management, you can run two different types of reports:

Snapshot Reports

Include data from the Snapshot Database, which contains information from the Identity Management object store and the Identity Management user store. An example of a Snapshot Report is the User Profile report. You define the data that is added to the Snapshot Database using Snapshot Definitions that specify the information to include.

Non-Snapshot Reports

Include data from other data sources, such as the Audit Database. For example, Identity Management includes default audit reports. (These reports have the prefix "Audit - " in their name in the User Console). By default, Identity Management only includes Audit reports, but you can create your own custom reports that include data from any data source, such as the Workflow or Task Persistence databases.

Each report within Identity Management requires initial configuration before you can run it. The configuration steps depend on the type of report that you want to run.

The following steps summarize the procedures that this chapter contains.

For Snapshot Reports

1. Create a snapshot definition file to define the data that is added to the snapshot database.
2. Capture snapshot data for the report.
3. Modify the Report Task in the User Console and perform the following actions:
   a. Associate a snapshot definition with the task.
   b. Add the rptParamConn connection object to the task.

4. Request the report using one of the following methods:
   - Run the report immediately
   - Schedule the report
5. View the report in the User Console.

For Non-Snapshot Reports:

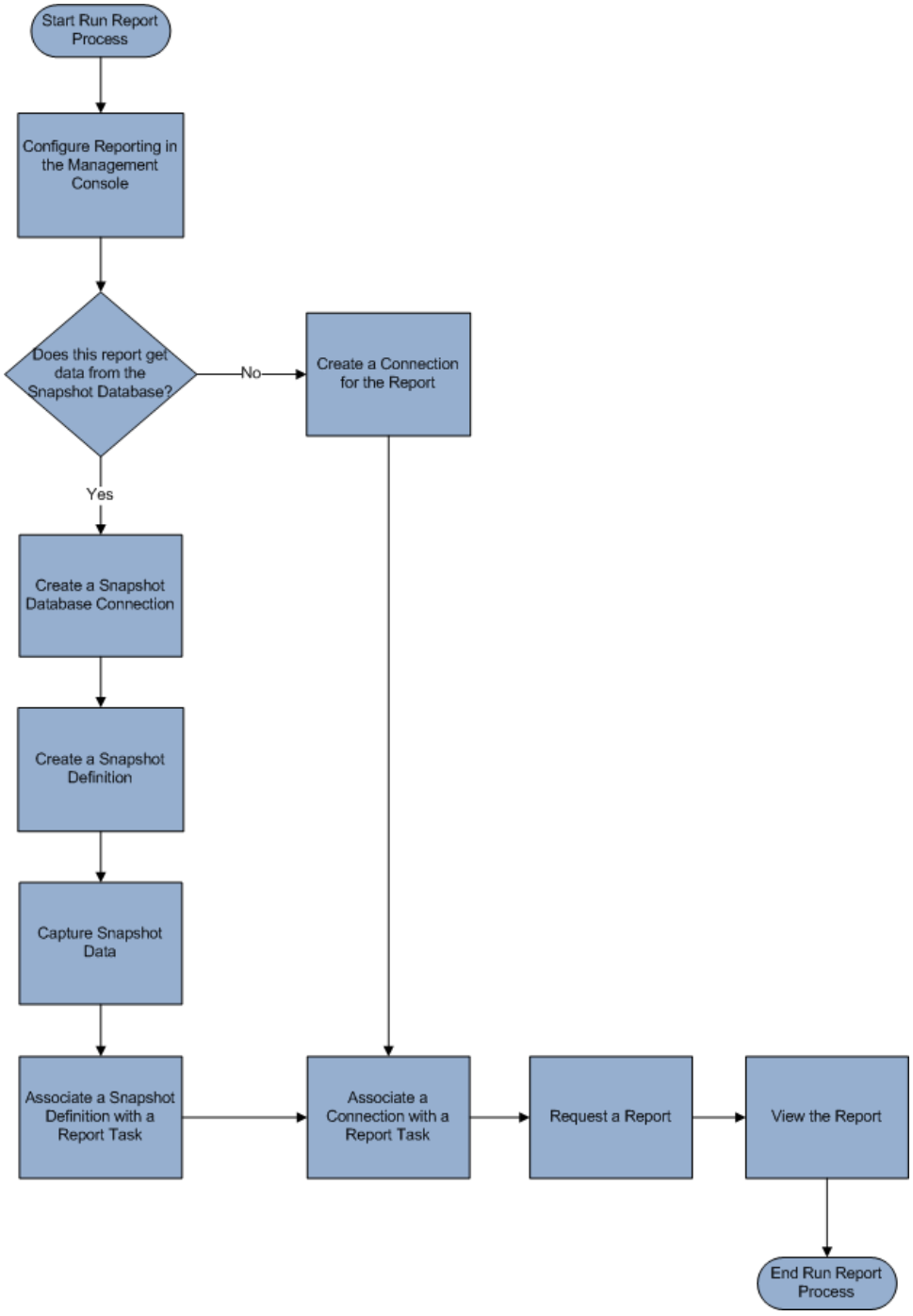
1. Create a connection object with the data source information for the report.
2. Modify the Report Task in Identity Management and add the connection object to the task.
3. Request the report using one of the following methods:
   - Run the report immediately
   - Schedule the report
4. View the report in the User Console.

Once the initial configuration for your report is complete, you can then request a report within Identity Management. You can run a report immediately, or you can schedule a report to run at a later time. You can also create a recurring schedule for your report within Identity Management.

Lastly, you can view the report within the User Console, or you can export the report to various formats.

The Report Process

The following graphic details the process required to run and view reports:



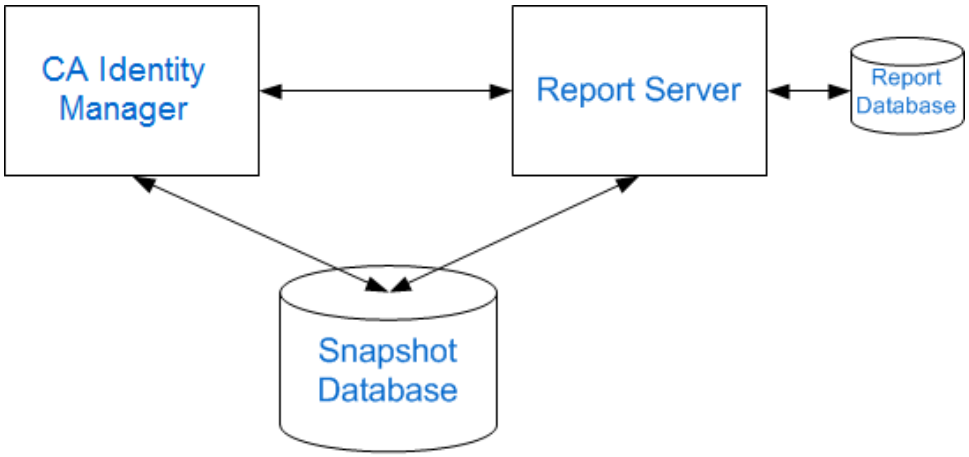
How to Run a Snapshot Report

Identity Management reports enable you to see the current state of a Identity Management environment. You can use this information to ensure compliance with internal business policies or external regulations.

You generate Identity Management reports from management data which describes the relationship between objects in Identity Management environment. Examples of management data include the following:

- Profile attributes of the users
- List of roles that contain a certain task
- The members of a role or group
- The rules that comprise a role

In Identity Management, the reporting setup requires the following three major components:



Note: The Snapshot Database in this illustration graphic could also be the Audit Database or Workflow Database.

Report Server

Also known as CA Business Intelligence, this server generates reports, communicating directly with Identity Management and the Snapshot Database.

Report Database

The database where the CA Report Server (Business Objects) stores its own data.

Identity Management

Identity Management allows you to export Identity Management object data to the Report Database.

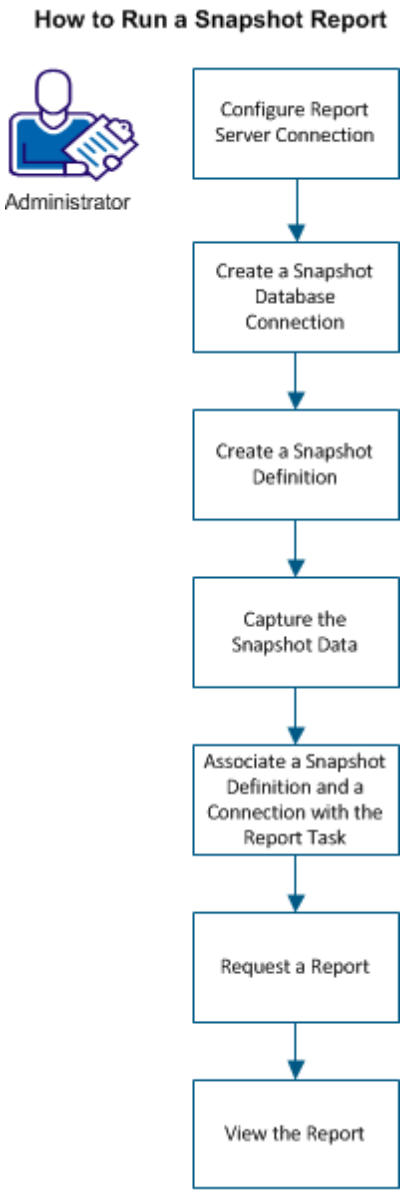
Snapshot Database

A separate database containing the snapshot data of objects in Identity Management

Important! The Report Server uses Business Objects Enterprise. If you already have a Report Server in your environment and want to use it with Identity Management, the minimum version required by Identity Management is CA Business Intelligence 3.2 SP5.

A snapshot report include data from the Snapshot Database, which contains information from the Identity Management object store and the user store. An example of a snapshot report is the User Profile report. You define the snapshot data that is added to the Snapshot Database and using snapshot definitions, specify the information to include.

The following graphic illustrates the process for running a snapshot report:



To run the snapshot report, perform the following steps:

1. Configure the Report Server connection (see page 255).
2. Create a snapshot database connection.
3. Create a Snapshot Definition.
4. Capture the snapshot data.
5. Associate a snapshot definition and a connection with the report task.
6. Request a report.
7. View the report (see page 258).

Configure the Report Server Connection

Configure the connection between Identity Management and the Report Server.

Note: We recommend that all systems involved in reporting be set to the same time zone and time.

To configure reporting

1. In the User Console, click Tasks, System, Reporting, Report Server Connection.
2. Enter the Report Server settings. Note the following:
   - Host Name and Port—hostname and the HTTP URL port number of the system where the Report Server is installed.
   - Reports folder name—location of the default Identity Management reports.
   - User ID—user created for the Report Server.
   - Password—password for the user created in the Report Server.
   - Secure Connection—select the checkbox to enable Secure Sockets Layer (SSL) connection between Identity Management and Report Server.

     Note: Before you select the Secure Connection checkbox, verify that you have installed the certificate from the BOServer. For more information about how to configure SSL, see the Chapter "Report Server Installation" in the *Installation Guide*.
   - Web Server—Set to Non-IIS for Tomcat
3. Click Test Connection to verify the connection.
4. Click Submit.

The reporting connection is established.

Create a Snapshot Database Connection

Identity Management needs to know where to export snapshot data to. Create a database connection from Identity Management to the Snapshot Database.

To create a snapshot database connection

1. In the User Console, click Tasks, Reports, Snapshot Tasks, Manage Snapshot Database Connection, Create Snapshot Database Connection.
2. Create a new snapshot database connection by completing all the necessary fields.
3. Click Submit.

   A new Snapshot Database connection is created.

Create a Snapshot Definition

A snapshot reflects the state of objects in Identity Management at a given time. You use this snapshot data to build a report. To capture Identity Management object data, you create a snapshot definition that exports the data to the Snapshot Database. Using the snapshot definition, you define the rules to load users, endpoints, admin roles, provisioning roles, groups, and organizations.

Follow these steps:

1. In the User Console, go to Tasks, Reports, Snapshot Tasks, Manage Snapshot Definition, Create Snapshot Definition.
2. Select Create or Copy an object of type Snapshot Type.
3. Click Ok.
4. Under the Profile tab, complete the following fields to create a snapshot definition profile:

   Snapshot Definition Name

   Identifies the unique name that is given for the snapshot definition.

   Snapshot Definition Description

   Displays any additional information that you want to describe the snapshot.

   Enabled

   Specifies that Identity Management creates a snapshot that is based on the current snapshot definition at the scheduled time.

   Note: If this option is not selected, the snapshot definition is not captured at the scheduled time. Also, the snapshot definition is not listed in the Capture Snapshot Data screen.

   Number of snapshots retained

   Specifies the number of successful snapshots retained in the Snapshot Database.

   Note: If you do not specify a value for this field, Identity Management stores unlimited snapshots.
5. On the Snapshot Policies tab, select the objects that are related to the policies to export.
6. On the Role Settings tab, select one or more role components and available attributes for the snapshot to export.

   Note: In the Snapshot Policies tab, if you select Access Role, Admin Role, or Provisioning Role object, select the attributes in the Role Settings tab.
7. On the User Attributes Details tab, select one or more user attributes for the snapshot to export.

Note: In the Snapshot Policies tab, if you select only the User object, by default, all user attributes related data are exported.

8. On the Endpoint Account Attributes tab, select one or more account attributes for an endpoint type.

   Note: For a selected endpoint type, by default, all data related to endpoint account attributes is exported. To capture data related to a specific attribute, select the appropriate attribute. For more information about selecting attributes that are necessary to export for an endpoint type, see the Default Reports section in the *Configuration Guide*.

9. (Optional) Select Export Orphan Accounts check box to include endpoint accounts with no global user in the Provisioning Server.

   Note: To export report data for non-standard, non-standard-trend and orphan account reports, select exceptionAccount attribute and Export Orphan Accounts check boxes.

10. Click Submit.

    Identity Management is configured to create snapshots of the objects mentioned in the snapshot definition.

Now that you have created a snapshot definition, you can capture snapshot data immediately or can schedule the snapshot data export at a later time. The topic Capture Snapshot Data provides more information.

More information:

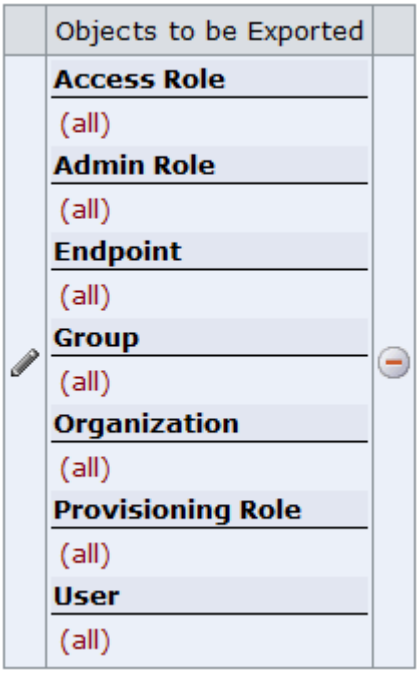
Recurrence Tab (see page 246)

Example: Creating a Snapshot Definition for a User Entitlement Data

The following example illustrates the process to create a snapshot definition for a user entitlement report:

1. In the User Console, go to Tasks, Reports, Snapshot Tasks, Manage Snapshot Definition, Create Snapshot Definition.
2. Select Create a new object of type Snapshot Type.
3. Enter the Snapshot definition name, description, and the number of snapshots retained.
4. In the Snapshot Policy Definition Tab, click Add.

   From the drop-down, select the user and select All. Similarly, add Endpoint, Provisioning Role, Admin Role, Access Role, Organization, and Group as shown in the following screen:



5. In the Role Setting Tab, select all the user role checkboxes.
6. In the User Attributes Tab, select the required attributes from the Available Values list and move them to the Current Values list.
7. Click Submit.

Manage Snapshots

Identity Management lets you view, modify and delete your snapshot definitions. When you view or modify a snapshot definition, the Profile and Maintenance tabs are shown. The Maintenance tab will only appear after a snapshot has been captured once. From the Maintenance tab, you can delete your snapshots (even if the status of the snapshot is failed).

To view, modify, or delete a snapshot definition, go to Reports, Snapshot Tasks, Manage Snapshot Definition and click on the task you want to execute.

Note: If a snapshot definition is being used to export data to the Snapshot Database, you cannot delete the snapshot definition. When you delete a snapshot definition that is being used, the export of the data to the Snapshot Database will stop, but the snapshot definition will still be available.

Capture Snapshot Data

If you want to capture snapshot data immediately or schedule the snapshot data export at a later time or on a recurring schedule, run the Capture Snapshot Data task. This task exports the data immediately (defined by the snapshot definition) to the Snapshot Database.

Important! Exporting snapshot data can take a long time if you have a large amount of data to export. We recommend you schedule your snapshots when exporting numerous data.

To capture snapshot data

1. In the User Console, go to Tasks, Reports, Snapshot Tasks, Capture Snapshot Data.
2. Select Execute now to run the data export immediately, or select Schedule new job (see page 246) to run the data export at a later time or on a recurring schedule.
3. Click Next.
4. Choose a snapshot definition.
5. Click Submit.

   Snapshot data is exported to the Snapshot Database.

Note: If the Capture Snapshot Data task seems to be taking a long time, you can check the progress of the task by going to the System tab and clicking View Submitted Tasks.

Recurrence Tab

Use this tab to schedule your job. The fields in this tab are as follows:

Execute now

Runs the job immediately.

Schedule new job

Schedules a new job.

Modify existing job

Specifies that you want to modify a job that already exists.

Note: This field appears only if a job has already been scheduled for this task.

Job Name

Specifies the name of the job you want to create or modify.

Time Zone

Specifies the server time zone.

Note: If your time zone is different from the server's time zone, a drop-down box is displayed so you can select either your time zone or the server's time zone when scheduling a new job. You cannot change the time zone when modifying an existing job.

Daily schedule

Specifies that the job runs every certain number of days.

Every (number of days)

Defines how many days between job runs.

Weekly schedule

Specifies that the job runs on a specific day or days and time during a week.

Day of Week

Specifies the day or days of the week the job runs.

Monthly schedule

Specifies a day of week or day of month that the job runs on a monthly basis.

Yearly schedule

Specifies a day of week or day of month that the job runs on a yearly basis.

Advanced schedule

Specifies additional scheduling information.

Cron Expression

For information about filling out this field, see the following:

http://quartz-scheduler.org/api/2.1.0/org/quartz/CronExpression.html

Execution Time

Specifies the time of day, in 24-hour format, that the job is run. For example, 14:15.

Associate a Snapshot Definition with a Report Task

Assign a snapshot definition to a report task so Identity Management knows which snapshot definition to use when running the report. Also, information for Identity Management reports can come from multiple sources, and each report should be associated with a specific data source, depending on the information you want to view in the report.

To associate a snapshot definition and connection with a report task

1. In the User Console, go to Tasks, Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for the report task you want to associate a snapshot definition with.
3. Go to the Tabs tab and click on the Edit button next to the Associate Snapshot Definitions tab.
4. Click Add.
5. Search for the snapshot definition to associate with the report task and click Select.

   When associating a snapshot definition with a report task, note the following:

   - A report can be associated with one or more snapshot definitions.
   - A snapshot definition can be associated with more than one report.
   - Multiple snapshots associated with a single report task must not use the same recurrence time.
6. Click Ok.
7. Go to the Search tab and click Browse to locate the search screens.
8. Edit the search screen for the report task and choose rptParamConn under Connection Object for the Report.
9. Click Ok.
10. Click Select.
11. Click Submit.

Synchronize Endpoint Accounts with Account Templates

This task synchronizes an endpoint account after modification of an associated account template. For example, perhaps an Active Directory account has no groups, but the associated account template is defined to include groups.

Follow these steps:

1. Log in to the User Console.
2. Select Tasks, Endpoints, Manage Endpoints, Check Endpoint Account Synchronization.
3. Select an endpoint.

   A screen appears showing accounts on that endpoint, associated account templates, and which attributes are not synchronized.
4. Click Synchronize to make the attributes for those accounts match what is defined in the account template.

   Changes that you make to account templates affect existing accounts as follows:

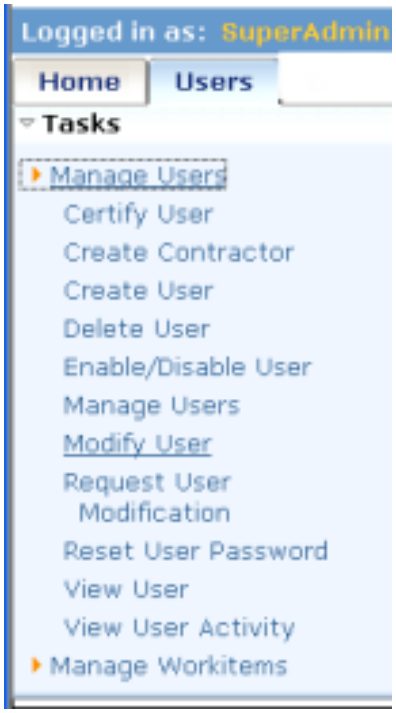
   - If you change the value of a capability attribute, the corresponding account attribute is updated to be synchronized with the account template attribute value. See the description of weak and strong synchronization.
   - Certain account attributes are designated by the connector as not being updated on account template changes. Examples include certain attributes that the endpoint type allows to be set only during account creation, and the Password attribute.

A Sample Admin Task

When you create an admin task, you define the content and layout of screens in the task, including:

- The name of the task
- The category where the task appears
- The tabs and fields to use in the task, and field display properties
- The fields an administrator can use in a search query, and the fields displayed in the search results

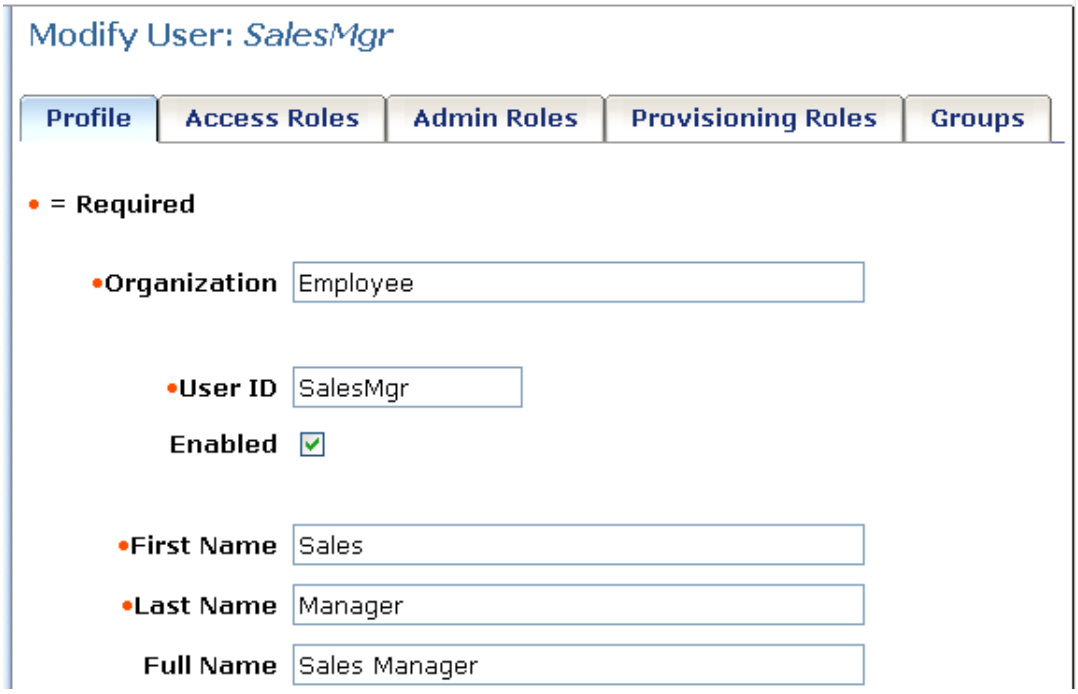
To understand the elements of a task, consider the Modify User task. In this case, Users is the category, Manage Users is a subcategory, and Modify User is the task. You create the category and task names when you create a task.



When you choose Modify User, a search screen appears. A *search screen* provides options for finding the object to view or modify. Each option is called a *filter*, which is a limit to the objects found by the search.

After you fill in the search screen, a screen with tabs appears. For example, the following figure shows the tabs for the Modify User task. The Profile tab appears first and shows user attributes; the other tabs show role and group privileges for the user.

For the task you create, you decide which tabs to include and determine their order and content.



For example, using the Modify User task as a template, you could create a Modify Contractor task, which has changes to:

- The fields on the Profile tab
- The tabs to include in the task and their content
- The category under which the task appears

  You might create this task under a new category, Contractor.

The Modify Contractor task includes some of the fields on the Profile tab in the Modify User task plus other fields, such as the start date of the contract and the contractor's company. Administrators can search for a contractor by searching on the contractor's name, company, and start date.

The new task also includes a Contractor Roles tab where you add roles for contractors.

Request a Report

To view the report, request a report to a user with report administration privileges. Approval is required because some reports may require a long time or significant system resources to run. If your report request requires an approval, the system sends you an email alert.

Follow these steps:

1. Log in to the User Console with report tasks user privileges.
2. Select Tasks, Reports, Reporting Tasks, and Request a Report.

   A list of reports appears.
3. Select the report that you want to request.

   A parameters screen appears.

Provide any parameter information required.

Note: If you are running a snapshot report and no snapshots are available for this report, you must first capture a snapshot.

- Some reports show system status at a specific point in time. When you request this type of report, you select a point in time for which you want to see report data. This point in time is named a *snapshot*.

  Note: The snapshot dates and times you can choose are predetermined. Typically, your system administrator, or another user with report administration privileges, configures snapshots. If no snapshots are available for the report you want to request, contact a system administrator.

- Some reports show activity over a time period. Titles for these reports usually begin with the word *Audit*. When you request this type of report, you specify a time period for which you want to see report data. For example, you can run the Audit-Reset Password report for the past 30 days.

4. Click Schedule Report, and select a schedule for your report.

   Now

   Specifies that the report runs immediately.

   Once

   Specifies that the report runs once, during a specific time period. Select the start date, end date, start time, and end time when you want to generate the report.

   Note: Consider selecting this option if the report you are requesting requires a large amount of data. To conserve system resources, choose a time when there is less system activity.

5. Click Submit.

   The report request is submitted. Depending on your environment configuration, the request runs immediately, or it runs after approval by an administrator.

   Typically, a system administrator or another user with report administration privileges must approve a report request before the system completes it. Approval is required because some reports can require a long time or significant system resources to run. If your report request requires approval, the system sends you an email alert.

View the Report

Depending on your environment configuration, a report will be available to view when an administrator approves the request for that report. If your report request is pending approval, the system sends you an email alert. The report that you want to view does not appear in the search list until it is approved.

Note: In order to view reports in Identity Management using the View My Reports task, enable third-party session cookies in your browser.

Follow these steps:

1. In the User Console, go to Tasks, Reports, Reporting Tasks, and click View My Reports.
2. Search for the generated report that you want to view.

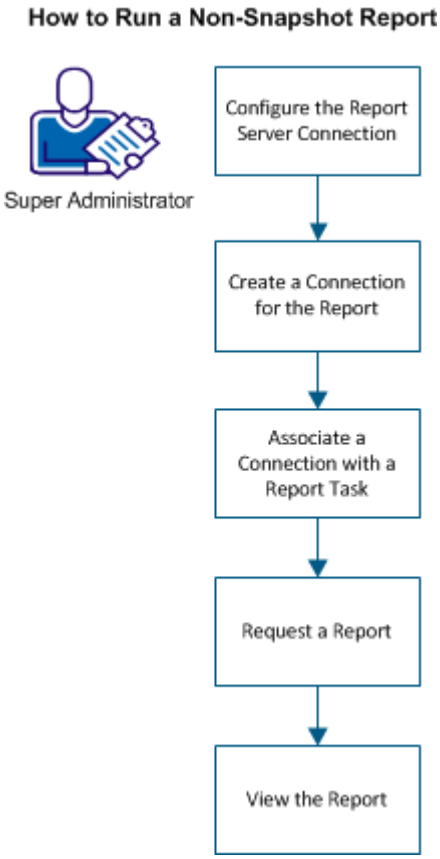
   Both recurrence generated reports and on-demand report instances are displayed.

   Note: If the status of the report is Pending/Recurring, the report is not generated and may take time to complete.
3. Select the report that you want to view.
4. (Optional) Click Export this report (top left corner) to export the report to the following formats:
   - Crystal Report
   - PDF
   - Microsoft Excel (97-2003)
   - Microsoft Excel (97-2003) Data-Only
   - Microsoft Excel (97-2003) - Editable
   - Rich Text Format (RTF)
   - Seperated Values (CSV)
   - XML

How to Run a Non-Snapshot Report

A non-snapshot report includes data from other data sources, such as the Audit, Workflow, or Task Persistence databases. Identity Management includes default audit reports with prefix "Audit-" to their name, in the user console. By default, Identity Management includes only Audit reports, but you can create your own custom reports that can include data from any data source.

This scenario describes how a Super Administrator configures a report database connection and run a non-snapshot report.



The following diagram illustrates the process running a snapshot report:

To run the non-snapshot report, perform the following steps:

1. Configure the report server connection. (see page 255)
2. Create a connection for the report.
3. Associate a connection with the report task.
4. Request a Report.
5. View the Report. (see page 258)

Configure the Report Server Connection

To collect data from the report server, you must configure the connection to the Report Server. Before you begin the procedure, gather the following information about the reporting server:

Name	Description
Host Name	The host name of the machine where the report server is installed
Port	The port name of the machine where the Report Server is installed
Reports folder name	The location of the default Identity Management reports.
User ID	Specifies the user created for the Report Server.
Password	Specifies the password for the user created in the Report Server.
Secure Connection	Specifies the security connection for the report server. Select the checkbox to enable Secure Sockets Layer (SSL) connection between Identity Management and Report Server. **Note**: Before you select the Secure Connection checkbox, verify that you have installed the certificate from the BOServer. For more information about how to configure SSL, see the Chapter "Report Server Installation" in the *Installation Guide*.
Web Server	Specifies the web server. Set to Non-IIS for Tomcat.

Note: We recommend that all systems involved in reporting be set to the same time zone and time.

Follow these steps:

1. In the User Console, click System, Reporting, Report Server Connection.
2. Enter the Report Server settings.
3. Click Test Connection to verify the connection.
4. Click Submit.

The reporting connection is established.

Create a Connection for the Report

Information for Identity Management reports can come from multiple sources. To specify connection details to additional data source for the report, create a JDBC Connection within Identity Management.

Follow these steps:

1. In the User Console, go to Tasks, System, JDBC Connection Management, Create JDBC Connection.
2. Create a new connection object, or choose a connection object based on a specific JNDI data source.
3. Complete all the necessary fields, and click Submit.

   A new JDBC Connection is created.

Important! We recommend that you do *not* use the Identity Management object store database for generating reports.

Associate a Connection with a Report Task

Information for Identity Management reports is captured from multiple sources and each report must associate with a specific data source, depending on the information you want to view in the report.

To associate a connection with a report task

1. In the User Console, go to Tasks, Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for the report task you want to associate a connection with.
3. Go to the Search tab and click Browse to locate the search screens.
4. Edit the search screen for the report task and choose a connection under Connection Object for the Report.
5. Click Ok.
6. Click Select.
7. Click Submit.

Request a Report

To view the report, request a report to a user with report administration privileges. Approval is required because some reports may require a long time or significant system resources to run. If your report request requires an approval, the system sends you an email alert.

Follow these steps:

1. Log in to the User Console with report tasks user privileges.
2. Select Tasks, Reports, Reporting Tasks, and Request a Report.

   A list of reports appears.
3. Select the report that you want to request.

   A parameters screen appears.

   Provide any parameter information required.

   Note: If you are running a snapshot report and no snapshots are available for this report, you must first capture a snapshot.

   - Some reports show system status at a specific point in time. When you request this type of report, you select a point in time for which you want to see report data. This point in time is named a *snapshot*.

     Note: The snapshot dates and times you can choose are predetermined. Typically, your system administrator, or another user with report administration privileges, configures snapshots. If no snapshots are available for the report you want to request, contact a system administrator.
   - Some reports show activity over a time period. Titles for these reports usually begin with the word *Audit*. When you request this type of report, you specify a time period for which you want to see report data. For example, you can run the Audit-Reset Password report for the past 30 days.
4. Click Schedule Report, and select a schedule for your report.

   Now

   Specifies that the report runs immediately.

   Once

   Specifies that the report runs once, during a specific time period. Select the start date, end date, start time, and end time when you want to generate the report.

   Note: Consider selecting this option if the report you are requesting requires a large amount of data. To conserve system resources, choose a time when there is less system activity.

5. Click Submit.

   The report request is submitted. Depending on your environment configuration, the request runs immediately, or it runs after approval by an administrator.

   Typically, a system administrator or another user with report administration privileges must approve a report request before the system completes it. Approval is required because some reports can require a long time or significant system resources to run. If your report request requires approval, the system sends you an email alert.

View the Report

Depending on your environment configuration, a report will be available to view when an administrator approves the request for that report. If your report request is pending approval, the system sends you an email alert. The report that you want to view does not appear in the search list until it is approved.

Note: In order to view reports in Identity Management using the View My Reports task, enable third-party session cookies in your browser.

Follow these steps:

1. In the User Console, go to Reports, Reporting Tasks, and click View My Reports.
2. Search for the generated report that you want to view.

   Both recurrence generated reports and on-demand report instances are displayed.

   Note: If the status of the report is Pending/Recurring, the report is not generated and may take time to complete.
3. Select the report that you want to view.
4. (Optional) Click Export this report (top left corner) to export the report to the following formats:
   - Crystal Report
   - PDF
   - Microsoft Excel (97-2003)
   - Microsoft Excel (97-2003) Data-Only
   - Microsoft Excel (97-2003) - Editable
   - Rich Text Format (RTF)
   - Seperated Values (CSV)
   - XML

Set Reporting Options

Configure the number of report instances a user can generate for a specific report.

To modify the reporting options

1. Select Tasks, Reports, Reporting Tasks, Set Reporting Options.

   Identity Management connects to the IAM Report Server and retrieves a list of all the reports.

2. Choose a report, and click Modify.

   The report's attributes pane appears.

3. Edit the following fields:

   Name

   Specifies the display name for the selected report.

   Number of Instances

   Specifies the number of allowed instances that can be generated by a user for this report.

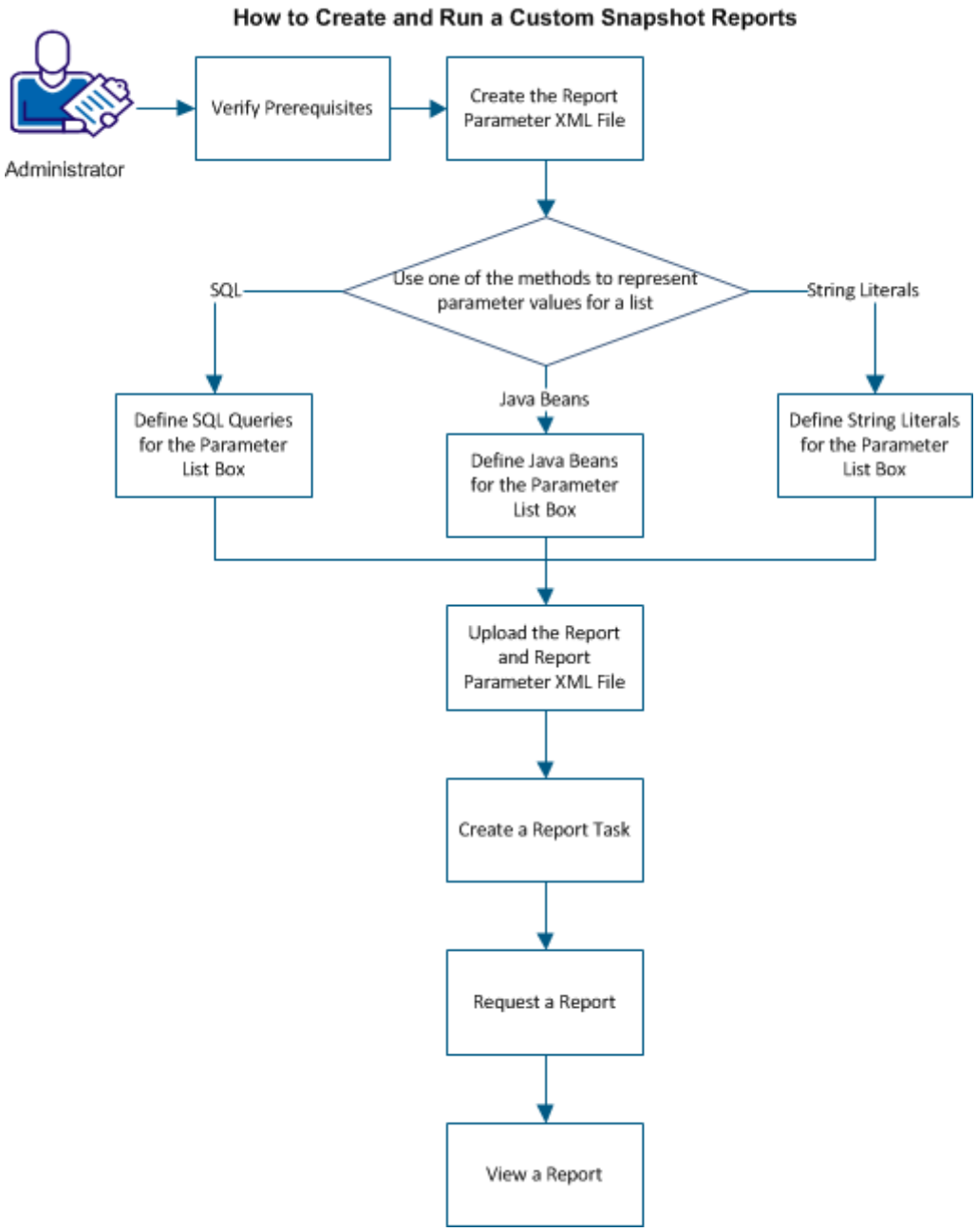
4. Click Ok.

   The reporting attributes are changed.

How to Create and Run a Custom Snapshot Report

Identity Management allows you to create and customize reports to suit your business needs. Identity Management provides a Report Parameter XML file which includes all the parameters related to reporting attributes. According to your business needs, you can choose the required attributes in order to populate report data from the snapshot data source.

The following graphic illustrates the process to create and run a custom snapshot report:



As a system administrator, complete the following steps:

1. Verify prerequisites (see page 262)
2. Create the Report Parameter XML file (see page 262)
3. Use one of the following methods to represent parameter values for a list:
   - Define SQL Queries for the Parameter List Box (see page 265)
   - Define Java Beans for the Parameter List Box (see page 266)

- Define String Literals for the Parameter List Box (see page 266)

4. Upload the Report and Report Parameter XML file (see page 266)
5. Create the report task
6. Request a report
7. View a report (see page 258)

Create a Report in Crystal Reports

In order to use custom reports in Identity Management, create a report (RPT file) in Crystal Reports Developer. For more information on how to create a report in Crystal Reports, refer to your Crystal Reports documentation.

Note: To reference the Identity Management schema in order to create custom reports, the Identity Management database schema is in the following location:

```
[set the Installation Path variable]\db\objectstore
```

Create the Report Parameter XML File

A parameter is one of the fields in a report that can be used to filter reports. You can generate a report by filtering the data using parameters. To allow the customization of the report search screen, each report (RPT file) is associated with a Report Parameter XML file. In Identity Management, you can create report tasks and create search screen so that a user can enter or select required data during the generation of a report.

Note: You only need a report parameter XML file if the report queries attributes on the object.

The Report Parameter XML file must have the same name as the report (RPT file) with a .xml extension. For example, if you upload a report named test1.rpt into the Report Server, your XML file should be named test1.xml.

The Report Parameter XML file has the following elements:

<product>

Identifies the product for which the parameters are used. You can create different parameters for multiple products using the same parameter XML file.

<screen>

Defines the parameters that displayed on a screen. You can use the screen element to bind the parameters to a specific screen. The screen ID is alphanumeric and unique, and is used to identify the screens and their parameters.

<parameters>

Specifies the collection of parameters for a screen.

<param>

Defines the parameter element that passes along specified data to the report. The following attributes are used in the <param> element:

id

Defines which parameter in the report to associate with.

Note: The id must have the same name as the parameter in the Crystal Report.

name

This field is not currently used by Identity Management. Set this attribute to the same value as id.

displaytext

Specifies the user-friendly text to be displayed in the screen for the parameter.

type

Defines the type of parameter. The screen display changes based on this attribute. The parameter types supported are as follows:

- **Text Box**

  Example: <param id="param1" displaytext="First Name" name="param1" type="string"/>

- **Date and Time**

  Example: <param id="dateVal" displaytext="Date" name="dateVal" type="date_str"/>

  <param id="timeVal" displaytext="Time" name="timeVal" type="time_str"/>

  <param id="datetimeVal" displaytext="Date & Time" name="datetimeVal" type="date_time_str"/>

- **Drop-down List**

  Example: <param id="lastname1" displaytext="Name" name="lastname1" type="dropdown" default="key1%1FMy Value1%1Ekey2%1FMy Value2" selected_value="My Value2"/>

- **List Box**

  Example: <param id="lstlastname1" displaytext="Name" name="lstlastname1" type="listbox" rows="10" default="key1%1FSuper%1Ekey2%1Fsql2kSuser01%1E key1F%Super"/>

- **Radio Box**

  Example: <param id="optionslist" displaytext="Option 1" name="optionslist" type="radiobox" value="option1"/>

  <param id="optionslist" displaytext="Option 2" name="optionslist" type="radiobox" value="option2"/>

  <param id="optionslist" displaytext="Option 3" name="optionslist" type="radiobox" value="option3"/>

- **Check Box**

  Example: <param id="enabled" displaytext="Enabled" name="enabled" type="checkbox"/>

row

Defines how many rows are visible in a list box.

Default: 5

default

Defines the default value displayed on the screen for a given parameter. This attribute can be used with the string, list box, and drop-down list types.

Define SQL Queries for the Parameter List Box

You can define SQL queries as part of a list box or drop-down box in the Report Parameter XML file. When you associate a parameter to the report and create a report task, the parameter is displayed in the list box or drop-down box to the user. To use SQL in the drop-down box or list box parameter, provide a valid SQL statement in the sql attribute.

Example:

```
<param id="lstlastname2" displaytext="Name" name="lstlastname2" type="sqlstr"
multiselect="true" sql="select lastname, lastname from tblusers where firstname like
'S%/>
```

In the previous example, all the last names of users with a first name that starts with *S* are provided to the report.

However, the condition of the first name that starts with *S* is a static one. This query is not flexible enough for a user to load the value-based on the parameter value entered in one of the previous screens that was used in the same report parameter group. In order to use the previous value that was entered in another screen, the SQL statement can be augmented with *##<parameter id>##*.

For example, if you have a parameter with the id=User, which was of type String:

```
<param id="User" displaytext="First Name" name="firstname" type="string"/>
```

If you want to use the input value for that parameter in SQL, the SQL statement could be as follows:

```
<param id="lstlastname2" displaytext="Name" name="lstlastname2" type="sqlstr"
multiselect="true" sql="select lastname, lastname from tblusers where firstname like
'##User##'/>
```

Identity Management replaces ##User## with the value entered for the parameter with the id=User.

Note: The parameter value to be substituted cannot be in the same screen as the SQL parameter. For example, if the *lstlastname2* is in screen 3, the User parameter should be in one of the previous screens.

Define Java Beans for the Parameter List Box

If using SQL is not ideal, you can use java beans to calculate values and provide the list of <key, value> pairs to Identity Management. The java beans should be in the classpath of Identity Management.

Example:

```
<param id="lastname2" displaytext="Name using Javabean" name="lastname2"
type="dropdown" class="com.ca.ims.reporting.unittests.TestDataCollector"/>
```

In the previous example, the TestDataCollector retrieves the values in its own way and sends the data for the drop-down list to the report. The <key, value> pairs are separated by %1F.

Be sure the java bean is in the iam_im.ear\custom directory.

Note: For more information about implementing java beans, see your Business Objects documentation.

Define String Literals for the Parameter List Box

The simplest way of representing the parameter values for a list or drop-down box is by using string literals. The key value is delimited by %1F and each <key, value> pair is separated by %1E.

Example:

```
<param id="lastname1" displaytext="Name" name="lastname" type="dropdown"
default="key1%1FMy Value1%1Ekey2%1FMy Value2" selected_value="My Value2"/>
```

Upload the Report and Report Parameter XML File

After you create the report (RPT) and the corresponding Report Parameter XML file, upload both files to the Report Server (Business Objects).

Follow these steps:

1. Log in to the Business Objects Central Management Console.
2. Click Folders.
3. Select the IM Reports folder.
4. Create an Object Package.
5. In the new object package, browse to add the Crystal Report.

6. Browse for the new report (RPT) you created.

   Note: Ensure that the IM Reports folder is selected as the folder to save the report in.

7. Click OK.

   Crystal Report file is added.

8. In the new object package, Add a new Local Document and browse for the new Report Parameter XML file.
9. Select the File Type as *Text*.
10. Click OK.

The Report and the Report parameter xml file is now uploaded. In order to verify, go to the IM Reports folder and verify that both the new files are available.

Create the Report Task

Report tasks are used to create, manage, view, and delete the templates for the reports that are generated in the User Console.

Follow these steps:

1. In the User Console, go to Tasks, Roles and Tasks, Admin Tasks, Create Admin Task.
2. Select Create a new admin task and click OK.
3. In the Profile tab, complete the following fields:

   Name

   Defines the name of the report. Each report task name must be unique.

   Tag

   Defines a unique identifier for the task. It is used in a URL, a web service, or properties files. It must consist of letters, numbers, and/or underscores, beginning with a letter or underscore.

   Category

   Specifies the category to which the current task belongs.

   Note: Select the Reports category.

   Category 2

   Specifies the sub-category to which the current task belongs. Enter any string in this field.

   Primary Object

   Specifies the object on which the task operates.

   Note: Select Report Instance as the primary object.

   Action

   Specifies the action that is performed on the primary object.

   Note: Select Create as the action.
4. To create a new search screen for the report task, perform the following steps:
   a. In the Search tab, click Browse to locate the search screens.

      The list of available search screens is displayed.

   b. Click New.

      The Create Screen pane appears.

   c. Select Report Template Selection Screen from the list, and click OK.

      Identity Management connects to the Report Server and displays all the reports.

   d. Complete the following fields:

Name

Defines the name of the report. Each report task name should be unique.

Tag

Acts as a unique identifier within a task. It can contain ASCII characters (a-z, A-Z), numbers (0-9), or underscore characters, beginning with a letter or underscore.

Title

Defines the title of the new search screen. The title must be unique.

Report Template

Identifies the report to associate with the search screen.

Note: Choose one of the reports you added to the Report Server.

Connection Object for the Report

Defines the connection details of the data source to be used for the report.

5. Click Ok.

   The new search screen is now created for reports.

6. In creating a Tabs tab for the report task, perform the following steps:
   a. Click Tabs.

      The tabs that are visible to the user are displayed.

   b. Select the Standard Tab Controller.

   c. If your report uses a snapshot definition, perform the following steps:
      a. From Which tabs should appear in this task?, select Associate Snapshot Definitions.

         The Associate Snapshot Definitions tab is added to the list of tabs.

      b. Click  to edit the Associate Snapshot Definitions tab.

      c. Click Add to associate the report task with a snapshot definition.

         A list of available snapshot definitions appear.

      d. Select a Snapshot Definition and click OK.

         The report task is associated with a snapshot definition.

   d. Click Submit.

      The report task is created.

   e. Assign the new report task to an Admin role.

      The Identity Management Admin role users can use the new report task.

The report task is now ready to be used by the Admin.

Note: A report (RPT file) can only be associated with *one* report task.

Request a Report

To view the report, request a report to a user with report administration privileges. Typically, a system administrator or another user with report administration privileges must approve a report request before the system completes it. Approval is required because some reports may require a long time or significant system resources to run. If your report request requires an approval, the system sends you an email alert.

Follow these steps:

1. Log in to the User Console as a user who has access to the report tasks.
2. From the navigation menu, select Tasks, Reports, Reporting Tasks, Request a Report.

   A list of reports appears.
3. Select the report that you want to request.

   A parameters screen appears.
4. Provide any parameter information required.

   Note: If you are running a snapshot report and no snapshots are available for this report, you must first capture a snapshot.

   - Some reports show system status at a specific point in time. When you request this type of report, you select a point in time for which you want to see report data. This point in time is named a *snapshot*.

     Note: The snapshot dates and times you can choose are predetermined. Typically, your system administrator, or another user with report administration privileges, configures snapshots. If no snapshots are available for the report you want to request, contact a system administrator.
   - Some reports show activity over a time period. Titles for these reports usually begin with the word *Audit*. When you request this type of report, you specify a time period for which you want to see report data. For example, you can run the Audit-Reset Password report for the past 30 days.
5. Click Schedule Report, and select a schedule for your report.

   Now

   Specifies that the report runs immediately.

   Once

   Specifies that the report runs once, during a specific time period. Select the start date, end date, start time, and end time when you want to generate the report.

   Note: Consider selecting this option if the report you are requesting requires a large amount of data. To conserve system resources, choose a time when there is less system activity.
6. Click Submit.

The report request is submitted. Depending on your environment configuration, the request runs immediately, or it runs after approval by an administrator.

View the Report

Depending on your environment configuration, a report will be available to view when an administrator approves the request for that report. If your report request is pending approval, the system sends you an email alert. The report that you want to view does not appear in the search list until it is approved.

Note: In order to view reports in Identity Management using the View My Reports task, enable third-party session cookies in your browser.

Follow these steps:

1. In the User Console, go to Tasks, Reports, Reporting Tasks, and click View My Reports.
2. Search for the generated report that you want to view.

   Both recurrence generated reports and on-demand report instances are displayed.

   Note: If the status of the report is Pending/Recurring, the report is not generated and may take time to complete.
3. Select the report that you want to view.
4. (Optional) Click Export this report (top left corner) to export the report to the following formats:
   - Crystal Report
   - PDF
   - Microsoft Excel (97-2003)
   - Microsoft Excel (97-2003) Data-Only
   - Microsoft Excel (97-2003) - Editable
   - Rich Text Format (RTF)
   - Seperated Values (CSV)
   - XML

Troubleshooting

The following section details troubleshooting topics around reporting.

Viewing a Report Redirects To the Infoview Login Page

When viewing a report in Identity Management, you may be re-directed to the Business Objects Infoview login page.

View the report if redirected

1. Be sure that you are using the fully-qualified domain name of the CA Report Server (Business Objects).
2. Right-click on the Infoview login web page and select View Source.
3. Find the URL for the report.
4. Copy and paste the URL into a new browser window.
5. If you do not see the report, use an HTTP trace tool to provide more information.
6. If you do see the report, try the following to fix the browser settings:
   - Accept third-party cookies.
   - Allow session cookies.
   - Remove High security settings.

Generating User Accounts for over 20,000 Records

If over 20,000 records exist, some extra steps are necessary to generate user accounts report.

To generate a user accounts report for over 20,000 records

1. Open the Business Objects Central Management console.
2. Click Servers and select *servername*.pageserver.
3. Select Unlimited records for the entry Database Records To Read When Previewing Or Refreshing a Report.
4. Using a Crystal Reports designer, open the user accounts report.
5. Using Database, Set Datasource Location, set the database location to your snapshot database.
6. Save this change.
7. Using Database, Datasource Expert, right-click Command on the right side window.

   It shows the SQL syntax on the left side and the Parameter List.
8. Enter the parameter name as you find it in the Parameters Fields in the report template.

9. Change the query in the left side and add that parameter in the query.

   For example, if you have reportid parameter, the query will be:

   ```
   Select *  from endPointAttributes, endpointview, imreport6
   where endPointAttributes.imr_endpointid = endpointview.imr_endpointid and
   endPointAttributes.imr_reportid = endpointview.imr_reportid
    endpointview.imr_reportid = imreport6.imr_reportid and imreport6.imr_reportid
   = {?reportid}
   ```

10. Save the report.

Chapter 12: Workflow

This section contains the following topics:

Workflow Overview

The Identity Management workflow feature allows a Identity Management task to be controlled by a workflow process. A *workflow process* is one or more steps that must be performed before Identity Management can complete a task that is under workflow control. A *job* is a runtime instance of a workflow process.

WorkPoint Designer is software from Workpoint LLC, a subsidiary of Planet Group, Inc., that is integrated with Identity Management. WorkPoint Designer lets you manage workflow processes and workflow jobs.

A workflow process consists of one or more steps, called *activities*, that must be performed in order to accomplish some business task, such as creating or modifying an employee user account. Typically, a workflow process includes one or more manual activities which require an authorized user, or participant, to approve or reject the task.

A *participant* is a person who is authorized to perform a workflow activity. In Identity Management, participants are also called *approvers*, since they must approve or reject the task under workflow control. A *participant resolver* is a rule or set of criteria for determining who the participants are.

The individual manual activities in a workflow are called *work items* in Identity Management.

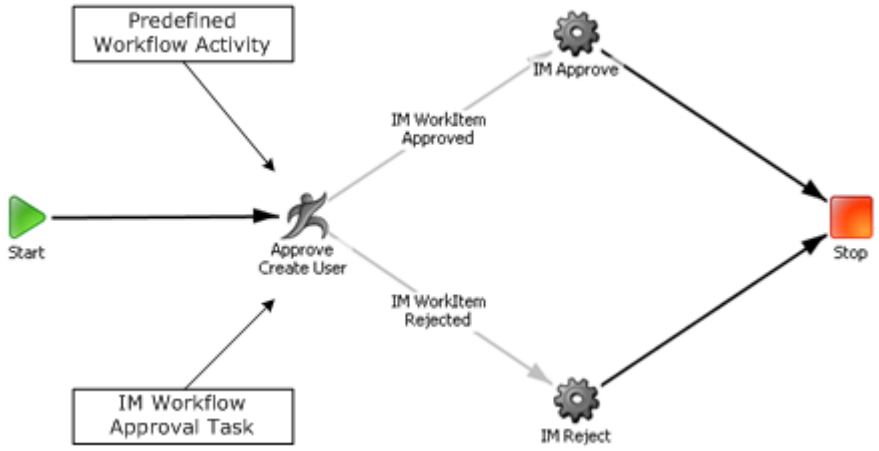
A *work list* is a workflow-generated list of approval tasks, or *work items*, that appears in the User Console of the participant authorized to approve the task.

WorkPoint Process Diagram

In general, Identity Management tasks trigger Identity Management events. For example, to create a user, an administrator selects a Create User task. When this task is initiated, the event CreateUserEvent is triggered.

The following diagram is an example of a simple workflow process (the predefined process CreateUserApproveProcess) as it appears in WorkPoint Designer. This process is invoked by a CreateUserEvent if the Create User task is under workflow control.

The process includes a manual activity, Approve Create User, which corresponds to a Identity Management workflow approval task of the same name. The participant must approve or reject the approval task, typically by clicking a button in the User Console, before the task under workflow control can run to completion.



Workflow and Email Notification

When you initiate a task, Identity Management submits the task for processing. When the task finishes, it displays an acknowledgement message as follows:

```
Confirmation:  Task completed.
```

However, if the task is under workflow control and it requires approval, the message is as follows:

```
Alert:  Task pending.
```

In addition to on-screen messages, Identity Management can automatically generate email notifications when:

- An event or task requiring approval or rejection by a workflow approver is pending.
- An approver approves an event or task.

- An approver rejects an event or task.
- An event or task is completed.

More Information:

Email Notifications (see page 365)

WorkPoint Documentation

For general information about workflow concepts and for instructions about workflow processes, activities, and jobs in WorkPoint Designer, see the WorkPoint documentation. To do so, open the following HTML page:

admin_tools\WorkPoint\docs\designer\default.htm

admin_tools

Defines the installation directory of the Identity Management administrative tools. The default installation directory is as follows:

- **Windows:** [set the Installation Path variable]\tools
- **UNIX:** [set the alternate Installation Path variable]/tools

Note: Workpoint is a third-party product installed with Identity Management. Identity Management supports a subset of functionality in WorkPoint. For example, Identity Management does not support the WpConsole. However, the WorkPoint documentation describes all functionality in the product. Portions of the Workpoint documentation do not apply to Identity Management users.

Workflow Control Methods

Identity Management provides two methods of placing tasks under workflow control.

Template Method

Identity Management includes workflow process templates you can use to place tasks under workflow control. The *template method* lets you use these templates to configure and manage workflow entirely from within the User Console. Introduced in Identity Management r12, these generic process templates can be configured to control most Identity Management tasks and events.

The template method enables the following new features:

- Both task-level and event-level workflow control
- Simplified participant resolver configuration for workflow approvers

- Work item delegation, which covers out-of-office scenarios by letting a user delegate another user to approve work items
- Work item reassignment, which lets a running task be reassigned to another user for approval

WorkPoint Method

Identity Management also includes a set of predefined workflow processes with default event mappings that correspond to specific Identity Management tasks. The *WorkPoint method* requires you to configure and customize these processes from within WorkPoint Designer. These predefined processes are compatible with releases prior to Identity Management r12.

The WorkPoint method also enables the following new features:

- Both task-level and event-level workflow control
- Work item delegation, which covers out-of-office scenarios by letting a user delegate another user to approve work items
- Work item reassignment, which lets a running task be reassigned to another user for approval

Note: For greater flexibility and ease of use, CA recommends that you use the template method whenever possible.

More Information:

Use Workflow Control - Template Method (see page 278)

Use Workflow Control - Template Method

Introduced in Identity Management r12, the template method allows you to configure workflow process templates in the User Console, without having to open WorkPoint Designer.

Template method advantages are:

- Multi-stage process templates can address most workflow needs without requiring customization within WorkPoint Designer.
- Templates support both task-level and event-level workflow control.
- The same workflow process template can be configured for use with many different tasks while the process design itself remains unchanged.
- Participant resolvers can be specified easily in the User Console.
- Work item delegation can be performed in the User Console.

Prerequisite: Enable Workflow

You must have workflow enabled before you can use it to control Identity Management tasks. By default, workflow is disabled.

Follow these steps:

1. In the Management Console, select an Environment.
2. Go to Advanced Settings, Workflow.
3. Select the Enabled check box, and click Save.

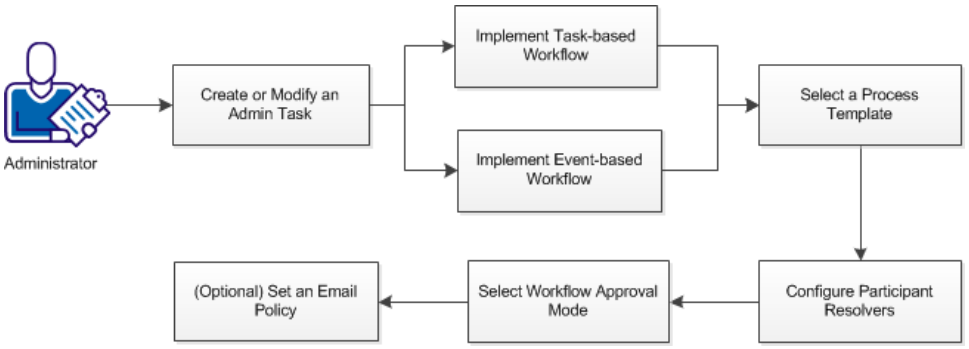
   Note: The Event Mappings on this screen apply only if you use the WorkPoint method to configure workflow. If you use the template method (recommended), do not map events to processes using this Management Console.
4. Restart the application server.
5. (Optional) Configure the WorkPoint Administrative Tools (see page 298).

More Information:

Map a Process to an Event Globally (see page 305)
Workflow Control Methods (see page 277)

Place Admin Tasks under Workflow Control - Template Method

As an administrator, you can place admin tasks under workflow control using the template method.



Follow these steps:

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify (or Create) Admin Tasks.
2. Search for the task you want under workflow control, and click Select.
3. Do *one* of the following:
   - Implement task-level workflow (see page 280) by clicking the Workflow Process edit button on the Profile tab.
   - Implement event-level workflow (see page 283) by selecting one or more events on the Events tab.
4. Select a process template (see page 286).
5. Configure participant resolvers (see page 289).
6. Select workflow approval mode.
7. (Optional) Set an email policy for the workflow process (see page 294).

   Note: If you select the EscalationApproval process, a field named Approval Timeout (min) is displayed. This field is specified in minutes and cannot be empty. By default, the time is set to 60 minutes.

After workflow control is configured, a user with the appropriate role performs the admin task, and the designated workflow participant approves or rejects the task or event.

Task or Event-Based Workflow

Identity Management lets you associate workflow processes with either tasks or events. This means that participants can approve or reject either an entire Identity Management task, or a specific event within a task.

For example, some Identity Management tasks generate several events, and an approver may need to review all events before deciding to approve or reject a request. This is possible under task-level workflow. When a workflow process is associated with a specific event within a task, an approver cannot see the overall task context within which a request is made.

Task-Level Workflow

Task-level workflow lets approvers review all events before deciding to approve or reject a request. Task-level workflow occurs before any task activity is processed. No events or nested tasks execute before the workflow process job begins.

If task-level workflow is rejected, no part of the task is executed.

Note: A task that is configured for task-level workflow control can also be configured for event-level workflow control simultaneously. Concurrent event-level workflow may be applied globally or for a specific task.

More Information

Event-Level Workflow (see page 283)
Global Process to Event Mapping (see page 303)

Task-Level Process Attribute

Workflow processes that are compatible with task-level workflow all have a special attribute defined within WorkPoint Designer. This process level user data attribute, called TASK_LEVEL, is set to true by default in the following process templates:

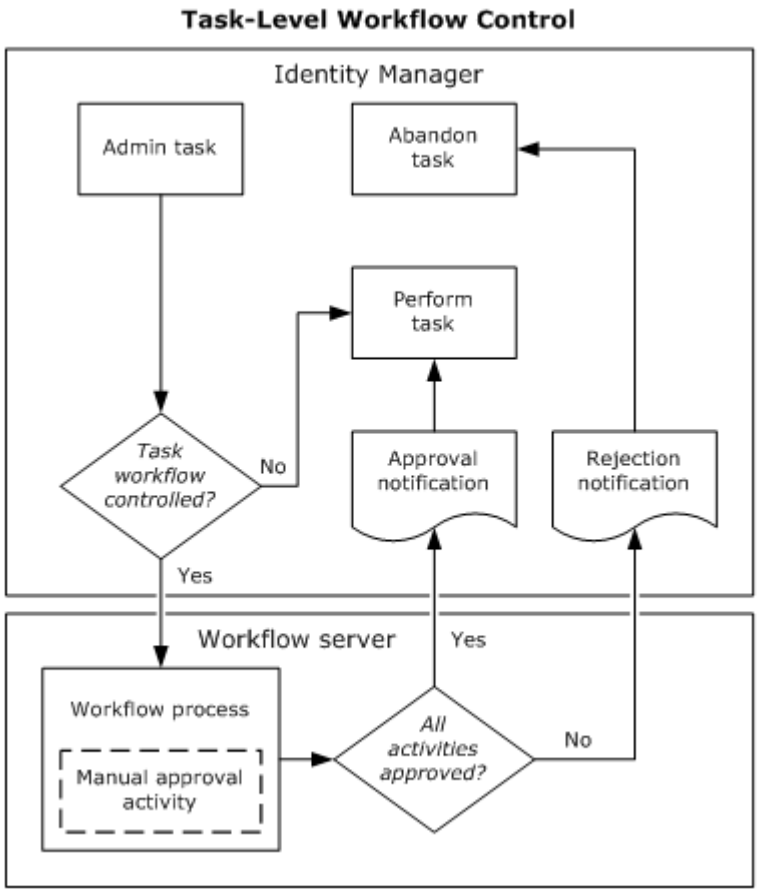
- SingleStepApproval
- TwoStageApprovalProcess
- EscalationApproval

When you select an admin task for task-level workflow, only these process templates are available.

Note: Although TASK_LEVEL is set to true, the process templates can still be used for event-level workflow. Do not change this TASK_LEVEL attribute value.

Task-Level Control Diagram

The following diagram illustrates the interaction between Identity Management and the workflow server when a typical task-level workflow process is initiated:



More Information:

Event-Level Control Diagram (see page 284)

How to Configure Task-Level Workflow

Task-level workflow occurs before any task activity is processed. No events or nested tasks execute before the workflow process job begins.

To configure non-policy based task-level workflow

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify (or Create) Admin Tasks.

   A Select Admin Task screen appears.

2. Search for the task you want under workflow control, and click Select.

   A Modify (or Create) Admin Task screen appears.

3. On the Profile tab, verify that Enable Workflow is checked.

4. On the Profile tab, click the Workflow Process button.

   The Task Level Workflow Configuration tab appears.

5. Select one of the following process templates from the Workflow Process list:

   - SingleStepApprovalProcess
   - TwoStageApprovalProcess
   - EscalationApprovalProcess

   The Task Level Workflow Configuration tab expands.

6. Configure participant resolvers as required by the process template.

   The participant requests are added to the process.

7. Click OK.

   Identity Management saves your task-level workflow configuration.

8. Click Submit.

   Identity Management processes the task modification.

Note: To configure policy based task-level workflow see the Policy-Based Workflow (see page 328) section.

Event-Level Workflow

An event can be mapped to a workflow process. When an event that is mapped to a workflow process is triggered, the workflow process begins. The task that triggered the event is placed in a pending state and is considered under workflow control.

A workflow process may require a participant to approve or reject an event or task before the process can be complete. A task that requires manual workflow approval by a participant takes longer to complete than a task not under workflow control.

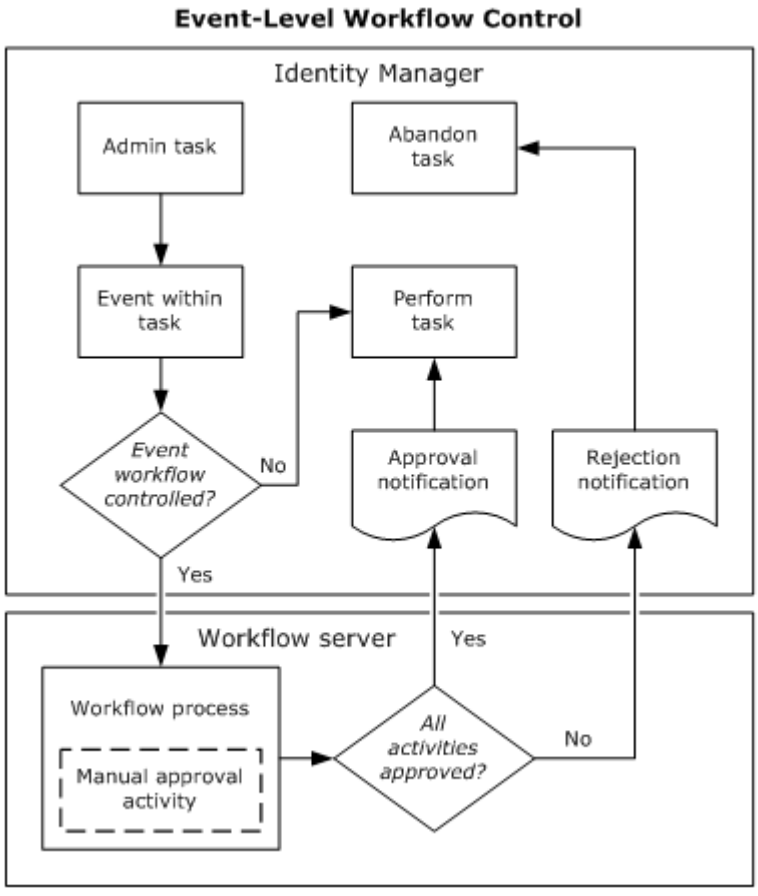
After all activities in a workflow process have been carried out, the event mapped to the workflow process is released from workflow control. When all events triggered by a given task are released from workflow control, the workflow-controlled task is complete.

While the task is under workflow control, the contents of the task screens are saved in the task persistence database. The workflow job state (workflow-relevant data) is stored in the WorkPoint database.

Note: The Events tab lists the events that are generated by each tab in a task. After adding a new tab to a task, you must submit and then reopen the task using Modify Admin Task before the new events are displayed on the Events tab.

Event-Level Control Diagram

The following diagram illustrates the interaction between Identity Management and the workflow server when a typical event-level workflow process is initiated:



More Information:

Task-Level Control Diagram (see page 282)

How to Configure Event-Level Workflow

Event-level workflow begins when an event that is mapped to a workflow process is triggered. The task that triggered the event is placed in a pending state until the participant approves or rejects the task.

To configure non-policy based event-level workflow

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify (or Create) Admin Task.

   A Select Admin Task screen appears.

2. Search for the task you want under workflow control, and click Select.

   A Modify (or Create) Admin Task screen appears.

3. On the Profile tab, verify that Enable Workflow is checked.

4. On the Events tab, select an event to map to a process template.

   The workflow mapping screen appears.

5. Select one of the following process templates from the Workflow Process list:

   - SingleStepApproval
   - TwoStageApprovalProcess
   - EscalationApprovalProcess

   The workflow mapping screen expands.

6. Configure participant resolvers as required by the process template.

   The participant requests are added to the process.

7. Click OK.

   Identity Management saves your event-level workflow configuration.

8. Repeat steps 3 - 6 for each event you want under workflow control.

9. Click Submit.

   Identity Management processes the task modification.

To configure policy-based event level workflow, see the Policy-Based Workflow (see page 328)section.

Note: The Workflow Process list includes processes for use with both the template method and the WorkPoint method:

- When a template method process is selected (either SingleStepApproval, TwoStageApprovalProcess, or EscalationApproval), the page expands to enable participant resolver configuration.
- When a WorkPoint method process is selected, the page does not expand. Participant resolvers are configured in WorkPoint Designer.

Types of Process Templates

A workflow process template has the following characteristics:

- Defined in WorkPoint Designer.
- Has manual activities, which correspond to Identity Management approval tasks.
- Includes special attributes which contain information to identify participants (also called approvers).

Workflow process templates include no information for selecting specific participants. This is provided by Identity Management after a user configures a workflow and its participant resolvers. This information is mapped to an event for event-level workflow control, and to a task for task-level workflow control.

When using the template method, all workflow and participant configuration is done within the User Console.

There are three process templates for use with the template method:

- SingleStepApprovalProcess
- TwoStageApprovalProcess
- EscalationApprovalProcess

How a Process Template Works

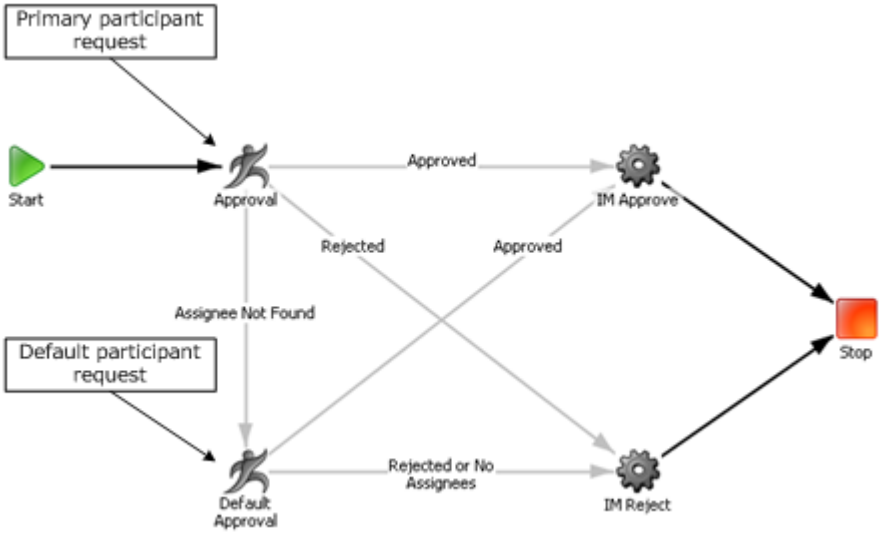
A workflow process template contains a number of places where it requests lists of participants. When the template is mapped to a Identity Management task or event, configure participant resolvers for these lists.

At runtime, as shown in the following figure, Identity Management provides the participant lists to the workflow process based on your configured information:

Single Stage Template Diagram

The following diagram illustrates the SingleStageApproval process template as it appears in WorkPoint Designer. The process template includes two manual activities:

- An approval node for the primary participant. If this user approves or rejects the request, the process runs to completion.
- An approval node for a default participant. This user can approve or reject the request if the primary participant is not found.

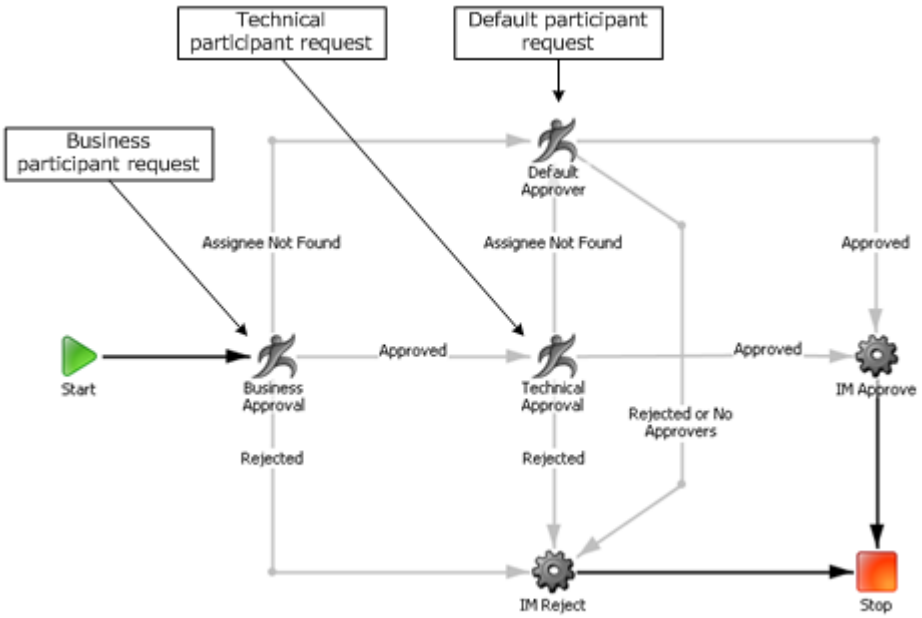


Two Stage Template Diagram

The following diagram illustrates the TwoStageApproval process template as it appears in WorkPoint Designer. The TwoStageApproval process template includes three manual activities:

- An approval node for the business participant. If this user approves the request, the process proceeds to the technical approver and, if this user rejects the request, the process runs to completion.

- An approval node for the technical participant. If this user approves or rejects the request, the process runs to completion.
- An approval node for a default participant. This user can approve or reject the request if either the business or technical participant is not found.

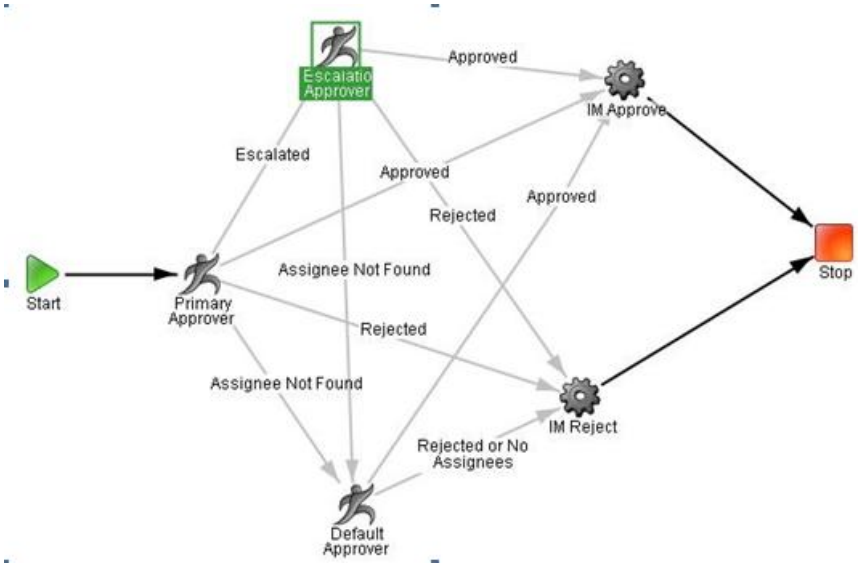


Escalation Approval Template Diagram

The following diagram illustrates the EscalationApproval process template as it appears in WorkPoint Designer. The process template includes the following manual activities:

- An approval node for the primary participant. If this user approves or rejects the request, the process runs to completion.
- An approval node for a default participant. This user can approve or reject the request if the primary participant is not found.

- A timed transition approval node from the primary approver to the escalation approver. This user can approve or reject the request if the primary participant is found but does not respond in the configured time out period.



Note: To add the timeout option to an existing process, add the user data field PARTICIPANT_TIMEOUT to the activity node and add 'Escalated' Transition to the node where you need the work item to be escalated.

Using the Escalation Approval Template

To use the Escalation Approval Template, import the following ZIP file when upgrading from r12.5 to 1.53:

`12.5to12.5SPUpgradeWFScripts.zip`

The ZIP file is located in the following default locations:

- Windows: [set the Installation Path variable]\tools\WorkflowScripts
- UNIX: [set the alternate Installation Path variable]/tools/WorkflowScripts

Types of Participant Resolvers

For the template method, there are seven types of participant resolvers:

Approval Task Role Members

Specifies the participants are members of roles that grant access to the approval task.

User List

Specifies the participants are a specified list of users.

Group Members

Specifies the participants are members of a specified list of groups.

Admin Role Members

Specifies the participants are members of a specified list of admin roles.

Admin Task Members

Specifies the participants are members of admin roles associated with a specified list of admin tasks.

Dynamic Resolver

Specifies the participants are dynamically selected depending on the task or event being approved.

Null Resolver

Resolves to a null list with no users.

Custom

Specifies the participants are determined by a custom participant resolver.

Business Owner Resolver

Specifies the list of participants configured in the Catalog Rule as Business Owners of an Entity.

Admin Owner Resolver

Specifies the list of participants configured in the Catalog Rule as Admin Owners of an Entity.

Manager Resolver

Specifies the participant that is set as Manager of the User Object.

Approval Task Role Members

This resolver assigns the activity to all members of all Identity Management roles that grant access to the approval task. This resolver requires no further configuration.

List of Users

This resolver assigns the work item to a specified list of users.

Scoping is not enforced. Any user may be added to or removed from the list by anyone who has access to the workflow configuration screen.

This resolver has the following validation rules:

- At least one user name must be provided.
- The user names must be those of a currently existing users.

Group Members

This resolver assigns the work item to all members of all groups specified in the group list.

Evaluation of who the group members are is performed at the time the work item is created, not at the time the participant resolver is specified.

Scoping is not enforced. Any group may be added to or removed from the list by anyone who has access to the workflow configuration screen.

This resolver has the following validation rules:

- At least one group must be specified
- The group names must be those of currently existing groups

Admin Role Members

This resolver assigns the work item to all members of the admin roles specified in the admin role list.

Evaluation of who the role members are is performed at the time the work item is created, not at the time the participant resolver is specified.

Scoping is not enforced. Any role may be added to or removed from the list by anyone who has access to the workflow configuration screen.

This resolver has the following validation rules:

- At least one admin role must be specified.
- The admin role names must be those of currently existing admin roles.

Admin Task Members

This resolver assigns the work item to all members of all admin roles associated with the admin tasks specified in the admin task list.

Scoping is not enforced. Any task may be added to or removed from the list by anyone who has access to the workflow configuration screen.

Evaluation of who the role members are and what roles are present on the tasks is performed at the time the work item is created, not at the time the participant resolver is specified.

This resolver has the following validation rules:

- At least one admin task must be specified.
- The admin task names must be those of currently existing admin tasks.

Dynamic Resolver

This resolver returns a list of users according to a dynamic rule resolved at run-time. Use the following selection to set dynamic rule constraints:

Approvers

Specifies the type of user to approve this task.

Note: This only shows those objects that can contain users (or approvers).

User or Object

Specifies the user or object where the approvers can be found.

- Object associated with the event—The event under workflow control.
- Initiator of this task—The user who initiated the admin task.
- Primary object of this task—The object being created/modified by the task.
- Previous approver of this task—The previous approvers of this task.

User Associated with this account

Updates the User or Object Attributes field to list Identity Management user attributes instead of endpoint account attributes. The resolver works off attributes at the Identity Management user level. This check box applies when you select an endpoint account object, such as an Active Directory account.

Attribute

Specifies the attribute that contains the approvers.

Note: The Attribute list is sorted in alphabetical order and contains a list of unique display names. Extended attributes are excluded from the list.

Event Object Type

Specifies the type of object from the event.

Note: This appears only if "Object associated with the event" is selected.

Note: Dynamic Resolver with Create Group requires existence of the object. Group membership/administrators information can be used with dynamic/match attribute resolvers for existing groups only.

The Dynamic Resolver has been enhanced to add the previous approver to the supported object list. If the physical attribute hosting manager information is selected, the configuration routes the approval to a manager.

To configure the resolver for Manager Approval Resolver:

- Set Approvers to Users
- Select "Previous approver of this task" from User or Object drop-down list
- Set the Attribute to the physical attribute containing manager information

Matching Attribute Resolver

This resolver works on objects of type User only. A value from any object available is matched against a field on the user object. Use the following selection to set matching attribute rule constraints:

Approvers

Specifies the type of user to approve this task.

User or Object

Specifies the value that approvers will have in the attribute selected below.

Note: The value retrieved from the user or object should be an acceptable value for a search on user for the selected attribute.

- Object associated with the event—The event under workflow control.
- Initiator of this task—The user who initiated the admin task.
- Primary object of this task—The object being created/modified by the task.(Only available for task level event mapping.)
- Previous approver of this task—The previous approvers of this task.

User Associated with this account

Updates the User or Object Attributes field to list Identity Management user attributes instead of endpoint account attributes. The resolver works off attributes at the Identity Management user level. This checkbox applies when you select an endpoint account object, such as an Active Directory account.

Use or Object Attribute

Specifies the attribute that contains the value to use in the search for approvers.

Approver Search Attribute

Specifies the attribute that is used in the search to match the value identified above.

Note: When you set 'Approve Create User' task as a Match Attribute Resolver that works on Users, Participant Resolver, you must change the method signature for the imApprovers script in Workpoint Designer to point to the unique name for TwoStageProcessDefinition.

Null Resolver

The null resolver returns no users. Depending on how the workflow process is designed, this can cause the process to skip the approval entirely. The null resolver requires no further configuration.

Custom Participant Resolver

The custom participant resolver is a Java object that determines workflow activity participants and returns a list to Identity Management, which then passes the list to the workflow engine. Typically, you write a custom participant resolver only if the standard participant policies cannot provide the list of participants that an activity requires.

Note: You create a custom participant resolver using the Participant Resolver API. For more information, see the *Programming Guide for Java*.

Set an Email Policy for a Workflow Process

You can specify an email policy for each step of the workflow process. Based on the defined email policy, an email is sent when a process reaches a corresponding step or activity. For workflow process-related email notifications, you can only select *When to Send* type *Workflow Pending Email*.

Note: For more information about email policies, see How to Create Email Notification Policies.

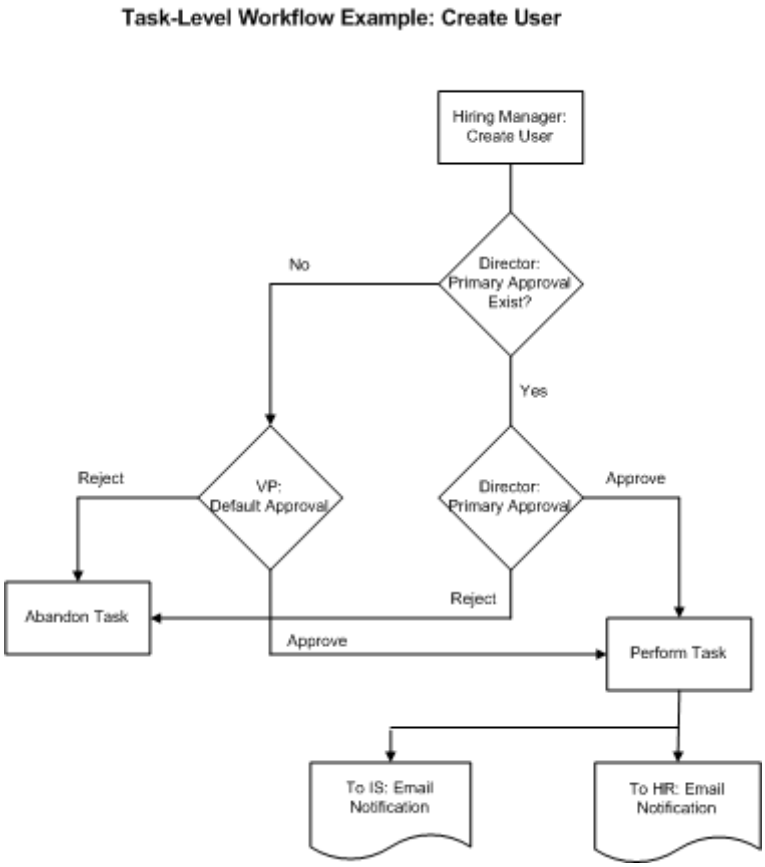
Workflow Example: Create User

A company's Identity Management administrator needs to define a workflow and user roles to handle the following scenario:

- The company Sales Manager hires a new Sales Representative. The Sales Manager must be able to create a Identity Management user for the new hire.
- To streamline the hiring process, the participants want to perform only a single work item to approve (or reject) the task.
- The Sales Director should be primary approver for all new hires. If the Sales Director is not found, the VP of Sales should be the default approver.
- If the new hire is approved, Identity Management should send new user email notifications to both the Human Resources (HR) and Information Services (IS) departments.

Create User Control Diagram

The following diagram illustrates the logic flow for the create user scenario:



Workflow Example Implementation

To implement this example scenario, the administrator needs to perform the following tasks:

- Ensure that the task initiator is a member of the required admin role.

  The Sales Manager needs to be a member of the User Manager admin role. This role gives the Sales Manager the required authority to initiate the Create User admin task for the new hire Sales Representative.

- Enable task-level workflow for the Create User admin task.

  Task-level workflow guarantees that only one work item is generated to complete the Create User task. Because there are several individual events associated with the Create User task, event-level workflow would generate several work items, and also would be harder to configure.

- Configure the participant resolvers.

  The number of possible participant resolvers is determined by the selected workflow process template. The SingleStageApproval template includes primary and default approvers, other templates allow for more.

  Because this scenario requires only two individuals approvers, the User List participant resolver provides the simplest solution. This resolver allows individual approvers to be selected by name, rather than multiple users to be selected by role or group.

- Configure email notification.

  The Management Console allows email notification for specific tasks and events. For this scenario, task email is enabled and email notifications are sent when the Create User task completes.

  A custom email template is required to send email to the HR and IS departments with the appropriate subject line and message text.

More Information

Email Notifications (see page 365)
Place Admin Tasks under Workflow Control - Template Method (see page 279)
How to Configure Task-Level Workflow (see page 283)

How to Use the WorkPoint Method

The WorkPoint method applied to Identity Management releases prior to r12. There are 14 predefined WorkPoint workflow processes that by default are mapped to specific Identity Management events. You must use WorkPoint Designer to configure participant resolvers and otherwise modify workflow processes.

The WorkPoint method also requires you to use the Management Console to map a workflow process to an approval event to place the corresponding task under workflow control at a global level within the environment.

This section lists the high-level steps involved with placing admin tasks under workflow control using the WorkPoint method.

Note: For greater flexibility and ease of use, CA recommends that you use the template method whenever possible.

To use the WorkPoint method

1. Configure Workpoint Administrative Tools. (see page 298)
2. In the Management Console:
   a. Ensure workflow is enabled for your environment by selecting the Enabled check box in Advanced Settings, Workflow.

      Note: The Event Mappings on this screen apply only if you use the WorkPoint method to configure workflow. If you use the template method (recommended), do not map events to processes using this Management Console.

   b. (Optional) For global event mapping, associate one or more events to the appropriate predefined workflow process.
   c. If necessary, restart the Identity Management environment.
3. In the User Console:
   a. For task-specific event mapping, associate one or more events to the appropriate predefined workflow process. (optional)
4. In WorkPoint Designer:
   a. Associate an approval task with a workflow process (optional).
   b. Configure participant resolvers with a workflow process (optional).
5. In the User Console:
   a. After workflow control is configured, the user with the appropriate role performs the admin task.
   b. The designated workflow participant approves or rejects the event.

More Information:

Mapping Processes to Events (see page 304)
Associate a Workflow Activity with an Approval Task (see page 309)
Participant Resolvers: WorkPoint Method (see page 310)

Configure WorkPoint Administrative Tools

WorkPoint Designer is software from Workpoint LLC, a subsidiary of Planet Group, Inc., that is integrated with Identity Management. WorkPoint Designer lets you manage workflow processes and workflow jobs. WorkPoint Administrative Tools include WorkPoint Designer and WorkPoint Archive. In order to configure WorkPoint Administrative Tools, install the Identity Management Administrative Tools. If you have not installed the Identity Management Administrative Tools, you can run the installer and select the Identity Management Administrative Tools option.

Note: To use the Administrative Tools for workflow, a supported JDK must be installed on the system where the Administrative Tools are installed. For a complete list of supported platforms and versions, see the Identity Management Support Matrix on the Identity Management support site.

The workflow client tools are located in the WorkPoint directory in the Identity Management Administrative tools. The Administrative Tools are placed in the following default locations:

- **Windows:** [set the Installation Path variable]\tools
- **UNIX:** [set the alternate Installation Path variable]/tools

The tools in this directory let you do the following:

- Create the workflow database schema
- Load the default workflow scripts
- Design and monitor Workflow processes and jobs

Configure WorkPoint Administrative Tool on JBoss

To configure the WorkPoint Administrative Tools on JBoss, edit the init.bat/sh and the workpoint-client.properties files.

Edit init.bat/init.sh

To edit init.bat/init.sh

1. In a text editor, edit one of the following files:
   - **Windows:**

     admin_tools\Workpoint\bin\init.bat
   - **UNIX:**

     *admin_tools/*Workpoint/bin/init.sh
2. Uncomment the EJB_CLASSPATH line in the JBoss section of the file.

   Note: Be sure that all sections for other application servers are commented.

3. Create a directory named JBoss under *admin_tools*\Workpoint\lib.
4. Copy the content of the jb*oss_home*\client directory to the *admin_tools*\Workpoint\lib\JBoss directory.

Edit workpoint-client.properties

Edit the workpoint-client.properties file based on the type of application server you selected during the Identity Management installation.

To configure the workpoint-client.properties file

1. Open *admin_tools*\Workpoint\conf\ workpoint-client.properties in a text editor.

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

   - **Windows:** [set the Installation Path variable]\tools
   - **UNIX:** [set the alternate Installation Path variable]/tools
2. Locate the section titled JBOSS SETTINGS.
3. Uncomment all of the property values in that section.

   For example:

   ```
   java.naming.provider.url=localhost
   java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
   java.naming.factory.url.pkgs=org.jboss.naming
   ```

   Note: You may need to edit the java.naming.provider.url property value. For example, replace localhost with jnp://*server_name or ip*:*port*. Ensure you use the jnp port number 1099.
4. Save the file.

Configure WorkPoint Administrative Tools on WebLogic

To configure the WorkPoint Administrative Tools on WebLogic, edit the init.bat/sh and the workpoint-client.properties files.

Edit init.bat/init.sh

To edit init.bat/init.sh

1. In a text editor, edit one of the following files:
   - **Windows:**

     admin_tools\Workpoint\bin\init.bat
   - **UNIX:**

     admin_tools/Workpoint/bin/init.sh
2. Uncomment the EJB_CLASSPATH in the WebLogic section of the file:

   Note: Be sure that all sections for other application servers are commented.
3. Copy the wlclient.jar file from *weblogic_home*\server\lib to the following location:

   admin_tools\Workpoint\lib\

Edit workpoint-client.properties

Edit the workpoint-client.properties file based on the type of application server you selected during the Identity Management installation.

To configure the workpoint-client.properties file

1. Open *admin_tools*\Workpoint\conf\ workpoint-client.properties in a text editor.
2. Locate the WebLogic section of the file.
3. Uncomment all property values in that section.
4. Save the file.

   Note: The java.naming.provider.url property must point to the fully-qualified domain name and WebLogic port number of the system on which you installed the Identity Management Server.

Configure WorkPoint Administrative Tools on WebSphere

To configure the WorkPoint Administrative Tools on WebSphere, edit the init.bat/sh and the workpoint-client.properties files.

Edit init.bat/init.sh

To edit init.bat/init.sh

1. In a text editor, edit one of the following files:
   - **Windows:**

     admin_tools\Workpoint\bin\init.bat
   - **UNIX:**

     admin_tools/Workpoint/bin/init.sh
2. Uncomment the IBM WebSphere section.

   Note: Do not comment the WP_CLASSPATH entry in the COMMON WP_CLASSPATH section.
3. Be sure that all sections for other application servers are commented.
4. If necessary, replace the values for JAVA_HOME and WAS_HOME with the appropriate paths for your environment.

Edit workpoint-client.properties

Edit the workpoint-client.properties file based on the type of application server you selected during the Identity Management installation.

To configure the workpoint-client.properties file

1. Open *admin_tools*\Workpoint\conf\ workpoint-client.properties in a text editor.

   admin_tools is the installed location of the Administrative tools. The Administrative Tools are placed in the following default locations:
   - **Windows:** [set the Installation Path variable]\tools
   - **UNIX:** [set the alternate Installation Path variable]/tools
2. Locate the section titled IBM WEBSPHERE SETTINGS.
3. Uncomment all of the property values in that section.

   For example:

   ```
   java.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFactory
   java.naming.provider.url=iiop://localhost:bootstrap_port
   ```

   Note: The bootstrap port number must match the port number specified in the WebSphere Administrative Console. To locate the correct port number, go to Server, Endpoints, Bootstrap server address.

4. Update BOOTSTRAP_ADDRESS port for the WebSphere profile as follows:
   a. In the WebSphere Administrative Console, navigate to Application Servers, server_name, Communications.
   b. Expand Ports.
   c. Edit the workpoint-client.properties file under iam_im.ear/config.
   d. Change the default port 2809 in the WebSphere section to the profile's port for the BOOTSTRAP_ADDRESS.
5. Save the file.

Starting WorkPoint Designer

To start WorkPoint Designer, run the following file:

- **Windows:** *admin_tools*\WorkPoint\bin\Designer.bat
- **UNIX:** *admin_tools*/WorkPoint/bin/Designer.sh

where *admin_tools* is the installation directory of the Identity Management administrative tools. The Administrative Tools are placed in the following default locations:

- **Windows:** [set the Installation Path variable]\tools
- **UNIX:** [set the alternate Installation Path variable]/tools

Note: You must have workflow components installed and configured before you can run WorkPoint Designer. For instructions, see the "Configure WorkPoint Administrative Tools" section for your application server.

More Information:

Configure WorkPoint Administrative Tool on JBoss (see page 298)
Configure WorkPoint Administrative Tools on WebLogic (see page 299)
Configure WorkPoint Administrative Tools on WebSphere (see page 300)

WorkPoint Processes

Identity Management includes a number of workflow processes that are predefined in WorkPoint Designer. You can use the predefined processes with their default event mappings, map the workflow processes to other events, modify workflow processes by adding or removing activities, and create new workflow processes.

Global Process to Event Mapping

The mapping of a workflow process to an event at a global level can be a non policy-based or policy-based.

For more information on how to map an event to a workflow process using policy-based workflow, see Global Event Policy-Based Workflow Mapping.

This table shows the default global workflow process and event mappings that are specified in the Management Console.

Important! These are global mappings. The mapped workflow process executes whenever the corresponding event is generated by any task in the environment.

Workflow Process	Mapped Event
CertifyRoleApproveProcess	CertifyRoleEvent
CreateGroupApproveProcess	CreateGroupEvent
CreateOrganizationApproveProcess	CreateOrganizationEvent
CreateUserApproveProcess	CreateUserEvent
DeleteGroupApproveProcess	DeleteGroupEvent
DeleteOrganizationApproveProcess	DeleteOrganizationEvent
DeleteUserApproveProcess	DeleteUserEvent
ModifyAccessRoleMembershipApproveProcess	AssignAccessRoleEvent RevokeAccessRoleEvent
ModifyAdminRoleMembershipApproveProcess*	AssignAdminRoleEvent RevokeAdminRoleEvent
ModifyGroupMembershipApproveProcess*	AddToGroupEvent RemoveFromGroupEvent
ModifyOrganizationApproveProcess	ModifyOrganizationEvent
ModifyObjectApproveProcess	ModifyObjectEvent
SelfRegistrationApproveProcess	SelfRegisterUserEvent

Note: Workflow processes marked with an asterisk (*) are not mapped to events by default.

More Information:

Map a Process to an Event Globally (see page 305)
Map a Process to an Event in a Specific Task (see page 306)

Mapping Processes to Events

You create and modify workflow processes in WorkPoint Designer. When you create a workflow process for Identity Management, you do so with a particular Identity Management task in mind. The execution of this task is controlled by the workflow process.

In addition to creating the workflow process, you must also do the following:

- Identify the event that is generated by the Identity Management task, described in Admin Tasks and Events. You can create a workflow process for any Identity Management task that generates an event.
- Map the workflow process to an event by doing one of the following:
  - Assign a Workflow Process to an Event Globally.

    With this global mapping, the workflow process occurs whenever the event is generated in the environment, regardless of the task that generates the event.
  - Assign a Workflow Process to an Event Generated by a Specific Task.

    With this task-specific mapping, the workflow process occurs only when the specified task generates the event.

  Note: If you map an event to a workflow process both globally and to a specific task, the workflow process associated with the specific task takes precedence.
- Specify a participant resolver for the workflow activity in the workflow process.
- Associate a workflow activity with an approval task.

More Information:

Map a Process to an Event Globally (see page 305)
Map a Process to an Event in a Specific Task (see page 306)
Workflow Activities (see page 307)
Participant Resolvers: WorkPoint Method (see page 310)

Map a Process to an Event Globally

You map a workflow process to an event globally so the workflow process executes when the event is generated by any task in the environment.

Note: Although the following procedure works, the Global Event Level Policy-Based (see page 343) procedure is the now recommended method of mapping a process to an event.

To map a non-policy based workflow process to an event globally

1. Open the Management Console by entering the following URL in a browser:

   `http://hostname/iam/immanage`

   hostname

   Defines the fully qualified domain name of the server where Identity Management is installed. For example, myserver.mycompany.com:port.

2. Click Environments, and select the name of the appropriate Identity Management environment.
3. Click Advanced Settings, and then click Workflow.
4. Do the following to map an event to a workflow process:
   a. Select an event from the Event list box.
   b. Select a workflow process from the Approve Process list box.
   c. Click Add.
5. After you finish mapping events to workflow processes, click Save.
6. Restart the Identity Management environment for changes to take effect.

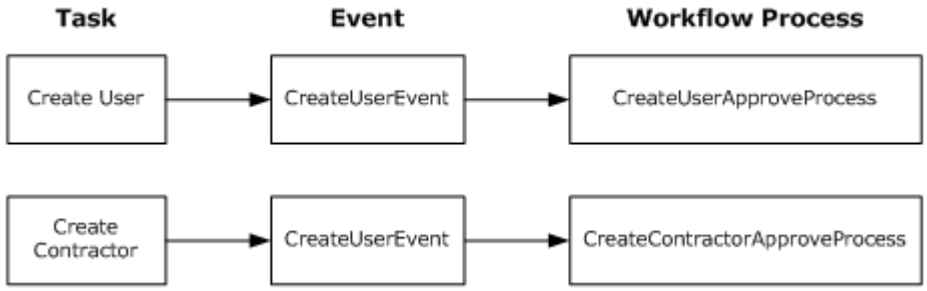
More Information:

Global Process to Event Mapping (see page 303)

Map a Process to an Event in a Specific Task

You can assign a workflow process to an event that is generated by a particular task. In this case, the workflow process occurs only when the mapped event is generated by the specified task.

Task-specific mapping provides variable control over the workflow processes that can be executed for the same event. For example, the following diagram shows two different tasks generating the same event but triggering two different workflow processes:



In this diagram, each task uses a different workflow process.

Create User

Specifies the default admin task that triggers CreateUserEvent, which is mapped to CreateUserApproveProcess, a default workflow process.

Create Contractor

Specifies a custom task based on Create User. In this case, CreateUserEvent is mapped to CreateContractorApproveProcess, a custom workflow process created for approving new contractor accounts.

To map a non-policy based workflow process to an event in an existing task

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for an administrator task.
3. Select a task (for example, the Modify User or Create User tasks) and click Select.
4. On the Events tab, select a workflow process for the event in the task.

   Note: Workflow must be enabled for the event names and the workflow process drop-down menu to appear on this tab.
5. Click the Edit button to view the Workflow mapping screen.
6. Using the Workflow Process drop-down menu, assign a workflow process to the event name and click OK.
7. Click Submit.

To map a non-policy based workflow process to an event in a new task

1. In the User Console, select Roles and Tasks, Admin Tasks, Create Admin Task.

   Note: Be sure you select an existing workflow approval task (such as Approve Create Group or Approve Create User) as the template for your new workflow approval task.

2. On the Profile tab, enter the information in the appropriate fields.

3. On the Events tab, select a workflow process for the event in the task.

   Note: Workflow must be enabled for the event names and the workflow process drop-down menu to appear on this tab.

4. Using the Workflow Process drop-down menu, assign a workflow process to the event name and click OK.

5. Click Submit.

Note: To map a policy-based workflow process to an event, see the Policy-Based Workflow (see page 328) section.

Note: The Workflow Process list includes processes for use with both the template method and the WorkPoint method:

- When a template method process is selected (either SingleStepApproval, TwoStageApprovalProcess, or EscalationApproval), the page expands to enable participant resolver configuration.
- When a WorkPoint method process is selected, the page does not expand. Participant resolvers are configured in WorkPoint Designer.

More Information:

Global Process to Event Mapping (see page 303)

Workflow Activities

Identity Management includes a number of workflow activities that are predefined in the WorkPoint Designer. These activities are assigned to predefined workflow processes.

The predefined workflow processes are single-step processes—that is, each process contains a single predefined activity.

Each predefined activity corresponds to a workflow approval task with the same name that is predefined in Identity Management. You can use the predefined activities in other workflow processes, and you can create new activities.

You can use the predefined workflow processes without modification or add more activities to them. For information about adding an activity to a workflow process, see the WorkPoint documentation.

Processes, Tasks, and Activities

The table below lists the predefined workflow activities and the predefined workflow process that each activity is assigned to by default.

Note: The predefined workflow activities and their corresponding workflow approval tasks have the same name.

Workflow Process	Workflow Task/Activity
CertifyRoleApprovalProcess**	Approve Certify Role
Consultation Process*	
CreateGroupApproveProcess	Approve Create Group
CreateOrganizationApproveProcess	Approve Create Organization
CreateUserApproveProcess	Approve Create User
DeleteGroupApproveProcess	Approve Delete Group
DeleteOrganizationApproveProcess	Approve Delete Organization
DeleteUserApproveProcess	Approve Delete User
ModifyAccessRoleMembershipApproveProcess	Approve Modify Access Role Membership
ModifyAdminRoleMembershipApproveProcess	Approve Modify Admin Role Membership
ModifyGroupMembershipApproveProcess	Approve Modify Group Membership
ModifyIdentityPolicySetApproveProcess	Approve Modify Identity Policy Set
ModifyOrganizationApproveProcess	Approve Modify Organization
ModifyUserApproveProcess	Approve Modify User
SelfRegistrationApproveProcess	Approve Self Registration
SingleStepApproval*	
TwoStageApprovalProcess*	

Note: Workflow processes marked with one asterisk (*) are designed for use with the template method. They are configured in the User Console, and therefore have no default associated tasks or activities. The CertifyRoleApprovalProcess (**) is a sample process demonstrating a custom participant resolver.

Associate a Workflow Activity with an Approval Task

To associate a workflow activity with a workflow approval task, you define a name/value pair in WorkPoint Designer.

Note: If a name/value pair is not defined for a workflow activity by default, Identity Management uses a task with a name that matches the approval task.

To associate a workflow activity with an approval task

1. Start WorkPoint Designer.
2. Click File, Open, Process.
3. Select a workflow process and click Open.
4. Right click the activity node in the process, and select Properties.
5. Select Text from the Type drop-down menu.
6. Enter the following in the User Data tab:
   - **Name**—TASK_TAG.
   - **Value**—Approval Task Tag name.
7. Click Add.
8. Click OK to save your changes.

Create Approval Tasks for Endpoints

You can create Approval Tasks for account management screens. For tasks that approve account modifications, the approval screen must be specific to an endpoint type, so that the approver can see the changed values. To create an approval task for a Create or modify task, follow this procedure:

To create an approval task for an endpoint

1. In the User Console, click Roles and Tasks, Admin Tasks, Create Admin Task.
2. Select "Create a copy of an admin task" used to manage accounts on the endpoint.

   The name would start with create and state the name of the endpoint type. Create Active Directory Account is an example.
3. Make the following changes on the Profile tab.
4. Change the name of the new task.
   - Change the task tag.
   - Change the action to Approve Event.
5. Make the following changes on the Tabs tab:
   a. Remove all Relationships tabs.
   b. Copy and then edit the approval screens on the tabs as necessary.

      Note: You may run into problems when using the account screens in an approval task and changes may need to be made to the default account screen to make them work in an approval task.
6. Click Submit.

Participant Resolvers: WorkPoint Method

To specify participants using the WorkPoint method, define the following activity properties in WorkPoint Designer:

- The name of the predefined Identity Management script which enables communication between Identity Management and the workflow server. The script issues a request to Identity Management for activity participants, and supplies that list to the workflow server.
- References to one or more participant resolvers.

Types of Participant Resolvers

Rather than entering a specific list of participants in workflow activity properties, the participants are referenced by an arbitrary name that is mapped to a *participant resolver*.

For the predefined process model, there are four types of participant resolvers:

Role Participant Resolver

Specifies the participants are members of a particular role.

Group Participant Resolver

Specifies the participants are members of a particular group.

Custom Participant Resolver

Specifies the participants are determined by a custom participant resolver.

Filter Participant Resolver

Specifies the participants are selected through a search filter.

Role Participant Resolvers

With role type participant resolvers, Identity Management retrieves all of the members for that role and returns those members as participants.

If no resolver type is specified in the UserData parameter of the Activity dialog box, the role type resolver is used by default.

If you do not specify any participant resolvers in the User Data tab of the WorkPoint Activity Properties dialog box, by default, Identity Management finds all available roles containing this approval task and returns back those role members as participants.

To configure role participant resolvers

1. Start WorkPoint Designer.
2. Click File, Open, Process.
3. Select a workflow process and click Open.
4. Right click the activity node in the process, and select Properties.
5. Select Text from the Type drop-down menu.
6. Enter the following in the User Data tab:
   - **Name**—APPROVER_ROLE_NAME
   - **Value**—The name of a Identity Management role (for example, Security Manager)

7. Click Add.

   Note: This role does not need to contain any approval tasks.

8. Select Text from the Type drop-down menu.
9. In the User Data tab, enter the following name/value pair (optional):

   Value—APPROVERS_REQUIRED

   Value—YES.

10. Click Add.

    Note: The default approval setting is APPROVERS_REQUIRED=NO. In this case, an activity is approved automatically if no participants are found.

    If APPROVERS_REQUIRED=YES and Identity Management finds no participants, the activity is not successfully completed.

11. Click OK to save your changes.

Group Participant Resolvers

With group type participant resolvers, Identity Management retrieves all of the members for that group and returns those members as participants.

To configure group participant resolvers

1. Start WorkPoint Designer.
2. Click File, Open, Process.
3. Select a workflow process and click Open.
4. Right click the activity node in the process, and select Properties.
5. Select Text from the Type drop-down menu.
6. Enter the following in the User Data tab:
   - **Name**—APPROVER_GROUP_UNIQUENAME
   - **Value**—The name of a Identity Management group
7. Click Add.
8. Select Text from the Type drop-down menu.
9. In the User Data tab, enter the following name/value pair (optional):
   - **Name**—APPROVERS_REQUIRED
   - **Value**—YES.
10. Click Add.

Note: The default approval setting is APPROVERS_REQUIRED=NO. In this case, an activity is approved automatically if no participants are found.

If APPROVERS_REQUIRED=YES and Identity Management finds no participants, the activity is not successfully completed.

11. Click OK to save your changes.

Custom Participant Resolvers

The custom participant resolver is a Java object that determines workflow activity participants and returns a list to Identity Management, which then passes the list to the workflow engine. Typically, you write a custom participant resolver only if the standard participant policies cannot provide the list of participants that an activity requires.

Note: You create a custom participant resolver using the Participant Resolver API. For information, see the *Programming Guide for Java*.

To configure a custom participant resolver

1. Open the Management Console by entering the following URL in a browser:

   `http://hostname/iam/immanage`

   hostname

   Defines the fully qualified domain name of the server where Identity Management is installed. For example, myserver.mycompany.com:port.

2. Click Environments, and select the name of the appropriate Identity Management environment.
3. Click Advanced Settings, Workflow Participant Resolver.
4. On the WorkFlow Participant Resolver screen, click New and enter:

   Name

   Specifies the custom participant resolver name, for example, GroupFinder.

   Description

   Specifies a description of the custom participant resolver.

   Class

   Specifies the Java class name, for example, com.netegrity.samples.GroupFinder

5. Click Save.
6. Start WorkPoint Designer.
7. Click File, Open, Process.
8. Select a workflow process and click Open.
9. Right click the activity node in the process, and select Properties.

10. Select Text from the Type drop-down menu.
11. Enter the following in the User Data tab:

    Name

    APPROVER_CUSTOMRESOLVER_NAME

    Value

    Specifies a unique name for the custom resolver. This must match the name you entered on the Custom Type Participant Resolver screen in the Management Console, for example, GroupFinder.
12. Click Add.

    Note: The default approval setting is APPROVERS_REQUIRED=NO. In this case, an activity is approved automatically if no participants are found.

    If APPROVERS_REQUIRED=YES and Identity Management finds no participants, the activity is not successfully completed.
13. Click OK to save your changes.

Filter Participant Resolvers

A filter participant resolver enables Identity Management to search for users or groups that match the filter criteria. You specify a search filter in WorkPoint Designer, and Identity Management returns matching approvers for the corresponding workflow activity.

You create a filter participant resolver on the User Data tab of the WorkPoint Activity Properties dialog box.

Participant Resolvers Filter Syntax

The following are three required attributes that combine to make a search filter:

- Approver attribute, such as title
- Approver attribute operation, such as equals
- Approver attribute value, such as manager

The required search filter attributes combine together in the following order:

attribute operation value

For example:

title equals manager or *department contains payroll*

Required Participant Resolver Filter Attributes

The following are *required* participant resolver filter attributes:

Note: For each filter, n is a positive integer indicating the search filter number. The default is 1.

APPROVER_FILTER_n_ATTRIBUTE

Specifies the approver attribute. For example, Title, Department, User ID. (Approver attribute name strings must match Identity Management user attribute name strings.)

APPROVER_FILTER_n_OP

Specifies the operation associated with the approver attribute. For example, equals, not_equals, or contains. (Operation keywords are not case-sensitive.)

The following are valid entries for this filter:

- EQUALS
- STARTSWITH
- NOT_EQUALS
- CONTAINS
- ENDS_WITH
- GREATER_THAN
- LESS_THAN
- GREATER_THAN_EQUALS
- LESS_THAN_EQUALS

APPROVER_FILTER_n_VALUE

Specifies the value associated with the approver. For example, manager, payroll, engineering.

Optional Participant Resolver Filter Attributes

The following are *optional* participant resolver filter attributes.

APPROVER_OBJECTTYPE

USER or GROUP (not case-sensitive)

The default is USER.

APPROVER_ORG_UNIQUENAME

A unique name for an approver's organization. (Organization name strings must match Identity Management organization name strings.)

The default is root.

APPROVER_ORG_AND_LOWER

The approver's organization or sub-organizations:

- 0 means search in the approver's organization.
- 1 means search in all sub-organizations of the approver's organization.

The default is 1.

APPROVER_FILTER_NO

The number of search filter that you are using. If you have two filters, then this number would be 2.

The default is 1.

Note: This filter is required if the number of filters is greater than one.

APPROVER_FILTER_n_CONJ_TYPE

You can combine search filters using OR or AND conjunction types.

Note: Filters separated by the OR conjunction take precedence over those separated by AND.

For example, you can specify the AND conjunction type if you are searching for "title equals manager" AND "department equals development."

Note: n is a positive integer greater than 1 indicating the search filter number.

Add a Participant Resolver Filter

To add participant resolver filters

1. Start WorkPoint Designer.
2. Click File, Open, Process.
3. Select a workflow process and click Open.
4. Right click the activity node in the process, and select Properties.
5. Select Text from the Type drop-down menu.
6. Enter the following in the User Data tab:
   - **Name**—APPROVER_FILTER_1_ATTRIBUTE
   - **Value**—A unique role identifier (for example, title).
7. Click Add.
8. Repeat steps 6 and 7 for each attribute in the search filter.

Note: The default approval setting is APPROVERS_REQUIRED=NO. In this case, an activity is approved automatically if no participants are found.

If APPROVERS_REQUIRED=YES and Identity Management finds no participants, the activity is not successfully completed.

9. Click OK to save your changes.

Example: Filter Participant Resolver

The user store in the following table contains four users—Holly, Sarah, John, and Dave—with user ID, job title, and department attributes.

User	ID	Title	Department
Holly	admin1	sysadmin	administration
Sarah	test1	sysadmin	development
John	admin2	manager	development
Dave	admin3	sysadmin	accounting

Identity Management applies the three filters defined in the following table against the preceding user store:

Name	Value
APPROVER_FILTER_NO	3
APPROVER_FILTER_1_ATTRIBUTE	uid
APPROVER_FILTER_1_OP	equals
APPROVER_FILTER_1_VALUE	admin*
APPROVER_FILTER_2_CONJ_TYPE	AND
APPROVER_FILTER_2_ATTRIBUTE	department
APPROVER_FILTER_2_OP	equals
APPROVER_FILTER_2_VALUE	administration
APPROVER_FILTER_3_CONJ_TYPE	OR
APPROVER_FILTER_3_ATTRIBUTE	title
APPROVER_FILTER_3_OP	equals
APPROVER_FILTER_3_VALUE	sysadmin

Identity Management applies the filters in the following sequence:

1. Evaluates the second and third filters connected by the OR conjunction.

   "department equals administration" OR "title equals sysadmin"

   This excludes John and returns Holly, Sarah, and Dave.

2. Evaluates the first and second filters connected by the AND conjunction, (where * is a wild card character.

   "uid equals admin*" AND "department equals administration"

   This excludes Sarah, and returns Holly and Dave.

The final users returned from the user store are Holly and Dave.

Participant Resolver Order of Precedence

If you do not specify any participant resolvers, by default Identity Management identifies all available roles containing the approval task and returns those role members as participants.

If you specify more that one participant resolver, Identity Management evaluates them using this order of precedence:

1. Custom
2. Role
3. Filter
4. Group

Identity Management identifies and applies the first resolver in this order of precedence, and ignores any subsequent remaining resolvers.

You should only have one resolver at a time. Also, make sure that the resolver is configured properly so that Identity Management correctly identifies participants.

Specify Workflow Resource Script

Identity Management is shipped with a script, named IM Approvers, that passes information between Identity Management and the workflow server.

When a list of participants is required for a workflow activity, the script passes to Identity Management the activity name, the participant identifier provided on the User Data tab of the WorkPoint Activity Properties dialog box, and any other information provided on the User Data tab. Identity Management searches for the participants and passes the list back to the script. The script then provides the list to the workflow server.

When you have a new workflow process definition and the workflow process activity is a Identity Management workflow approval task, the IM Approvers script must be specified in the Resources tab of the WorkPoint Activity Properties dialog box.

To specify the IM Approvers script in WorkPoint Designer

1. In the Resources tab, click Select.
2. In the Select Resources dialog box, select Rule from the drop-down list. This action lists the rules (scripts) that you can associate with the activity.
3. Select the script name IM Approvers and click Add.
4. Click OK, and then click Apply on the Activity Properties dialog box.

Note: Do not modify the IM Approvers script.

Specify Participants for Certify User Tasks

Certify User tasks generate the event CertifyRoleEvent. This event can be subject to workflow approval through the predefined process CertifyRoleApproveProcess.

Identity Management also includes the predefined participant resolver CertifyRoleParticipantResolver, which appears in your environment by default. Participants for activities in a CertifyRoleApprovalProcess are specified through CertifyRoleParticipantResolver.

To provide participant configuration information

1. Open the Management Console by entering the following URL in a browser:

   `http://hostname/iam/immanage`

   hostname

   Defines the fully qualified domain name of the server where Identity Management is installed. For example, myserver.mycompany.com:port.
2. Click Environments, and select the name of the appropriate Identity Management environment.
3. Click Advanced Settings and then click Miscellaneous.
4. Define name/value pairs that specify the approvers for each role to be certified:
   - In the Property field, use the format: *role-type.role-name*

     role-type must be one of these roles: admin, access, provisioning.

     role-name is the name of any existing role.

     The role-name and role-type must be separated by a period (.).
   - In the Value field, specify the IDs of the approvers, and separate the IDs with a semi-colon (;).

In the following example, user certification can be approved for the following roles and by the following participants:

- jsmith01 and ajones19 can approve certification for the User Manager role
- plewis12 is the only approver for the System Manager role
- rtrevor8 and pkitt3 can approve certification for My Access Role

Property	Value
admin.User Manager	jsmith01;ajones19
admin.System Manager	plewis12
access.My Access Role	rtrevor8;pkitt3

Note: Any unspecified roles will not have approvers for a CertifyRoleEvent.

Processes in WorkPoint Designer

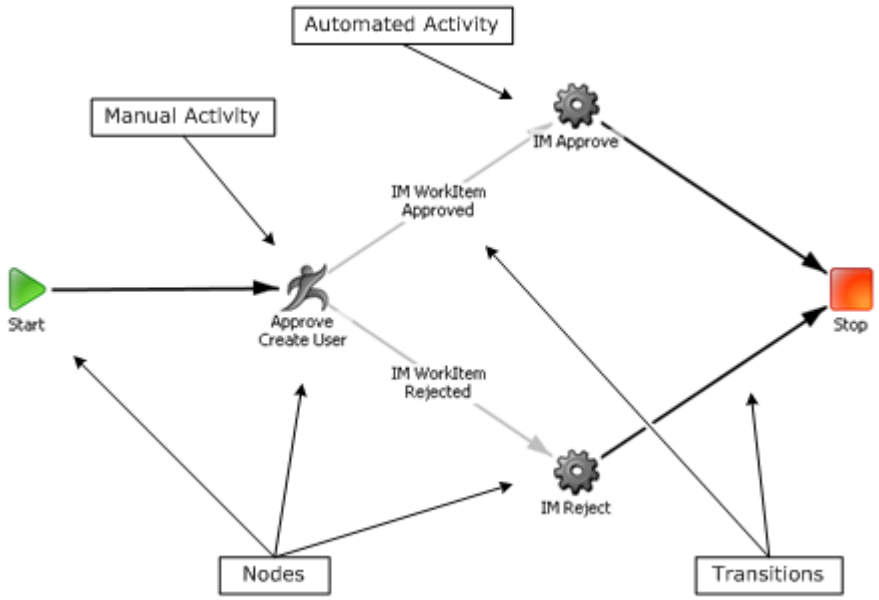
In WorkPoint Designer, you can customize the default workflow processes and activities that come with Identity Management, and you can create new ones.

This document presents WorkPoint workflow information that is specific to Identity Management. For complete information, see the WorkPoint Designer documentation.

Note: When creating a workflow process, consider doing so by making a copy of an existing Identity Management process, and then modifying the new process to suit your needs. A workflow process created in this way includes default Identity Management-specific elements and nodes such as transition scripts and automated activities.

WorkPoint Process Diagram

The following diagram shows a typical workflow process with the minimum set of components for a process that controls a Identity Management task. The diagram illustrates the predefined process CreateUserApproveProcess, which controls the execution of a Create User task.



WorkPoint Process Components

The workflow process contains the following nodes and transitions:

Start

Every workflow process begins with this node.

Stop

Every workflow process ends with this node.

Manual activity

A manual activity requires the approval or rejection of a Identity Management task by a participant, and must have the same name as a Identity Management workflow approval task.

A workflow process that controls a Identity Management task must include at least one manual activity requesting approval for that task.

Automated activity

An automated activity is assigned one of two scripts:

- Notify IM Approve—Informs Identity Management to execute the Identity Management task under workflow control.
- Notify IM Reject—Informs Identity Management to cancel execution of the Identity Management task.

In general, the Notify IM Approve script is activated if all manual activities are approved, and the Notify IM Reject script is activated if any manual activity is rejected.

Unconditional transition

An unconditional transition is a path from one node in the workflow process to another, and is not associated with a condition script.

Conditional transition

A conditional transition represents an alternative path from one node in the workflow process to another, and is associated with a condition script.

A condition script determines whether the transition occurs by evaluating the outcome of the associated activity. If the script returns true, the transition is performed and the process moves to the next indicated node.

It is possible to have two or more condition scripts return true. This allows an activity to be performed in parallel, since each script is associated with a different transition.

Note: You can use custom scripts in conditional transitions. For instructions, see the *Programming Guide for Java*.

Manual Activity Properties

Property settings specific to Identity Management are listed in the following table. These settings are defined in the specified tabs of the WorkPoint Designer Activity Properties dialog box.

Property Tab	Property Descriptions
Resources	IM Approvers—Specified in the Include list. This script passes information between Identity Management and the workflow server.
Agents	Nobody Auto Complete—Specified in the Asynchronous list and associated with the Available state. This script determines whether an activity should be considered approved if no activity participants exist.

Property Tab	Property Descriptions
User Data	Define name/value pairs that Identity Management uses to retrieve activity participants. Optionally, you can also define data to be passed to a custom participant resolver.

Conditional Transition Properties

The following default scripts appear in the Condition tab of the Transition Properties dialog box:

IM WorkItem Approved

Returns true if the associated activity is approved. The workflow process flows to the next node indicated by the transition.

IM WorkItem Rejected

Returns true if the associated activity is rejected. The workflow process flows to the next node indicated by the transition.

Jobs and Process Instances

A *workflow process* defines the steps that must take place before Identity Management can complete a particular task. A *job* is a runtime instance of a workflow process.

For example, the default workflow process CreateUserApproveProcess defines the steps that must occur for a new user to be approved. When a new user is actually created in Identity Management and the task is submitted for approval, a job instance of CreateUserApproveProcess is created in WorkPoint Designer.

You can open, view, and modify jobs in WorkPoint Designer using an interface that is very similar to that used for editing workflow processes.

Multiple jobs based on the same process can exist simultaneously.

Filtering Jobs

WorkPoint Designer includes filtering, which lets you search for jobs based on various criteria. For example, you can search for jobs that:

- Are based on one or more selected workflow processes
- Have a user-defined job reference or a unique job ID
- Are in a particular state (such as active, complete, or suspended)
- Were created or were started within a specified date range

Note: For instructions and reference information about job filtering, see the WorkPoint Designer documentation.

Job Status and Properties

When you open a job, the job's workflow diagram is displayed. Workflow activity nodes and transitions are rendered in color indicating whether that have been performed.

You can view, and in some cases modify:

- Properties of a job, including participant and job history information
- The state of an open job, for example whether it has completed
- Properties of individual nodes and transitions in a job

Activity and Work Item Properties

You can view, and in some cases, modify job activity properties and process activity properties, including the following:

- Activity state information
- Activity approval information
- Approval task (called *work item* in WorkPoint Designer) information, for example:
  - If no participant has the work item reserved (which removes the item from the work lists of other approvers), the state is Available, and no participant user ID is displayed.
  - If a participant has reserved but not yet completed the work item, the state is Open, and the participant's user ID and the reservation time are displayed.
  - If the work item has been completed, the state is Complete. The user ID of the participant who approved or rejected the task under workflow control is displayed, along with the time of completion.

Specific work item properties include:

- The work item name and current state
- State history information, including the user IDs of the participants responsible for given states
- Authorized work item participants information

Note: For more information about job, activity, and work item properties, see the WorkPoint Designer documentation.

Performing Workflow Activities

In a workflow process, a manual activity is performed by a person designated as an activity participant, who approves or rejects an event associated with an approval task. Participants perform this activity in Identity Management.

The following operations take place when an activity associated with a Identity Management approval task is performed:

1. Identity Management notifies the participants.
2. A participant approves or rejects the task.
3. The workflow server completes the activity.

Find and Notify the Participants

When a workflow activity associated with a Identity Management approval task begins, the workflow server passes information about activity participants to Identity Management. This information is defined in the activity properties. Identity Management uses this information to retrieve activity participants and alert them that an approval task is pending.

After identifying the participants, Identity Management adds a new work item (the approval task) to each participant's work list. Optionally, Identity Management also sends an email notification about the new work item to each participant.

Note: If the APPROVERS_REQUIRED activity property is set to false and no participants are found, the task is considered approved by default.

Note: A circle in the Status column indicates that the approval task is available for any participant to claim. A check mark indicates that the work list owner has accepted the approval task but has not yet completed it.

Accept and Perform the Approval Task

Once participants are found, the activity cannot be completed until one participant accepts the approval task and either approves or rejects the task under workflow control.

A participant accepts an approval task by clicking the work item name in the Workflow Activity Console, and then clicking Reserve Item. (Reserving an item removes it from the work lists of other approvers.)

Once a participant accepts an approval task, he commits to making the approval or rejection decision for the task under workflow control. And, since multiple participants cannot accept the same approval task, the approval task is removed from the work lists of other participants.

After a participant accepts an approval task, an approval screen appears, in which the participant can take one of these actions:

- Approve or reject the task under workflow control immediately.
- Release the approval task to make it available to other participants.
- Close the dialog box and complete the activity later. To reopen the Approve Create User dialog box shown above, the participant clicks the name of the approval task in his work list.

Additionally, the participant can update one or more modifiable fields, if any, on the approval screen. You can make fields on this screen modifiable when you create the task.

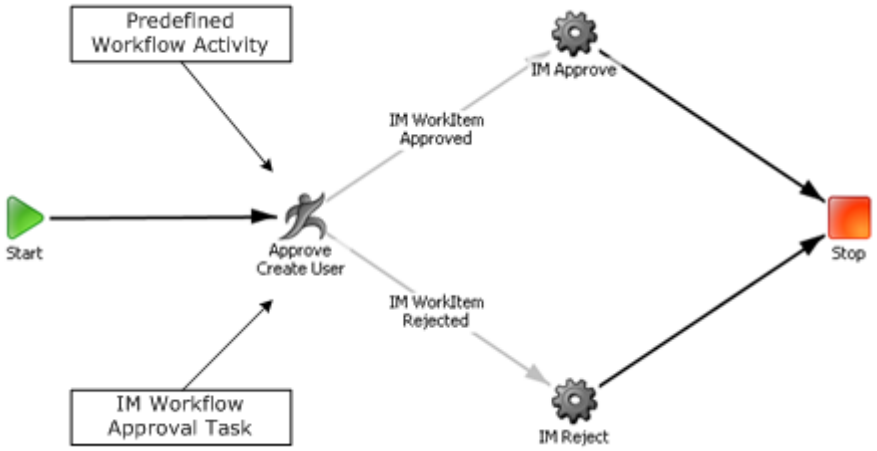
After the participant approves or rejects the task under workflow control, the activity is complete, and the workflow process can continue along the path determined by the outcome of the activity, as described in the next section.

Workflow Server Completes the Activity

A manual activity appears in the Designer window with two or more conditional transitions leading from it.

Each conditional transition is associated with a script. When a participant completes the activity, the scripts evaluate the activity outcome. The result of these evaluations determines the direction of the process flow.

The following illustration shows the Approve Create User activity in the Designer and the corresponding approval task of the same name in Identity Management.



When the activity participant (or approver) clicks the Approve or Reject button in Identity Management:

1. The Approve Create User activity in the process job instance ends. The scripts associated with the conditional transitions evaluate the outcome of the activity.
2. The job instance continues, depending on which conditional transition evaluates to true:
   - If the activity is approved, script IM WorkItem Approved returns true. The workflow takes the IM WorkItem Approved transition to the next node. This automated activity, IM Approve, notifies Identity Management to execute the Create User task.
   - If the activity is rejected, script IM WorkItem Rejected returns true. The workflow takes the IM WorkItem Rejected transition to the next workflow node. This automated activity, IM Reject, notifies Identity Management to cancel the Create User task.

Workpoint Job View

You can view the runtime status of Workpoint jobs in the User Console from the following:

- Approval tasks
- View Submitted Tasks.

In new environments, all approval tasks include the View Job tab by default. Only events created in this release support viewing the job images for all process definitions invoked for the selected event or task in View Submitted Tasks. Events created in earlier releases do not support the Workflow Job View feature.

Add the View Job Tab to Existing Approval Tabs

For Approval Tasks you must add the new View Job Tab to all existing tasks in order to view the job image for that work item.

Note: New environments contain this tab for all approval tasks.

To add the Job View tab to an existing task

1. From the Admin Tasks and Role category, execute the ModifyAdminTask by selecting Admin Task, Modify Admin Task.
2. Click Search and select an approval task (for example, Approve Create User), and click Select.

   The Modify Admin Task: Approve Create User dialog appears.
3. Click the Tabs tab and from the drop-down menu, select View Job (JobView) and click Submit.

   The View Job Tab has been added to the approval task.

   Repeat for all existing approval tasks.

Policy-Based Workflow

Policy-based workflow allows you to place an event or an admin task under workflow control based on the evaluation of a rule. This means that instead of an event or an admin task always launching a workflow process, the workflow process runs and generates a work item only if a rule associated with the event or an admin task is true.

An *approval rule* is a condition that determines whether to start a workflow process. If started, the workflow process places the event or an admin task under workflow control by adding a work item to an approver's work list.

An *approval policy* is the combination of the approval rule, the rule evaluation type, policy order, policy description, and the workflow process.

For example, when creating a new group, you can define an approval policy that places the CreateGroupEvent under workflow control and creates a work item only if the new group is part of a designated parent organization. If the new group is not part of that organization, the workflow process does not run and no work item is created.

If an event has multiple rules, then all workflow process associated with the event need to be approved in order for the event to be approved. Similarly for an admin task, you can define an approval policy that places the CreateGroupTask under workflow control and creates a work item only if the name of the new group starts with Sales. If the name of the new group does not start with Sales, the workflow process does not run and no work item is created.

You can create a policy rule that is always evaluated or only when a specified attribute of a managed object changes, for example, when an employee's salary changes value.

Note: In earlier versions of policy-based workflow, if any approver made any change to the attributes, they were sent for re-approval. With attribute level approve and reject, changes at any stage are approved only once. The work item is never submitted for re-approval even if the attribute contained in the rule is modified. Once an approver approves a change, they will not see the work item again until a new change is submitted or the task is resubmitted.

More Information:

Event-Level Workflow (see page 283)
Task-Level Workflow (see page 280)
Policy Order (see page 332)
Rule Evaluation (see page 331)

Objects of Rules

A Identity Management administrator can create approval policies for an event or admin task based on the following objects. The following are the objects for an event if they apply to a given event and are present during event execution:

- **Initiator of the task** – The Identity Management administrator who executes the task.
- **Primary object of the event** – The primary object associated with the event.
- **Secondary object of the event** – The secondary object associated with the event relative to the primary object.

The following are the objects for an admin task:

- **Primary Object of the Task** – The primary object associated with the task
- **Initiator of the Task** - The Identity Management administrator who executes the task.
- **Identity Policy Violations** - For identity policy violations, the rules are based on the policy name of the identity policy that caused the violation, for example, Policy Name EQUALS TitlePolicy. The violation message is displayed on the Task Details tab of the Approval Screen which is the same as the View Submitted Tasks Task Details. The SOD violation message is displayed under a new section heading named Identity Policy Violation. An approver can view these messages and decide to approve or reject the task.

  Note: If a rule is based on Identity Policy Violation, the evaluation is different from normal evaluation. An SOD violation once approved does not invoke any other workflow process even if there are other rules that may evaluate to true for that particular SOD violation. With normal evaluation, all workflow processes one-by one even if the same change while the normal evaluation is, it will invoke all the workflow processes one-by-one even if the same change has been approved by other approvers.

Rule Evaluation

Policy rules can be evaluated for an event in the following two ways:

- Always

  A policy with evaluation type of Always gets invoked if the policy evaluates to True irrespective of whether any attributes contained in the policy are changed or not. On the approval screen for a work item that was generated as a result of a policy evaluation type of Always, an approver can change any editable attribute on the approval screen.

  Note: If the approver clicks the Reject button, the event is rejected.

  For Always, evaluation type behaviour is the same for tasks and events.

- Only if an attribute specified in the approval condition changes

  A policy with evaluation type of OnChange gets invoked only if the policy evaluates to True and any of the attributes contained in the policy changed. On the approval screen for a work item that was generated as a result of a policy with evaluation type of Onchange, the approver can only change the value of those attributes contained in the policy, if those attributes have a permission of readwrite for that approval screen. All other attributes that exist on the approval screen have read-only permissions.

  Note: For event level workflow, if the approver clicks the Reject button, only changes made to the attributes contained in the approval policy are rejected and the next approval policy in order, gets evaluated.

  For task level workflow, if the approver clicks the reject button, the event is rejected.

  Note: For both rule types OnChange and Always, when an approver un-does all changes and clicks Approve, the changes are rejected and audited accordingly.

More Information:

Policy Order (see page 332)
Objects of Rules (see page 330)

Rule Evaluation Example

Consider the following policies, all for ModifyUserEvent in the Modify User admin task:

Policy	Rule	Evaluation
Policy1	User where (User ID = Smith01)	Always
Policy2	User where (Title = Manager)	When the Title attribute changes

Policy	Rule	Evaluation
Policy3	User where (Salary >= 80000)	When the Salary attribute changes

Policy1 is evaluated every time administrator invokes the Modify User task for user Smith01, regardless of which attribute changes.

Policy2 is evaluated when the administrator invokes the Modify User task to change the Title attribute for any user object. Policy2 is true if Title changes to Manager.

Policy3 is evaluated when the administrator invokes the Modify User task to change the Salary attribute for any user object. Policy3 is true if salary changes to 80000 or more.

In this example, if an administrator uses the Modify User task to change the Title attribute to Manager for user Smith01, then both Policy1 and Policy2 evaluate to true, and their respective workflow processes are started. In this case, the standard ordering priority applies.

Conditional rule evaluation allows an approver of one work item to change an attribute that affects another work item for the same event while the event is still pending. This is only possible for approval policies that have an evaluation type of Always. In the preceding example, if an administrator changes an attribute for user Smith01, then Policy1 is true and generates a work item. While approving the work item generated by Policy1, that approver may, on the same approval screen, change the Salary attribute for Smith01. In this case, the new Salary value for Smith01 determines whether or not Policy3 generates a work item for the same instance of ModifyUserEvent. If the approver changes the salary to 90000, then Policy3 generates a new work item which must be approved before the event itself is approved. Standard ordering priority applies.

Policy Order

All approval policies contain a Policy Order field in which a positive integer value, ordered from lowest to highest, specifies priority. The priority for each policy determines the following:

- The order in which approval rules are evaluated
- For rules that are true, the order in which workflow processes are started

A policy with a lower integer value has a higher priority, and its rule is evaluated before a policy with a higher integer value. For all policies for an event or admin task that are true, the policy with the highest priority starts its workflow process first.

Policy Order Example

This simple example demonstrates how policy ordering works. In this example, assume the policy rules are always evaluated.

If an event has multiple policies that are always evaluated, then for the event itself to be approved, all policies must be approved. However, if one policy associated with the event which has a policy evaluation type as ALWAYS is rejected, the event itself is rejected.

Note: If a policy associated with the even has an evaluation type as Onchange, only the changes associated with the attributes contained in that policy are rejected. The event itself is not rejected and the next policy in line is evaluated.

In this example, Policy1, Policy2, and Policy3 all have a policy evalutation type of ALWAYS. Policy1 evaluates to false, the workflow process named Process1 does not execute, and no work item is generated for User1. Event control immediately passes to Policy2. Policy2 and Policy3 both evaluate to true. Because of its higher priority, workflow Process2 runs first, and generates a work item for User2.

If User2 approves the work item, workflow Process3 runs and generates a work item for User3, who must then approve the work item for the event itself to be approved. These actions are shown in the following table:

Priority	Policy	Result	Workflow	Approver	Action
1	Policy1	False	Process1	User1	—
2	Policy2	True	Process2	User2	Approved
3	Policy3	True	Process3	User3	Approved

However if User2 rejects the work item, the event itself is rejected, and no work item is generated for User3, as shown in the following table:

Priority	Policy	Result	Workflow	Approver	Action
1	Policy1	False	Process1	User1	—
2	Policy2	True	Process2	User2	Rejected
3	Policy3	True	Process3	User3	—

Next, Policy1, Policy2, and Policy3 all have a policy evaluation type of ONCHANGE. If User2 rejects the work item, only changes associated with the attributes contained in Policy2 are rejected. Policy3 is then evaluated and Workflow Process3 runs and generates a work item for User3. If User3 rejects the work item, the event is rejected as all changes to this event were rejected. If User3 approves the work item, the event is approved and attribute changes contained in Policy3 get persisted.

Priority	Policy	Result	Workflow	Approver	Action
1	Policy1	False	Process1	User1	—
2	Policy2	True	Process2	User2	Rejected
3	Policy3	True	Process3	User3	Approved

Policy Description

An optional, non-searchable string description attribute has been added to the Approval policy managed object and appears on resulting work items.

Maximum number of characters supported: 255 characters

You can enter bundle/key information in the following format for the description:

```
$ (bundle=<fully qualified resource bundles name> : key=<key>)
```

Approval Policies and Multivalued Attributes

If a rule was set up for a multivalued attribute, there was no way to say that this rule should apply only on newly added or removed values for the multivalued attribute. By looking at the Policy Evaluation Type for a rule based on a multivalued attribute, this is now achieved. If the rule evaluation type is Onchange then this rule can only be applied to the new added or removed values of the multivalued attribute and not on all values of the multivalued attribute.

If the rule must be based on all values of the multivalued attribute irrespective of whether they were newly added or removed, the evaluation type for that rule must be Always.

Changes made to multivalued attributes are highlighted on the profile screen with an undo icon. If a rule evaluated to true because a new value was added or removed to a multivalued attribute, the approver approving this change sees ALL values contained in the multivalued attribute. Clicking the undo icon reverts the value for that attribute back to its original value. If an approver wants to see the removed values, clicking the Undo icon shows the original set of values.

Clicking the redo icon shows the new set of values letting the approver differentiate which were the removed values and which were the added ones. Clicking the approve button approves all the changes to this multivalued attribute. Clicking the reject button rejects all changes to this multivalued attribute. All subsequent rules pertaining to this multivalued attribute are not evaluated unless there is a new delta of values for this multivalued attribute.

Note: For rules based on multivalued attributes, the values contained in the multivalued attribute are the actual values and not the display values. For example, the display value for the state MA is Massachusetts. When creating an approval policy that is based on the state attribute, the rule should look like state=MA.

Consider the following example policies, all for ModifyUserEvent in the Modify User admin task:

Policy	Rule	Evaluation
Policy1	User where (State = MA)	OnChange
Policy2	User where (state = DC)	Always

Policy1 is evaluated every time an administrator invokes the ModifyUser task to change the state attribute and evaluates to true if the value MA is either added or removed from the state attribute.

Policy 2 is evaluated every time the administrator invokes the Modify User task for a user whose state contains the value DC.

Attributes Highlighted as Changed on Workflow Approval Screens

On an approval screen, additional attributes may appear highlighted as changed even if an administrator did not change them in the original task. This is because the screen can contain scripts that can change values of various attributes contained on the screen as a part of screen initialization or screen validation for a change of some other attribute.

Policy Examples

The following business use case examples demonstrate how you can apply workflow approval policies for an event:

Example 1:

Use Case – An administrator modifies a relational database account belonging to an employee.

Admin Task – ModifyMSSQLAccount

Event – ModifyMSSQLAccountEvent

Approval Rule – User where (Title = RDBAcctManager)

Workflow Process – ModAcctApproval (custom workflow process)

Object – Initiator of the task

Evaluation – Always evaluate the rule

Example 2:

Use Case – An administrator modifies an employee's salary to reflect a new raise.

Admin Task – Modify User

Event – ModifyUserEvent

Approval Rule – User where (Salary >= 100000)

Workflow Process – SalaryChangeApproval (custom workflow process)

Object – Primary object of the event (user)

Evaluation – Evaluate only when the Salary attribute changes

Example 3:

Use Case – An administrator adds a user to the Contractors group when that user's title changes to Contractor. This example could be divided into the following two approval policies:

Policy 1:

Admin Task – Modify User

Event – ModifyUserEvent

Approval Rule – User where (Title = Contractor)

Workflow Process – SingleStepApproval (default process template)

Object – Primary object of the event (user)

Evaluation – Evaluate only when the Title attribute changes

Policy 2:

Admin Task – Modify Group (or Modify Group Membership)

Event – AddToGroup

Approval Rule – Group where (Group Name = Contractors)

Workflow Process – SingleStepApproval (default process template)

Object – Secondary object of the event (group)

Evaluation – Always evaluate the rule

The following business use case examples demonstrate how you can apply workflow approval policies for a task:

Example 1:

Use Case – An administrator modifies an Active Directory account belonging to an employee.

Admin Task – ModifyActiveDirectoryAccount

Object – Initiator of the task

Approval Rule – User where (Title = ActiveDirectoryManager)

Workflow Process – Single Step Approval

Evaluation – Always evaluate the rule

Example 2:

Use Case – An administrator modifies a user whose employee code is HighSecurity.

Admin Task – Modify User

Object – Primary Object of the Task

Approval Rule – User where (employeenumber = HighSecurity)

Workflow Process – Single Step Approval

Evaluation – Always evaluate the rule

Example 3:

Use Case – An administrator modifies a user to assign admin roles CheckApprover and CheckSigner.

Admin Task – Modify User

Object – Identity Policy Violation

Approval Rule – IdentityPolicy where (Name = CheckRoles)

Workflow Process – Single Step Approval

Evaluation – Always evaluate the rule

How to Configure Policy-Based Workflow for Events

The procedure for configuring policy-based workflow is similar to that for configuring event-level workflow, with the additional steps of defining the approval policies which determine whether the workflow executes.

To Configure Policy-Based Workflow

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify (or Create) Admin Task.

   A Select Admin Task screen appears.

2. Search for the task you want under workflow control, and click Select.

   A Modify (or Create) Admin Task screen appears.

3. On the Profile tab, verify that Enable Workflow is checked.

4. On the Events tab, select an event to map to a process template.

   The workflow mapping screen appears.

5. Select the Policy-Based radio button, and then click Add.

   The Approval Policy screen appears.

6. Configure an approval policy (see page 341).

7. Configure participant resolvers as required by your selected workflow process.

   The participant requests are added to the process.

8. Click OK.

   Identity Management saves your event-level workflow configuration.

9. Click Submit.

   Identity Management processes the task modification.

Note: The Workflow Process list includes processes for use with both the template method and the WorkPoint method:

- When a template method process is selected (either SingleStepApproval, TwoStageApprovalProcess, or EscalationApproval), the page expands to enable participant resolver configuration.
- When a WorkPoint method process is selected, the page does not expand. Participant resolvers are configured in WorkPoint Designer.

More Information:

Participant Resolvers: WorkPoint Method (see page 310)
How to Configure an Approval Policy (see page 341)

How to Configure Policy-Based Workflow for Tasks

The procedure for configuring policy-based workflow for tasks is similar to that for configuring task-level workflow, with the additional steps of defining the approval policies which determine whether the workflow executes.

To Configure Policy-Based Workflow

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify (or Create) Admin Task.

   A Select Admin Task screen appears.

2. Search for the task you want under workflow control, and click Select.

   A Modify (or Create) Admin Task screen appears.

3. On the Profile tab, verify that Enable Workflow is checked
4. On the Profile tab, click on the pencil icon next to the Workflow Process field

   The workflow mapping screen appears.

5. Select the Policy-Based radio button, and then click Add.

   The Approval Policy screen appears.

6. Configure an approval policy (see page 341).
7. Configure participant resolvers as required by your selected workflow process.

   The participant requests are added to the process.

8. Click OK.

   Identity Management saves your task-level workflow configuration.

9. Click Submit.

   Identity Management processes the task modification.

Note: The Workflow Process list includes processes for use with the template method for task-level policy-based workflow:

- When a template method process is selected (either SingleStepApproval or TwoStageApprovalProcess), the page expands to enable participant resolver configuration.

More Information

How to Configure an Approval Policy (see page 341)

How to Configure an Approval Policy

Configuring an approval policy for an event or task involves the following steps.

1. Select an object to test.
2. Define an approval rule for the object.
3. For primary objects, decide if this is a conditional evaluation.
4. Enter the order of policy evaluation.
5. Configure a workflow process to run if the rule is true.

To configure an Approval Policy

1. On the Approval Policy screen, select an object for the rule to test from the drop-down list.

   The screen changes to reflect your selection.
2. From the new drop-down next to the object name, select a condition expression template.

   The screen changes to reflect your selection.
3. Create and edit your condition expression as required.
4. Select the Rule Evaluation option button to indicate if the rule is always evaluated, or only if an attribute in the approval condition changes.
5. Enter a positive integer value to specify the policy evaluation order (in case there are multiple policies for the event).
6. Select and configure the workflow process that executes if the rule evaluates to true.
7. Click OK to save the approval policy.

More Information:

How to Configure Event-Level Workflow (see page 285)
How to Configure Policy-Based Workflow for Events (see page 338)
How to Configure Policy-Based Workflow for Tasks (see page 340)

Policy-Based Workflow Status

Identity Management administrators can display the status of tasks containing workflow approval policies using the following standard system tools:

- View Submitted Tasks tab
- User History tab
- Reports and logs

The submitted task and task history information includes:

- Task and event information
- Workflow and approval rule information
- Approval rule evaluation results

See the System tab documentation for submitted task history descriptions.

More Information:

Description of Event Status (see page 414)
Task Status in Identity Management (see page 407)

Global Event Level Policy-Based Workflow Mapping

An event can be mapped to a workflow process from the Management Console, or be associated with policy-based workflow approval policies in a specific task. The new Configure Global Policy-based Workflow for Events task, lets administrators set up policy-based workflow mapping for events at the environment level. Unlike setting up policy-based workflow for an event in an admin task, the configured policy-based workflow mappings are applied to all tasks that generate the event.

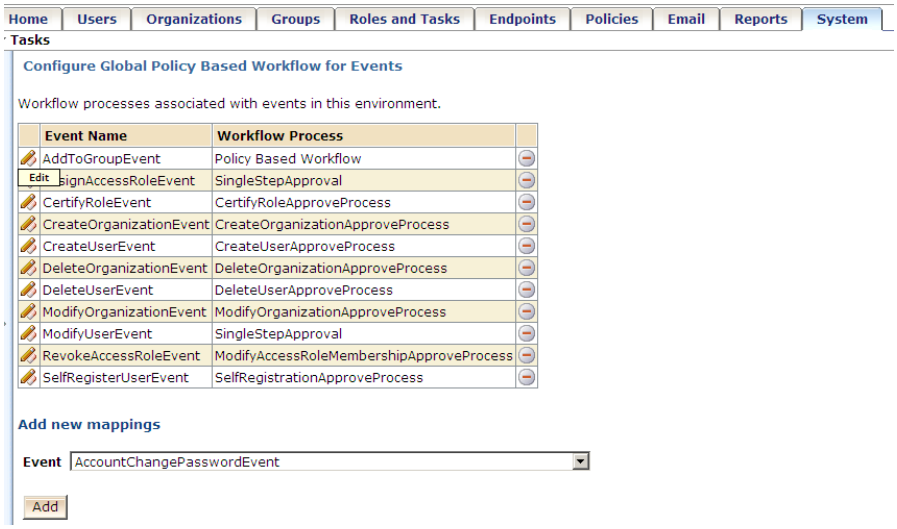
Note: The Configure Global Policy Based Workflow for Events task works only when workflow is enabled. Executing this task when workflow is disabled throws an error.

This task has been added to the System tab. When a task is submitted, the workflow process of each event in this task is retrieved in the following way:

Any workflow configured for the event for that admin task takes precedence. An event can be configured for either policy-based or non-policy based workflow. If policy-based workflow is configured for the event for that admin task the workflow process associated with the policy is invoked. If no rule matched, no workflow is invoked for the event. Likewise, if non-policy based workflow is configured for the event for that admin task, the workflow process associated with the policy is invoked. If no workflow was configured for the event for that admin task, global workflow configuration for that event takes precedence.

Configure Global Policy Based Workflow for Events Task Screen

The Configure Global Policy Based Workflow for Events tasks lets an administrator configure policy or non-policy based workflow for all events in the current environment. Clicking the task displays the default event mapping to workflow process definitions. Each event mapping can be modified or deleted, and new event mappings can be added for events that have not been configured.



The fields on this screen are as follows:

Workflow processes associated with events in this environment.

Specifies the workflow processes associated with approval policies.

Add New Mappings

Specifies an approval policy to map to a workflow process.

Add Button

Adds the new mapping.

Adding or modifying a mapping opens the Workflow Mapping screen where you can select the process mappings and approval policies. The behavior is the same as the event level workflow configuration. Clicking the Add button on the Workflow Mappings page brings up another page where you can configure an approval policy.

More Information

How to Configure Policy-Based Workflow for Events (see page 338)
How to Configure an Approval Policy (see page 341)

Online Requests

Identity Management lets you create general purpose online request tasks. The default online request implementation is comprised of a set of related tasks for both self-modification requests and administrative user modification requests. However, the online request feature could easily be implemented for other Identity Management request tasks.

A user modification request triggers a workflow process which generates a work item. Workflow participants can either approve and implement the work item, or reject it. The user initiating the task enters a description of the request in the history editor, a text area which Identity Management uses to maintain a history of the request. This history editor can be configured to allow participants to leave comments about the action they perform on the work item. These comments become part of the cumulative work item history.

New actions in addition to (or in place of) the standard approve and reject actions are also possible. For example, a business participant can clarify or comment on the request, and a technical participant can implement the request. These new activities can be represented by new workflow action buttons like “Clarify” and “Implement” which you can add to the standard “Approve” and “Reject” buttons on the approval task.

Online Request Tasks

There are five tasks that work together to make up the default online request implementation. These tasks demonstrate the use of custom requests, history, and workflow action buttons:

Note: The admin tasks (Change My Account and Create Online Request) are configured by default for event-level workflow using the Consultation Process template.

Change My Account

This is a self-modification admin task that creates a user account change request. It has a Request tab with a history editor for describing the request, and a Profile tab with read-only user details.

Create Online Request

This is a user modification admin task that creates an account change request for a particular user. It has a Request tab with a history editor for describing the request, and a Subject Profile tab with read-only user details.

Approve Online Request

This is an approval task that allows the business participant to approve or reject the task, or to request further clarification of the task. This task has a Request tab with a history display and a history editor for queries or comments, a read-only Subject Profile tab, and an Assignees tab.

Clarify Online Request

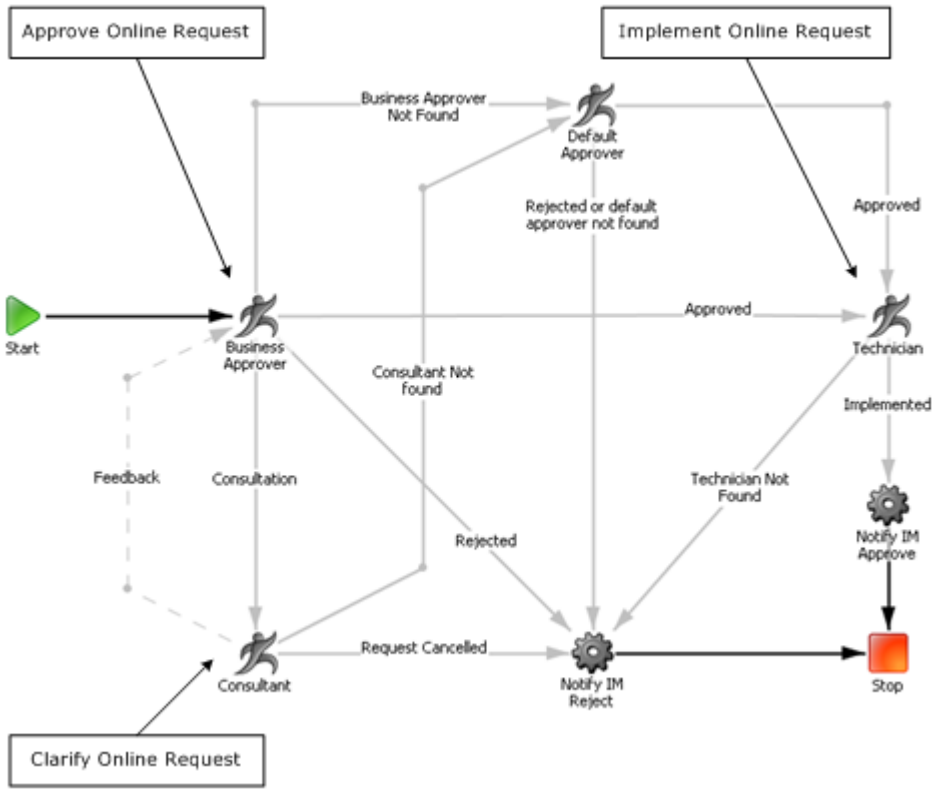
This is an approval task that allows the clarifying participant to respond to a clarification request, and sends the task back to the business participant for approval. It has a Request tab with a history display and a history editor for comments, and a read-only Subject Profile tab.

Implement Online Request

This is an approval task that allows the technical participant to implement the task and to add a comment to the task history. It has an Implement Request tab with a history display and a history editor for comments, a read-only Subject Profile tab, and an Assignees tab.

Online Request Process

The online request tasks are controlled by a workflow process template called Consultation Process, shown as it appears in WorkPoint Designer:



The Consultation Process includes four manual activities which correspond to approval tasks in the online request implementation:

- An activity for the business approver, who rejects the work item, approves the work item and passes it on to the technician, or requests further clarification from the consultant.
- An activity for the consultant, who clarifies the work item and sends it back to the business approver.
- An activity for a default approver, who takes over if either the business approver or consultant cannot be contacted.
- An activity for the technician, who implements the request and completes the work item.

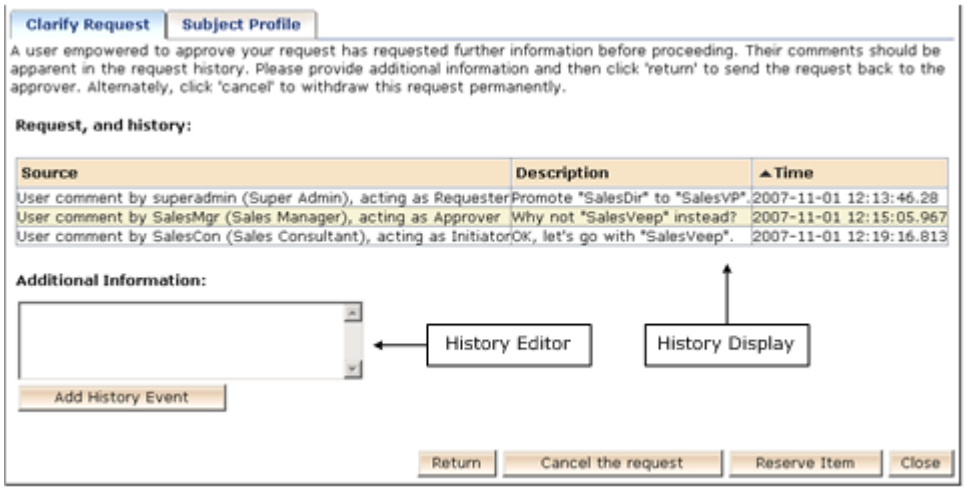
Online Request History

The online request history feature allows participants to create a record of work item actions. As responsibility for the work item passes from one participant to another, the new participant is able to review work item history before taking action.

Two controls are used to implement online request history:

- The history display is a read-only table containing details of previous history entries in chronological order.
- The history editor is a text box for creating new history entries. It also has an optional button for adding multiple entries without submitting the work item.

By default, the history editor and history display appear on the Request tabs for all tasks associated with the online request implementation. The following screen illustrates the history controls in the Clarify Online Request task:



Using Online Requests

The following steps describe the online request workflow process. For each step, the generated IM task appears in parentheses. At every step in the process, the participant can add a comment in the history editor. This comment appears in the history display to the next participant in the workflow process.

1. The Task Initiator requests a modification to a Identity Management user (Create Online Request).
2. The Business Approver receives a work item, and does one of the following:
   - Approves the work item (Approve Online Request).
   - Rejects the work item and terminates the workflow process. No new task is generated.
   - Requests a clarification from the consultant (Clarify Online Request).
3. The Consultant receives a work item, and does one of the following:
   - Adds a clarification and returns the work item to the Business Approver. No new task is generated.
   - Cancels the work item and terminates the workflow process. No new task is generated.
4. The Technician receives a work item, and implements the request (Implement Online Request).

Workflow Action Buttons

Approval tasks in Identity Management historically have Approve and Reject action buttons that appear on their corresponding work item screens. Workflow action buttons allow administrators to extend the functionality of Identity Management tasks and workflows by adding action buttons to approval tasks, and by removing or modifying existing buttons. (The standard Approve and Reject buttons are implemented in the same manner as custom workflow action buttons.)

For example, a workflow process might require an action that allows mid-level participants to escalate certain cases to a more senior participant for final approval or rejection. These mid-level participants could add a comment or recommendation using the history editor, and then send the work item to the senior participant to review and approve or reject.

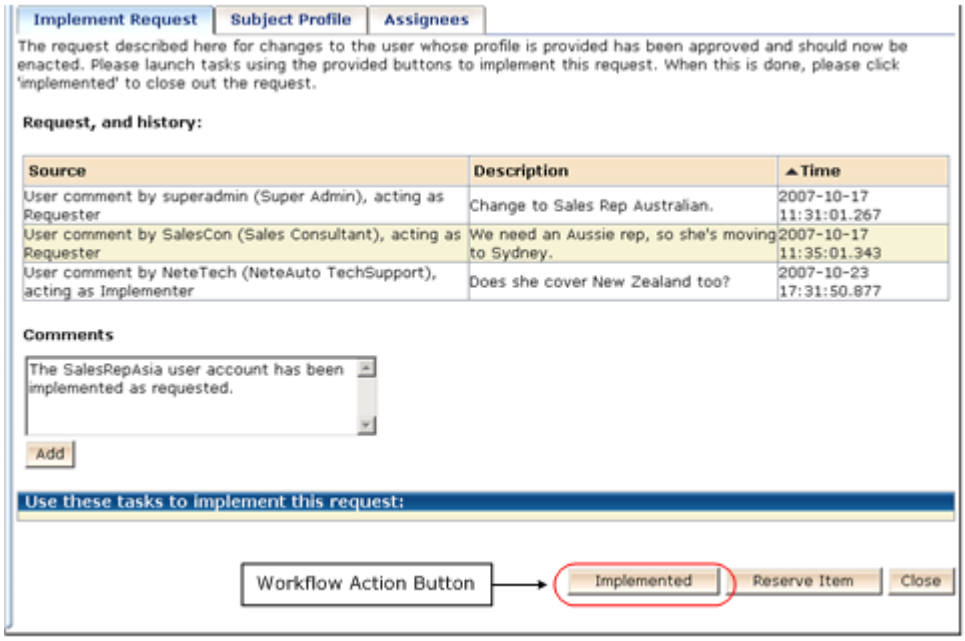
Adding or removing workflow action buttons requires appropriate changes to the WorkPoint workflow process that provides the business logic for handling these new actions.

More Information:

Button Configuration In Identity Management (see page 349)
Workflow Buttons in Approval Tasks (see page 348)
Button Configuration in WorkPoint Designer (see page 351)

Workflow Buttons in Approval Tasks

Workflow action buttons correspond to transition nodes pointing away from manual activity nodes on a WorkPoint process diagram. For example, in the Consultation Process, the Technician activity node has a single transition called Implemented. This corresponds to the "Implemented" button on the Implement Online Request approval task, shown in the following figure:



Note: The "Reserve Item" and "Close" buttons are governed by Identity Management programming logic and are not under workflow control.

More Information:

Workflow Action Buttons (see page 347)
Button Configuration In Identity Management (see page 349)

Button Configuration In Identity Management

To configure a workflow action button, click the button named Workflow Action Buttons on the Profile tab of an Approval task.

The button Profile tab has a table with a row for each workflow action button. Each button row has the following four properties, which correspond to columns in the table:

Display Name

The name that appears on the button in the approval screen. The name is a conditionally localized value, which can be either a string or a key for a localized string in a resource file.

Action

The value that is passed back to the workflow process when the option is selected. This value is an attribute of the corresponding transition node in the WorkPoint process diagram. The value is a non-localized string. The default settings are “approved” and “rejected”.

Tool Tip

A short description (or tool tip) of the button action which appears when a user hovers the mouse cursor over the button. The tool tip is a conditionally localized value, which can be either a string or a key for a localized string in a resource file.

Long Description

A longer description of the button action which adds a message describing the action on the "View Submitted Task" screen. If the description is blank, the message that is displayed on the "View Submitted Task" screen is the button name. The name is a conditionally localized value, which can be either a string or a key for a localized string in a resource file.

More Information:

Button Configuration in WorkPoint Designer (see page 351)

Adding Workflow Action Buttons

To add a new button to an existing workflow process, perform the following high-level steps:

1. Add the workflow button in Identity Management.

   For instructions, see How to Add a Workflow Action Button (see page 350).

2. If necessary, add localization keys.

   For instructions, see the *Configuration Guide*.

3. Add any new required nodes in WorkPoint Designer.

   For instructions, see the WorkPoint Designer online help.

4. Define a script in the WorkPoint Designer transition node.

   For instructions, see Button Configuration in WorkPoint Designer (see page 351).

More Information:

Button Configuration in WorkPoint Designer (see page 351)
How to Add a Workflow Action Button (see page 350)

How to Add a Workflow Action Button

ou can add workflow action buttons to approval tasks in Identity Management.

To add a workflow action button to an admin task

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.

   The Select Admin Task screen appears.

2. Search for the approval task, and click Select.

   The Modify Admin Task screen appears.

3. On the Profile tab, click the button named Workflow Action Buttons.

   The Workflow Action Button Profile tab appears.

4. Click "Add Button" to add a new button to the approval task.

5. Enter the button property information.

6. Click OK.

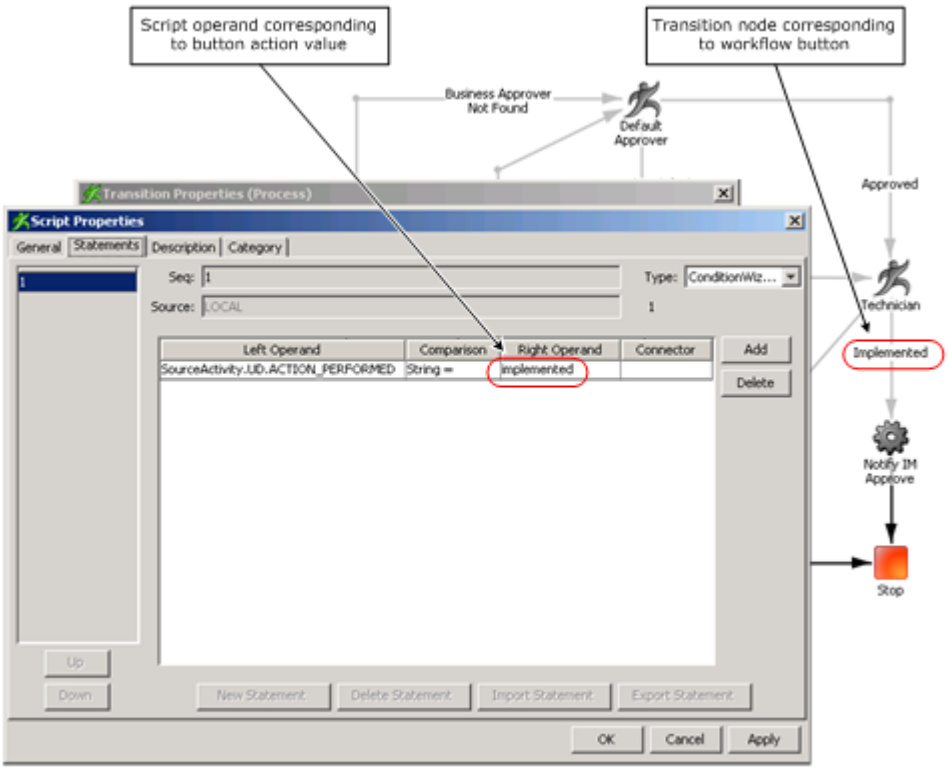
   Identity Management saves the new button information.

7. Click Submit.

   Identity Management processes the task modification.

Button Configuration in WorkPoint Designer

In WorkPoint Designer, workflow action buttons are configured using transition node script properties, as shown in the following figure:



By default, workflow action buttons use the following script properties to perform a string comparison:

- Left Operand--ACTION_PEFORMED, which is defined in the User Data properties of the preceding manual activity node.
- Right Operand--The Action value of the button, which is defined in the button profile tab of the User Console.

Note: See the WorkPoint Designer online help for information about activity node and transition node scripts and properties.

More Information:

Button Configuration In Identity Management (see page 349)

Work Lists and Work Items

A *work list* is a list of work items (or approval tasks) that appears in the User Console of the participant authorized to approve the task. Work items correspond to manual activities in a workflow process. Work items are represented as rows in the work list.

Work items can be added to a work list in the following ways:

- A participant resolver determining a list of approvers.
- Receiving delegated work items from another user.
- Reassigning it to another user.

Work items can be removed from a work list in the following ways:

- Completing (approving or rejecting) the work item.
- Reassigning it to another user.
- Reserving it. This removes it from the work list of all other participants.

Note: When you accept or reject a work item, the change is not immediate. For example, if you reject a work item, that item still appears in you work list until the workflow process records the information and progresses the process to the next node.

The information tabs that appear on a work item depend on whether the work item was generated by workflow under task-level or event-level control:

- **Profile**—Provides profile information about the object affected by the event (event-level only).
- **Task Details**—Provides detailed information for all events within the task (task-level only).
- **Approvers**—Lists all individual approvers and delegators for the task or event (task-level and event-level)

Displaying a Work List

Your work list appears automatically when you log into the User Console if you have been assigned as a participant to approve tasks (or work items) initiated by other users.

To display your work list manually

1. In the User Console, select Home, View My Work List.

   Your work list appears.

2. Click the name of a work item to display it.

   The selected work item appears.

Administrators can manage work items for users over whom they have scope.

Note: Managing a user's work items allows administrators to reserve a work item. Viewing a user's work list does not allow work item changes of any kind.

To view the work list of another user

1. In the User Console, select Users, Manage Work Items, View User's Work List.

   A select user screen appears.

2. Search for the user whose work list you want to view, and click Select.

   The user's work list screen appears.

To manage work items for another user

1. In the User Console, select Users, Manage Work Items, Manage User's Work Items.

   A select user screen appears.

2. Search for the user whose work items you want to manage, and click Select.

   The user's work list screen appears.

3. Click the name of a work item to display it.

   The selected work item appears.

Enabling Work List Search Screen

You can enable the pre-configured search screens to find work items.

1. In the Management Console, go to in Home, Environments, <environment>, Roles.
2. Click Import. Import the file worklistsearch.xml from this location:
   `<INSTALL_LOCATION>\CA\IdentityManager\IAM Suite\Identity Manager\tools\worklistsearch`
3. Click Finish.
4. Log into the CA Identity Manager environment.
5. In the User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.
6. Search for and select the View My Work List task.
7. Select the Search tab. Click Browse.

   A list of screen definitions appears.
8. Click New.

   A list of new screen types appears.
9. Select Work List Search Screen and click OK.
10. Enter details to configure the work list search screen and click OK.

    The new search screen is added to the list of screen definitions

Reserving Work Items

You can reserve a work item to "check it out" and remove it from the work list of other participants. Reserving a work item holds it for the user performing the reservation.

If the reserving user releases the work item, it becomes available again on the work list of other participants. If the reserving user approves or rejects the work item, it is completed, and no longer available to other participants.

More Information:

Delegation and Reserved Work Items (see page 355)
Reassignment and Reserved Work Items (see page 355)

Reassignment and Reserved Work Items

If a user has a work item reserved while it is reassigned, the user keeps it reserved. But if the user then releases that work item, he loses access to it.

An administrator can reassign, reserve, or release another user's work item, but cannot approve or reject another user's work item. Only the assigned work item participant can do that.

More Information:

Reassigning Work Items (see page 361)

Delegation and Reserved Work Items

While a delegation is active, either the delegate or the delegator may reserve a work item. A work item reserved by one user cannot appear on another user’s work list.

For example, if a delegate has a work item reserved while the delegation is withdrawn, the delegate keeps the work item reserved. But if the delegate then releases that work item, he loses access to it.

If a user who is a delegate is deleted while the delegate has a work item reserved, the delegate still retains the work item. If the delegate then approves the work item, auditing can no longer determine who delegated it.

If a delegate has a work item reserved while the delegation is withdrawn, the delegate retains access until the work item is completed or released.

More Information:

Delegating Work Items (see page 356)
Reserving Work Items (see page 354)

How to Reserve or Release a Work Item

You reserve a work item to "check it out" and remove it from the work list of other participants.

You release a reserved work item to make it available on the work list of other participants.

Note: The only way to free a reserved work item is to explicitly release it.

To reserve or release a work item

1. In the User Console, select Home, View My Work List.

   Your work list appears.

2. Select the work item you want to reserve or release.

   The expanded work item screen appears.

3. Click Reserve Item or Release Item.

   Identity Management confirms your action.

Delegating Work Items

Work item *delegation* lets a user (the delegator) specify that another user (the delegate) is allowed to approve tasks in the delegator's work list. A delegator can assign work items to another approver during periods when the delegator is “out of the office.” Delegators retain full access to their work items during the delegation period.

Delegated work items are not changed in any way. Logging indicates whether a work item was delegated.

Delegation works by allowing the delegate to “impersonate” the delegator and view the items on the delegator's work list. When viewing a work list, delegates see their own work items as well as the delegator's work items.

Delegation is not transitive. A delegate can only see work items that the delegator has assigned directly. For example, If user A delegates work items to user B, and user B delegates work items to user C, user C can only see work items belonging to user B, and not any work items that may have been work items delegated to user B by user A.

More Information:

Delegation and Reserved Work Items (see page 355)

Delegation Well-Known Attribute

Delegation uses the following well-known attribute:

%DELEGATORS%

This well-known attribute stores the names of users who are delegating to the user with the attribute, as well as the time when the delegation was created.

How to Enable Delegation

You must have workflow approval delegation enabled before you can use a delegate work items to another user. By default, delegation is disabled.

To enable workflow approval delegation

1. Open the Management Console by entering the following URL in a browser:

   `http://hostname/iam/immanage`

   hostname

   Defines the fully qualified domain name of the server where Identity Management is installed. For example, myserver.mycompany.com:port.

2. Click Environments, and select the name of the appropriate Identity Management environment.
3. Click Advanced Settings, and then click Workflow Approval Delegation.
4. Select the Enabled check box, and then click Save.

More Information:

How to Delegate for Yourself (see page 357)
How to Delegate for Another User (see page 360)

How to Delegate for Yourself

You can delegate work items to another user during periods when you are "out of the office." Delegators still retain full access to their work items during the delegation period.

To delegate work items for yourself

1. In the User Console, select Home, Out of Office Assistant.

   The Out of Office Assistant screen appears.

2. Click Add User.

   A select user screen appears.

3. Search for and select one or more users to act as delegate.

   The users are added to the delegate list.

4. Click Submit.

   The task is submitted and the delegation is saved.

Note: Users who are already delegates do not appear in the search results when adding a delegate.

More Information:

How to Enable Delegation (see page 357)

Time-Based Work Item Delegation

In previous releases, you could specify the start time, but not the end time for delegations. Newly created delegations have their dates for delegation set to true, with the Default start time set to now.

At modification time, start and end dates can be changed. The default end time is one week from start date.

To change the Start or End dates, do the following:

1. From the User Console's Home tab, select Out of Office Assistant.
2. Click the pencil icon next to the User ID whose delegation information want to change.

   The Edit Delegation Details screen appears.
3. Click on the Calendar next to Start Date to change the delegation start date.

   Note: An error message is displayed when the Delegation Start Date selected is before the current date.
4. If you want to select an end date, check the Has End Date checkbox.

   The End Date field is now available to set the end date.
5. Click on the Calendar next to End Date to set a date for the delegation to end.
6. Once the dates have been set, Click OK.

Alternately, you can do the same from the Delegate Work Items tab when Creating or Modifying a user.

Enable Time-Based Work Item Delegation

To enable time based work item delegation in an existing environment on upgrade, do the following:

From the Management Console

1. Navigate to the Environments page.
2. Drill down into the selected Environment, Advanced Settings, Work Item Delegation.
3. Uncheck Enabled check box.
4. Save the changes and restart the Environment.
5. Drill down into Advanced Settings, Work Item Delegation.
6. Check Enabled checkbox.
7. Save the changes and restart the Environment.

Note: This procedure is for existing environments only. Time-based workflow item delegation is enable for new environments.

Out of Office Assistant Screen

You use the following Out of Office Assistant screen to add and remove delegate for yourself:

The Out of Office Assistant screen displays a list of your current delegates. In addition to columns that identify the delegate, three additional columns are included in the list:

Start Date

Displays the date the delegation was created.

End Date

Displays the date the delegation is to end.

Has Delegates

Indicates whether the delegate has delegated work items to another user.

When you click the pencil icon next to the selected User ID, the Edit Delegation Details screen appears where you can change the Start Date and specify the End Date for the delegation.

How to Delegate for Another User

Administrators can delegate work items from one user (the delegator) to another. For example, a user may be out of the office unexpectedly, or an administrator may need to assign a large workload to multiple users.

Administrators can only delegate work items for users over whom they have scope. Similarly, they can only add or remove users they manage from the list of delegates.

To delegate work items for another user

1. In the User Console, select Users, Manage Work Items, Delegate Work Items.

   A select user screen appears.

2. Search for the user whose work items you want to delegate (the delegator), and click Select.

   A delegate work items screen appears.

3. Click Add User.

   A select user screen appears.

4. Search for and select one or more users to act as delegate.

   The users are added to the delegate list.

5. Click Submit.

   The task is submitted and the delegation is saved.

Note: Users who are already delegates do not appear in the search results when adding a delegate.

More Information:

How to Enable Delegation (see page 357)

How to Remove a Delegation

If a user logs into Identity Management with delegations in place, Identity Management displays the following reminder:

```
You have delegations in place. Please check that they are still required.
```

To remove a delegation for yourself

1. In the User Console, select Home, Out of Office Assistant.

   The Out of Office Assistant screen appears.

2. Click the minus sign (-) for delegates you want to remove.

   The delegates disappear from the list.

3. Click Submit.

   The task is submitted and the delegation is removed.

To remove a delegation for another user

1. In the User Console, select Users, Manage Work Items, Delegate Work Items.

   A user search screen appears.

2. Search for and select the user whose delegations you want to remove.

   The delegates list appears.

3. Click the minus sign (-) for delegates you want to remove.

   The delegates disappear from the list.

4. Click Submit.

   The task is submitted and the delegation is removed.

Note: You can only remove a delegate if you have scope over that user.

Reassigning Work Items

Reassignment allows users and administrators to change the assignees of a work item after it is created. An administrator can:

- View another user's work list
- Add and remove work item assignees
- Change the reserve status of work items

For example, an administrator can reassign a work item or release a reserved work item from a user who is not acting on it.

If a user has a work item reserved while it is reassigned, the user keeps it reserved. But if the user then releases that work item, he loses access to it.

If a delegate has a work item reserved while the delegation is withdrawn, the delegate retains access until the work item is completed or released.

More Information:

Reassignment and Reserved Work Items (see page 355)

The Approvers Tab

You perform reassignment on the Work Item Approvers tab, which displays a list of current work item approvers (or assignees). When you perform reassignment, you assign the open work item to all approvers in the list. Therefore, to reassign a work item to a new assignee, you must also remove the current assignee.

How to Reassign Work Items

Reassigning a work item from one user to another is a two-step process:

- Select a new approver.
- Remove the current approver.

Note: You must have scope over users to whom you want to reassign.

To reassign your own work item

1. Select Home, View My Work List.

   Your work list appears.

2. Select a work item to expand it.

3. Select the Approvers tab.

   The list of all current approvers appears, including the user whose work list you are managing.

4. Click Add Assignees.

   A select user screen appears.

5. Search for and select one or more users to whom you want to reassign.

   Note: For ALL and SUBSET approval modes, you can only reassign a work item to *one* user.

6. Click the minus sign button (-) to remove yourself as an assignee.

7. Click Perform Reassignment.

   The work item appears on the work lists of the reassigned users.

Note: An administrator can reassign, reserve, or release another user's work item, but cannot approve or reject another user's work item. Only the owner of the work item can do that.

To reassign another user's work item

1. Select Users, Manage Work Items, Manage User's Work Items.

   A select user screen appears.

2. Search for the user whose work items you want to reassign, and click Select.

   The Manage User's Work Items screen appears.

3. Select a work item to expand it.

4. Select the Approvers tab.

   The list of all current approvers appears, including the user whose work list you are managing.

5. Click Add Assignees.

   A select user screen appears.

6. Search for and select one or more users to whom you want to reassign.

7. Click the minus sign button (-) to remove the current assignee.

8. Click Perform Reassignment.

   The work item appears on the work lists of the reassigned users.

Bulk Operations on Work Items

With this release of Identity Management, the following bulk operations can be performed on selected work items:

- Approve
- Reject
- Reserve
- Release

In the User Console, the Configure Work List tab has been enhanced to include a new Supports bulk workflow operations check box. When this check box is enabled, the user can bulk approve, reject, release, and reserve work items that they own or work items from any delegators. Administrators can only perform these bulk operations on work items using the Manage User's Work Items task.

Note: Bulk operations cannot be enabled for any View type tasks, such as View My Work List.

Configure Work List Tab for Bulk Operations

To configure the Work List tab to support bulk operations on work items, follow this procedure.

From the Roles and Task Tab in the User Console

1. Select either:
   - Roles and Tasks.
   - Tasks, Roles and Tasks.
2. Select Admin Tasks, Manage Admin Tasks.
3. Click Search.
4. Select Manage User's Work Items.
5. From the Tabs Tab, click the pencil icon next to Work List.

   The Configure Work List screen appears.
6. Select Support bulk workflow operations.
7. Save the changes and submit the task.

   Bulk operations on work items are now available.

Chapter 13: Email Notifications

This section contains the following topics:

Email Notifications in Identity Management

Email notifications inform Identity Management users of tasks and events in the system. For example, Identity Management can send an email to approvers when an event or task requires an approval.

Identity Management provides the following methods for configuring email notifications:

- **Email Notification Policies**

  Email notification policies enable business administrators to create, view, modify, and delete email notifications by using tasks in the User Console. No coding is required to create email notifications.

  Administrators can define the content of an email, when it is sent, and who receives it. The content of the email, which is defined in an HTML editor, can contain dynamic information, such as the current date or event information, which Identity Management populates when the email is sent. For example, you can configure an email notification that is sent to an approver when a new user is created. The email can contain the user's login information, date of hire, and manager.

  Note: Email notification policies are Policy Xpress policies (see page 219) that are created and managed by a separate set of tasks.

- **Email templates**

  In this method, email notifications are generated from email templates. Identity Management provides default email templates that can used as installed, or that can be customized by system administrators. These administrators use an Email Template API to specify dynamic content, such as the list of recipients, and information about the event that triggers the email.

Identity Management can generate email notifications when the following occurs:

- An event requiring approval or rejection by a workflow approver is pending

  Note: If you have a Workpoint approval process that has more than one approval activity, the email notification configured in the User Console tasks sends a notification for each activity. If you use the email templates for the same notification, only one email is sent to approvers (when the event reaches the pending state).

- An approver approves an event or task
- An approver rejects an event or task
- An event or task starts, fails, or completes
- A user is created or modified

To use Identity Management email notifications, configure your SMTP settings (see page 368). If you are using the email template method, you also enable email notifications in Identity Management.

How to Select an Email Notification Method

The following table summarizes the differences between email notification policies and the email templates:

Activity	Email Management Tasks	Email Templates
Configuring email notifications	Administrators use admin tasks in the User Console to create, modify, view, and delete email notifications.	Administrators modify default templates in the Identity Management Administrative Tools.
Configuring when emails are sent	Identity Management can generate email notifications when certain events or tasks occur. The email management tasks and the email templates support the same events and tasks, however, the email management tasks provide more granularity in some cases. Email notifications are supported for the following tasks and events: ■ An event requiring approval or rejection by a workflow approver is pending ■ **Note:** If you have a Workpoint approval process that has more than one approval activity, the email notification configured using the email management tasks sends a notification for each activity. If you use the email templates for the same notification, only one email is sent to approvers (when the event reaches the pending state). ■ An approver approves an event or task ■ An approver rejects an event or task ■ An event or task starts, fails, or completes ■ A user is created or modified	
Adding dynamic content to emails	Administrators add dynamic content to the body of an email message by selecting from a list of options in the Content tab of the Create Email or Modify Email tasks. Identity Management automatically populates the dynamic content based on information in the event ot task that triggers the notification.	Administrators use the Email Template API to customize the default email templates, which are used to generate email notifications.

Activity	Email Management Tasks	Email Templates
Supporting existing email notifications	Email notifications that are configured using the email management tasks are based on Policy Xpress policies. If you upgraded from Identity Management Option Pack 1 to Identity Management 1.53, the email notifications that you configured in Policy Xpress will continue to work. However, you manage those email notifications using the email management tasks, instead of Policy Xpress.	Email notifications that you created using the email template method in previous versions of Identity Management will continue to work in Identity Management 1.53.

Configure SMTP Settings

Before enabling email notifications, configure the SMTP settings. See the following sections to configure SMTP settings for your application server.

Configure SMTP Settings on JBoss

1. In a text editor, open the mail service deployment descriptor as follows:

   Single node: *jboss_home*\server\default\deploy\mail-service.xml

   Cluster: *jboss_home*\server\all\deploy\mail-service.xml

2. Modify the mail.smtp.host property with the name of your SMTP server as follows:

   ```
   <-- Change to the SMTP gateway server -->
   <property name="mail.smtp.host" value="your_smtp_server "/>
   ```

   For example:

   ```
   <property name="mail.smtp.host" value="smtp.mailserver.company.com"/>
   ```

3. Save the mail-service.xml file.

4. In a text editor, open the following email properties file:

   Single node:
   jboss_home\server\default\deploy\iam_im.ear\config\com\netegrity\config\email.properties

   Cluster:*jboss_home*\server\all\deploy\iam_im.ear\config\com\netegrity\config\email.properties

5. To set the email return address that workflow generated email uses, locate the admin.email.address property and set the value to the appropriate email address. For example:

   ```
   admin.email.address=admin@company.com
   ```

6. If you are using the email template method, enable email notifications in the Management Console.

   You do not need to enable email notifications in the Management Console if you are using email notification policies.

Configure SMTP Settings on WebLogic

You configure email settings in the WebLogic Server Administration Console and in an email.properties file.

To configure email settings for WebLogic

1. In the WebLogic Server Administration Console, create a mail session with the following properties:

   - **mail.smtp.host** property: Set this value to your SMTP server. For example, mail.smtp.host=mymailserver.company.com
   - **mail.transport.protocol** property: Set this value to SMTP. For example, mail.transport.protocol=smtp
   - **JNDI Name**: nete/Mail
   - **Target**: the WebLogic server name

2. In a text editor, open the following email properties file for Identity Management:

   weblogic_domain\applications\iam.ear\config\com\netegrity\config\email.properties

3. Set the email return address used by workflow generated emails by locating the admin.email.address property and setting the value to the appropriate email address. For example:

   ```
   admin.email.address=admin@company.com
   ```

4. Enable email notification in the Management Console.

   Note: You do not need to enable email notifications in the Management Console if you are using email notification policies.

Configure SMTP Settings on WebSphere

The imsSetup utility that you run after installing the Identity Management components configures a new mail session object called mailMail.

For the email notification feature to work correctly, specify the server that WebSphere connects to when sending email in the Mail Transport Host field for the mailMail session.

The mailMail session is located in Resources, Mail Providers, Built-in Mail Provider, Mail Sessions, mailMail in the WebSphere Administrative Console.

Note: To view the mailMail object, change the Scope to Server in the Mail Session screen. If you do not change the scope to Server, the mailMail object is not displayed.

For more information on configuring a WebSphere mail provider, see the WebSphere documentation.

If you are using the email template method, enable email notification in the Management Console after you configure the SMTP settings.

Note: You do not need to enable email notifications in the Management Console if you are using email notification policies.

How to Create Email Notification Policies

You can use the User Console to create email notification policies that send emails when certain actions take place. For example, you can create an email notification policy that sends an email to notify approvers when a new user is created.

Follow these steps:

1. Select System, Email, Create Email.
2. Select one of the following options:
   - Create a new object of type Managed Email
   - Create a copy of an object of type Managed Email

     Uses an existing email notification policy as a template for creating a policy.
3. Provide basic information about the email notification policy in the Profile tab.

4. Specify when Identity Management sends the email in the When to Send tab.

   The When to Send tab provides several options to specify the actions that trigger email notifications.

5. Specify the recipients of the email in the Recipients tab.

6. Define the subject and content of the email in the Content tab.

   You can specify dynamic content, such as the date, task or event name, and user attributes in the email content.

More Information:

When to Send Tab (see page 372)
Recipients Tab (see page 374)
Content (see page 375)
Email Notification Profile Tab (see page 371)

Email Notification Profile Tab

The Profile tab in email management tasks allows you to specify basic information about an email notification policy. This tab includes the following fields that:

Email Name

Identifies the email notification policy in the User Console.

Note: The email name is not displayed when the email is sent. The name is only used to manage the email notification policy in the User Console.

Category

Groups email notification policies to simplify management.

Specify an existing category by selecting it from the drop-down list, or select the second option button and enter the name of a new category.

Description

Describes the email notification policy to administrators.

The description is not displayed when the email is sent.

Enabled

Specifies that Identity Management will send the email when the conditions defined on the When to Send tab are met.

Custom Data

Creates a custom data element in Policy Xpress that can be used to configure custom recipients or custom content.

Custom data elements can also be used as parameters in other data elements.

Note: Data (see page 225) provides more information about data elements.

When you click Custom Data, Identity Management opens a screen where you can add new data elements.

Entry Rules

Defines rules for when Identity Management sends email notifications in cases where the default rules in the When to Send tab are not granular enough.

For example, the When to Send tab provides a default rule that sends email when any attribute of a user profile is modified. If you want Identity Management to send an email only when a user's department changes, you can create a custom entry rule. (In this case, you create a custom data element that identifies when the department changes, and then you create an entry rule that uses the custom data element you created.)

Note: The section Entry Rules (see page 227) provides more information.

More information:

Data Elements (see page 225)
Entry Rules (see page 227)

When to Send Tab

Identity Management provides several default options that determine when email is sent. Some of these options require additional information, such as a task or event name. For example, sending an email when a task starts requires selecting the task that triggers the email.

You can select one or more of the following When to Send options:

User Created

Sends an email when a user has been created. The email is sent when the CreateUserEvent reaches completion.

User Modified

Sends an email when a user has been modified. The email is sent when the ModifyUserEvent reaches completion.

Workflow Pending

Sends an email when a workflow process assigns an approver. When you select this option, specify the applicable workflow process. Email that is defined with this policy sends individual email to approvers at every step of the selected workflow process.

Workflow Pending Email

Sends an email when a workflow process reaches a specified activity. When you select this option, specify the applicable workflow process. Email that is defined with this policy sends individual email notification for each approval step.

Event Started

Sends an email when an event reaches the Before state. When you select this option, specify the event.

Note: If you specify Event Started, and the email fails to send, the event associated with the notification will not execute.

Event Ended

Sends an email when an event reaches the After state. When you select this option, specify the event.

Event Approved

Sends an email when an event reaches the Approved state. When you select this option, specify the event.

Event Rejected

Sends an email when an event reaches the Rejected state. When you select this option, specify the event.

Event Failed

Sends an email when an event fails. When you select this option, specify the event.

Task Submitted

Sends an email when the task starts processing. When you select this option, specify the task.

Task Complete

Sends an email when the task completes. When you select this option, specify the task.

Task Failed

Sends an email if the task fails. When you select this option, specify the task.

Recipients Tab

You can configure multiple recipients for the To, CC, or BCC fields of an email. The recipient list may be static, or it may depend on the type of action that triggers the email, and the users involved.

To specify recipients, select the Edit icon next to the To, CC, or BCC field in the Recipients tab. Then, select one of the following options, which allow you to configure the list of recipients:

Workflow Approvers

Sends the email to all approvers in the workflow process. This option is only applicable if the email is sent for a workflow pending event.

Manager

Sends the email to the manager of the user whom the task has been performed on.

Note: To use the Manager recipient option, configure the manager attribute for the environment. To configure the manager attribute, go to Environments, *EnvironmentName*, Advanced Settings, Miscellaneous in the Management Console. Set managerattribute to the name of the physical attribute that stores the unique name of a user's manager.

For relational databases, specify the attribute using the following format:

tablename.attribute

Group members

Sends the email to all members of a group. Selecting this option opens a drop-down list with available group names.

Role members

Sends the email to all members of an admin role. Selecting this option opens a drop-down list with available role names.

Static address

Sends the email to a selected email address. You can specify the email address in the additional text area available.

Note: Do not specify more than one address in the text area.

User

Sends the email to the user whom the task was performed on.

Initiator

Sends the email to the person who made the request.

Custom

Allows you to select a custom data element to define the recipients.

When you select the custom option, a drop-down list appears with the custom data elements that are available for use.

Note: The section Data (see page 225) provides more information about data elements.

Content

You can define the subject and body of an email using simple text, or add them with dynamic content that is calculated when the email is sent.

The subject line is a plain text field where you can write your message. This message is the subject of the email.

The body is displayed in an HTML editor. You can insert and format any text to form the email body.

To include dynamic content, you select options from a drop-down list. The editor adds dynamic content indicators, which resemble the following, where the cursor is located:

{*type*}

type represents one of the supported dynamic content types.

For example, when you select the Attribute dynamic content type and specify the FirstName attribute, the HTML editor displays the following in the Content tab:

{'Attribute: FirstName'}

Note: To add dynamic content to the subject line, use the drop-down list below the subject line. To add dynamic content in the email body, use the drop-down list below the content box.

When the email message is sent, Identity Management replaces the dynamic content with the appropriate text. The text retains the formatting, such as bold characters, specified in the HTML editor.

Dynamic content types include the following:

Date

Specifies today's date in the format you specify.

Task

Specifies the task for which the email is sent.

Object Name

Specifies the name of the object in the event that triggers the email. If the event is a user event, this field is the user login name.

The object can be something other than a user. For example, it can be any managed object such as a group, admin role, and so on.

Attribute

Specifies the value of one of the user attributes. The user is the subject of the task. This option requires selecting the attribute from a drop-down list.

Manager Attribute

Specifies the value of one of the attributes of the user's manager. The user is the subject of the task. This option requires selecting the attribute from a drop down list.

Note: To use the Manager recipient option, configure the manager attribute for the environment. To configure the manager attribute, go to Environments, *EnvironmentName*, Advanced Settings, Miscellaneous in the Management Console. Set managerattribute to the name of the physical attribute that stores the unique name of a user's manager.

For relational databases, specify the attribute using the following format:

tablename.attribute

Custom

Allows you to select a custom data element to define the recipients.

When you select the custom option, a drop-down list appears with the custom data elements that are available for use.

Note: The section Data (see page 225) provides more information about data elements.

Modify Email Notification Policies

You modify an existing email notification policy to suit your business requirements.

To modify an email notification policy

1. Select System, Email, Create Email.

   Identity Management displays a search screen.

2. Search for and select the email notification policy to modify.
3. Change the settings in the Profile, When to Send, Recipients, and Content tabs as needed.

Disable Email Notification Policies

You can enable or disable email notification policies using the Enabled check box on the Profile tab when you create or modify an email notification policy. When an email notification policy is disabled, the selected email is not active and no email is sent.

Note: Email notification policies are enabled by default.

Use Case: Sending a Welcome Email

When a new employee is hired, Forward, Inc, wants to send an email to that user welcoming them to the company. The email must provide important information to the new employee, such as links to the employee home page, and information about their manager and department.

To create the email, the Human Resources administrator uses the Create Email task in the User Console to configure the following settings:

- On the When to Send tab, select User Created.
- On the Recipients tab, complete the following steps:
  - Click the Edit icon next to the To field.

    Select User, then click the plus sign. Select the Manager using the same method, then click OK.
  - Click the Edit icon next to the CC field.

    Select Initiator, click the plus sign, and then click OK to send a copy of the email to the user who created the employee in Identity Management.
- On the Content tab, complete the following steps:
  - In the Subject field, enter the following text: Welcome,

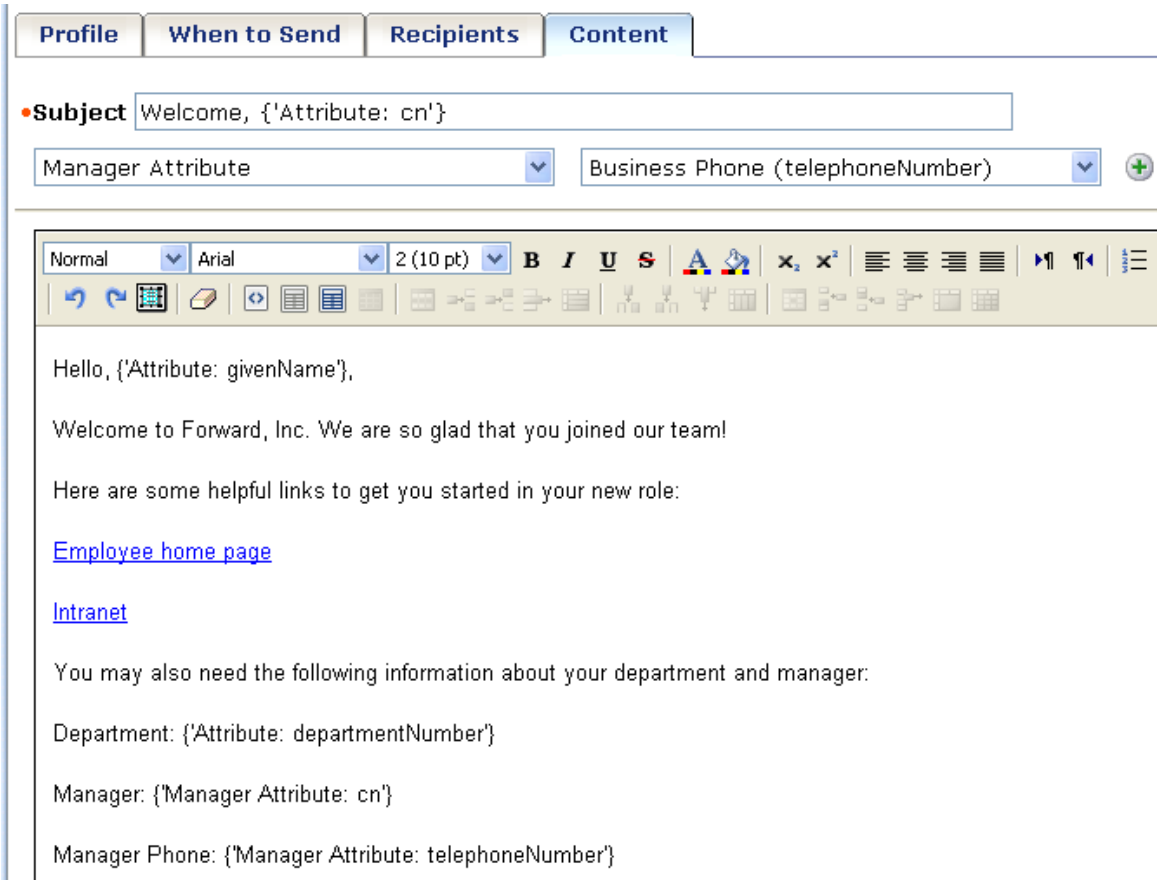
    With the cursor at the end of the text that you entered, select Attribute from the drop-down list. Then, select Full Name from the second drop-down list, then click the plus sign.

    The subject line resembles the following:

    Welcome, {'Attribute: eTFullName'}

    Note: The attribute name depends on the user store and attribute that you are using.
  - In the Content box, add any welcome text. Include links to the Employee portal and use the dynamic content options under the content box to display the user's department, manager, and manager's telephone number, as follows:

Equation 1: Screen shows the Content tab



How to Use Email Templates

Identity Management includes default email templates that you can use to generate email messages. You can use the default templates as installed, or customize them to suit your business needs.

To use email templates

1. Configure SMTP settings to enable Identity Management to send email notifications.
2. Enable email notification in the Management Console (see page 380).
3. Configure an event or task to send an email. (see page 380)
4. (Optional) Customize the default templates (see page 385), as needed.

Enable Email Notification

You can enable or disable email notification for Identity Management environment. If you enable email notifications, Identity Management sends email notifications for events and tasks you specify.

Note: To use the Forgotten Password feature, enable email notification.

Before you enable email notifications in Identity Management, configure the SMTP settings (see page 368) for your application server.

To enable email notifications

1. In the Management Console, click Environments.

   A list of Identity Management environments is displayed.
2. Click the appropriate Identity Management environment.
3. Go to Advanced Settings, Email.
4. Select the Enabled check box.
5. Configure the events and tasks that trigger email (see page 380).
6. Click Save.
7. Restart the instance of the application server on which Identity Management is installed.

Configure an Event or Task To Send Email

If email notifications are enabled, you can specify a list of events and tasks that trigger email notifications. For example, you may want email sent in the following circumstances:

- To a system administrator, at the completion of a Reset User Password task.
- To a new employee's manager, at the completion of a Create User task. In addition, when the AddToGroupEvent generated within the Create User task is approved, another email can be sent to all members of a group to which the new user is being added.

To specify events and tasks that trigger email notifications

1. In the Management Console, click Environments.

   A list of Identity Management environments is displayed.
2. Click the appropriate Identity Management environment.
3. Go to Advanced Settings, Email.

   The Email Properties screen opens.

4. Select the following Enable check boxes that apply:
   - Events E-mail Enabled

     Enables email notification for Identity Management events
   - Tasks Email Enabled

     Enables email notification for Identity Management tasks
5. Enter the location of the email templates that Identity Management uses to create the email messages.

   The email templates are located in a subdirectory in the following location:

   iam_im.ear\custom\emailTemplates

   Note: When you create an email template file with a file name using a different language, the operating system session should be operating in a language that supports the character set.
6. Specify the events for which email notifications are sent as follows:
   - To add an event, select the event in the Event list box, and click Add.

     Identity Management adds the event you selected to the list of events for which email notifications are sent.

     Note: If you select an event that is not associated with a workflow process, Identity Management sends an email notification when the event completes.
   - To delete an event, select the event's check box, then click Delete.
7. Specify the tasks for which email notifications are sent as follows:
   - To add a task, search for the task by selecting a condition in the first field, and entering a task name in the second field. Click Search.

     You can enter a partial task name by using the wildcard (*) character. For example, to search for a Create task, enter Create*.

     Select one or more tasks from the search results. Click Add.

     Note: Task-level email notifications are not available for tasks that have the action type View or Self View. To see the action type of a task, go to Modify Admin Task, Select a Task, and check the action field in the task profile.
   - To delete a task, select the task's check box, then click Delete.

     Deleting a task removes the task from the Task table. It does not delete the task.
8. When you finish configuring the tasks and events that trigger email notifications, click Save.
9. Restart the application server on which Identity Management is installed.

Email Content

Email notifications consists of a generic template plus task-specific details that are added to the email through the email API. For example, the following information can be inserted into an email for a Create User task:

- The name of the administrator who is executing the task
- The name of the new user
- The user's email address, department name, and other attribute data
- The organization where the user is being created
- Workflow approval status and approval time
- The task name and the names of the events in the task

Email Templates

Email notifications are generated from email templates. Identity Management provides default email templates that you can use as installed, or that you can use to create your own email templates.

Each email template contains the following:

- **Delivery information**--A list of email recipients. Identity Management automatically generates the list of recipients, based on users involved in the task. For example, an approval email is sent to all Approvers for the task.
- **Subject--**The text used in the message's subject line.
- **Content**--The message body. The body typically contains both static text and variables, which Identity Management resolves based on the task or event that triggers the email.

The default email templates are located in an emailTemplates directory where the Identity Management administrative tools are installed. The default installation location for the administrative tools is:

- For Windows--C:\Program Files\CA\IAM Suite\Identity Manager\tools\emailtemplates
- For UNIX--<home_directory>/CA/IAM Suite/Identity Manager/tools/emailtemplates

The emailTemplates directory contains five folders. Each folder is associated with a task or event state:

Directory	Contents
Approved	defaultEvent.tmpl--Informs recipients that an event has been approved
Completed	■ CertificationNonCertifiedActionCompletedNotification.tmpl--Informs the manager that a non-compliance action has been applied to an employee. ■ CertificationNonCertifiedActionPendingNotification.tmpl--Informs the manager that a non-compliance action will be applied to an employee. ■ CertificationRequiredFinalNotification.tmpl--Final reminder to a manager that the Certify User task must be completed for an employee. ■ CertificationRequiredNotification.tmpl--Informs the manager that a certification process has begun for an employee. The manager must complete a Certify User task for this employee. ■ CertificationRequiredReminderNotification.tmpl--Reminds the manager that the Certify User task must be completed for an employee. ■ Certify Employee.tmpl--Informs an administrator that the certification process for an employee is complete. ■ CreateProvisioningUserNotificationEvent.tmpl--Sends a temporary password to a user when that user's account is created in the provisioning directory. ■ defaultTask.tmpl--Informs recipients that Identity Management has completed a task. ■ ForgottenPassword.tmpl--Sends a temporary password to users who have used the forgotten password feature. ■ ForgottenUserID.tmpl--Sends a user ID to users who have used the forgotten user ID feature. ■ Self Registration.tmpl--Informs a user that a self-registration task has completed successfully.
Invalid	■ AssignProvisioningRoleEvent.tmpl--Informs recipients that a request to add a user to a provisioning role failed ■ DefaultEvent.tmpl--Informs recipients that an event failed ■ DefaultTask.tmpl--Informs recipients that a Identity Management task failed

Directory	Contents
Pending	■ defaultEvent.tmpl--Informs approvers that a work list item requires attention ■ ModifyUserEvent.tmpl--Same as the default template, but includes methods for retrieving the attributes of the User managed object
Rejected	defaultEvent.tmpl--Informs recipients that an event has been rejected

Use the Identity Management templates and template directory structure that are installed in the <*im_admin_tools_dir*>\Identity Manager\tools\emailTemplates directory as a base for creating custom email templates.

Template Directories

Each template directory described in Email Templates (see page 382) is associated with a particular task or event state. For example, if an email is to be sent for an event that has been rejected in a workflow process, Identity Management looks in a deployed rejected directory for the template to use. Identity Management then generates the email from the appropriate email template in the directory.

Email Templates in a Directory

Each deployed template directory contains one or more email templates. When a task or event occurs for which email is enabled, Identity Management searches the appropriate template directory for a template name that is the same as the name of the task or event. If such a template cannot be found, Identity Management uses the default template in the directory. Default template names are listed in Email Templates (see page 382). For example, Identity Management uses defaultEvent.tmpl in the Pending directory to inform approvers that they have a new work list item.

Sets of Template Directories

A set of template directories contains an approved, completed, pending, and rejected directory. You can deploy multiple sets of template directories and specify one set to be used for a given Identity Management environment.

Email Template Deployment (see page 403) provides information on deploying sets of template directories.

For information on configuring email template directories so that Identity Management uses the correct set for a given environment, see the *Identity Management Configuration Guide*.

Create Email Templates

To create custom email messages

1. Open the template that you want to modify.

   For example, if you want to create an email message for a pending Create User event, open defaultEvent.tmpl in the Pending directory.

2. Save the template in the same directory with a new name. The name must match the name of the event to which the email applies, and have the extension .tmpl.

   For example, name the message for the pending Create User event as follows:

   CreateUserEvent.tmpl

   Note: When you create an email template file with a file name using a different language, the operating system session should be operating in a language that supports the character set.

3. Modify the message template as needed, as described in the next section, Custom Email Templates (see page 385).

Custom Email Templates

An email template is a dynamic file that supports both HTML and embedded server-side JavaScript. A template lets you insert variable values into static text, allowing case-specific messages to be generated from a single template.

The same template can be used any number of times to print out boilerplate static text (such as the phrase has been approved) along with variable text specific to a given context (such as the name of the event being approved).

For example, here is a template for reporting the approval of an event:

```
<!-- Define the E-mail Properties --->
<%
   _to = _util.getNotifiers("ADMIN");
   _cc = "" ;
   _bcc = "";
   _subject = _eventContextInformation.getEventName() + " approved";
%>
<!--- Start of Body --->
<html>
<body text="Navy">
```

```
Event: <b> <%=_eventContextInformation.getEventName()%> </b><br>
<%=_eventContextInformation.getPrimaryObjectTypeName()%>:
<b><%=_eventContextInformation.getPrimaryObjectName()%></b><br>
In <%=_eventContextInformation.getSecondaryObjectTypeName()%>:
<b><%=_eventContextInformation.getSecondaryObjectName()%></b><br>
Status: <b>Approved</b>
</body>
</html>
```

Note: The Identity Management objects _util and _eventContextInformation used in the above example are described in Email Template API (see page 388).

If an approval is generated for the event CreateUserEvent, and user John Jones is created in organization HR, the body of the email notification generated from the approval template might look like this:

Event: **CreateUserEvent**
USER: **John Jones**
In ORGANIZATION: **HR**
Status: **Approved**

The following sections describe the syntax and Identity Management objects that make dynamic email messages possible.

Template Elements

Identity Management email templates support:

- Standard HTML tags.
- Server-side JavaScript.
- One or more implicit objects that Identity Management makes available to an instance of the template--that is, to an email message.
- Identity Management tags that let you embed JavaScript in the template, call the methods in the implicit Identity Management objects, and insert variable values into the template's static text.

Identity Management Tag Extensions

Email templates support the following tags:

<% %>

Embeds JavaScript into an email template.

<%= %>

Inserts a variable value into static text.

The tags are described in the following sections.

<% %>

This tag lets you embed JavaScript for in-line execution into an email template.

You can use any JavaScript object within the embedded JavaScript. You can also call Identity Management implicit object methods within the embedded JavaScript.

For example, the following code modifies the body of the approval template shown in Custom Email Templates (see page 385). JavaScript is used to determine if a secondary object is involved in the event (such as an ORGANIZATION object when a USER primary object is added). If there is no secondary object, the text relating to the secondary object is omitted from the message:

```
Event: <b> <%=_eventContextInformation.getEventName()%> </b><br>
<%=_eventContextInformation.getPrimaryObjectTypeName()%>:
<b><%=_eventContextInformation.getPrimaryObjectName()%></b><br>
<%
var secondaryType =        _eventContextInformation.getSecondaryObjectTypeName();
if (secondaryType != "") {
   template.add("In " + secondaryType + ": ");
   template.add("<b> "+_eventContextInformation.getSecondary
                                    ObjectName()+" </b><br>");
   }
%>
Status: <b>Approved</b>
```

<%= %>

This tag lets you insert a variable value into static text. The value can be:

- A variable defined in some previously executed JavaScript in the template--for example:

  ```
  <%
  var secondaryType
  =      _eventContextInformation.getSecondaryObjectTypeName();
  ...           // More JavaScript processing
  %>
  ...           // More HTML
  The primary object was created in <%=secondaryType%>.
  ```

- A value returned from a method in a Identity Management implicit object--for example:

  ```
  Event <%=_eventContextInformation.getEventName()%> is approved.
  ```

Email Template API

When a message is generated from a template, Identity Management makes the implicit objects below available to the message. These objects let you insert instance-specific information into a message by calling methods in the Email Template API.

A template can call the methods in any of the following objects:

- _contentType. Specifies the contentType for the email.
- _priority. Specifies the priority for the email.
- _to. Adds recipients to the message's To field.
- _cc. Adds recipients to the message's cc (send copy to) field.
- _bcc. Adds recipients to the message's bcc (send blind copy to) field.
- _subject. Specifies the subject of the email.
- _encoding. Specifies the encoding for the email.
- _additionalHeaders. Allows you to specify extra email header attributes in the email template.
- template. Lets you add a string of text to a message from lines of JavaScript code.
- _util. A utility object.
- _eventContextInformation. Contains information about the event generated by the current task, such as event name and approval status.
- _taskContextInformation. Contains a collection of information about the current task, such as task name, organization name, and constituent events.

These objects are described in the following sections.

contentType

Specifies the contentType for the email.

If no contentType is specified through _contentType variable, the default contentType "text/html" applies.

Methods: None.

Example:

```
<% _contentType = "text/html"; %>
```

priority

Specifies the priority for the email. Specify 0 for no priority (default) and 1 for high priority.

Methods: None.

Example:

```
<% _priority = "1"; %>
```

to

Adds recipients to the message's To field.

The value of the _to variable is a JavaScript string. Multiple recipients are permitted, but the string must conform to JavaScript syntax, as shown in the following example.

Methods: None.

Example:

```
<%
_to =
_util.getNotifiers("USER") + ',' +
_util.getNotifiers("USER_MANAGER","ManagerLookup=managerattribute");
_cc = "" ;
_bcc = "" ;
_subject = "Your new password ";
%>
```

Note: When emails alert participants that a task is in a Pending state and under workflow control, the _to object is pre-populated with the addresses of the participants. You cannot use the _to object in a Pending template.

cc

Adds recipients to the message's cc (send copy to) field.

The value of the _to variable is a JavaScript string. Multiple recipients are permitted, but the string must conform to JavaScript syntax, as shown in the following example.

Methods: None.

Example:

```
<%
_cc =
_util.getNotifiers("USER") + ',' +
_util.getNotifiers("USER_MANAGER","ManagerLookup=managerattribute");
%>
```

bcc

Adds recipients to the message's bcc (send blind copy to) field.

Email addresses specified in this field do not appear in the email.

The value of the _to variable is a JavaScript string. Multiple recipients are permitted, but the string must conform to JavaScript syntax, as shown in the following example.

Methods: None.

Example:

```
<%
_bcc =
_util.getNotifiers("USER") + ',' +
_util.getNotifiers("USER_MANAGER","ManagerLookup=managerattribute");
%>
```

subject

Specifies the subject of the email.

Methods: None.

Example:

```
<% _subject=_eventContextInformation.getEventName()+" approved";%>
```

encoding

Specifies the encoding for the email.

If no encoding is specified either through _encoding or through the LANG variable, characters in the email may not be correctly displayed. Be sure to set _encoding or LANG for the appropriate locale.

Methods: None.

Example:

```
<% _encoding = "UTF-8"; %>
```

additionalHeaders

_additionalHeaders

Specifies extra email header attributes in the email template.

You must assign a HashMap() to this attribute. The names and values stored in the HashMap must be strings.

Example: Add custom header attributes

The following example shows you how to add two custom header attributes, “X-TCCCSWD” and “myheader”:

```
<!-- Define the E-mail Properties --->
<%
_to = "siteadmin@ca.com";
_cc = "" ;
_bcc = "" ;
_subject = _eventContextInformation.getEventName() +" completed";
var additionalHeaders = new java.util.HashMap();
additionalHeaders.put("header_a","1");
additionalHeaders.put("header_b","foo");
_additionalHeaders = additionalHeaders;
%>
```

template

Lets you add a string of text to a message from lines of JavaScript code (that is, lines within the <% %> tag). The string can contain HTML tags, static text, and/or variable values returned by methods in Identity Management implicit objects.

Note: The template object is not preceded by the underscore (_) character.

Method:

- add(String)

  The argument must evaluate to a string, including any calls to methods in a Identity Management implicit object. In the example below, see _eventContextInformation.getSecondaryObjectName().

Example:

```
<%
var secondaryType =        _eventContextInformation.getSecondaryObjectTypeName();
if (secondaryType != "") {
   template.add("In " + secondaryType + ": ");
   template.add("<b> "+_eventContextInformation.getSecondary
                                    ObjectName()+" </b><br>");
   }
%>
```

util

Utility object.

Method:

- getNotifiers(String [,String])

  Returns email IDs based on a notification rule.

  The first argument supports the following predefined notification rules, enclosed in quotes:

  - "ADMIN". Sends the email to the administrator who initiated the task.
  - "USER". Sends the email to the user in the current context.
  - "USER_MANAGER". Sends the email to the manager of the user in the current context.

  You can also reference a custom notification rule that you create with the Notification Rule API. For information, see the *Programming Guide for Java*.

  The second argument is optional. You can use it to pass one or more user-defined name/value pairs into a custom notification rule. Separate each name/value pair with a comma, in the following format:

  `"name1=value1,name2=value2,..."`

Examples:

```
<%
_to = _util.getNotifiers("ADMIN");
_cc = "";
%>
<%
_to = _util.getNotifiers("MYRULE","type=loan,district=3");
_cc = "";
%>
```

Notifying a User's Manager

You can use the USER_MANAGER notification rule to send email to any user's manager. Identity Management uses this rule in the email templates supporting user entitlement certification.

Note: The USER_MANAGER Notification Rule only applies to events or tasks that create or manage a single user.

Because there are a number of different ways a user-to-manager relationship can be specified within a user directory, the default User Manager Notification Adapter resolves this relationship based on an attribute expression specified in the second parameter of the getNotifiers() method.

Example:

```
<%
_to = _util.getNotifiers("USER_MANAGER","ManagerLookup=managerattribute");
_cc = "";
%>
```

The User Manager Notification Adapter supports two look-up options:

- managerattribute **= <***Manager AttributeName***>**- where the User object maintains an attribute that indicates the DN or UserID of that user's manager
- commonattribute **= <***AttributeName***>** - where the user and the user's manager share a common attribute value, such as "department"

You configure these lookup options in the Miscellaneous Properties for an environment in the Identity Management Management Console.

To configure the USER_MANAGER notification rule:

1. In the Identity Management Management Console, select Identity Management Environments. Then, select the environment for which you are configuring email notification.
2. Select Advanced Settings>Miscellaneous Properties.

3. In the Miscellaneous Properties page, complete the configuration steps for the lookup option that you want to use:
   - To use the managerattribute=<Manager AttributeName> lookup option:
     a. In the Property field, enter managerattribute.
     b. In the Value field, enter the attribute that stores the manager's DN or user ID.
     c. Click Add.
     d. Click Save.
   - To use the commonattribute=<AttributeName> lookup option:
     a. In the Property field, enter commonattribute.
     b. In the Value field, enter the attribute that the user and the user's manager have in common.
     c. Click Add.
     d. In the Property field, enter ismanagerfilter.
     e. In the Value field, enter a search expression using the following syntax:

        <attribute> <operator> <filter>

        For example, title EQUALS manager
     f. Click Add.
     g. Click Save.

You can also write a custom adapter and create your own rules for notifying a user's manager. See the *Programming Guide for Java*.

eventContextInformation

Contains information about the event generated by the current task, such as event name and approval status. This information is called *context* information for the event.

The _eventContextInformation object is created from the ExposedEventContextInformation class in package com.netegrity.imapi.

This object is available for email messages based on Approved, Pending, and Rejected templates. For information about these templates, see Email Templates (see page 382).

Methods: All the following methods return a String.

Method	Description
getAdminName()	Returns the name of the person who submitted the task that generated the event. Deprecated in Identity Management 5.6. Use one of the following inherited methods: ■ getAdministrator() ■ getAdminFriendlyName()
getApprovalStatus()	Returns the approval status of the event. One of these values: APPROVAL_STATUS_APPROVED APPROVAL_STATUS_REJECTED
getApprovalTime()	Returns the time the event was approved.
getEventName()	Returns the name of the event. For a list of event names, see Identity Management Events.
getOrgName()	Returns the friendly name of the organization where the task is being executed. Deprecated in Identity Management 5.6. Use the inherited method getObjectOrganizationFriendlyName().
getPassword()	If the primary objects is type USER, returns the user's password.
getPrimaryObjectTypeName()	Returns the type of primary object. Primary object types returned: ACCESSROLE ACCESSTASK ADMINROLE ADMINTASK GROUP ORGANIZATION USER

Method	Description
getPrimaryObjectName()	Returns the name of the primary object affected by the event. A *primary object* is the object directly affected by the event. A *secondary object* is the object that the primary object is bound to, if any. For example: ■ The primary object type for CreateUserEvent is USER. The secondary object is the object where the user is created--that is, ORGANIZATION. ■ The primary object type for CreateAdminRoleEvent is ADMINROLE. This object is not bound to other objects, so no secondary object exists. With a primary object of type USER, getPrimaryObjectName() might return John Jones.
getSecondaryObjectTypeName()	If a secondary object was affected by the event, returns the object type. Secondary object types returned: ACCESSROLE ACCESSTASK ADMINROLE ADMINTASK GROUP ORGANIZATION USER
getSecondaryObjectName()	If a secondary object was affected by the event, returns the object name. See getPrimaryObjectName() for information about primary and secondary objects. With a secondary object of type ORGANIZATION, the method getSecondaryObjectName() might return HR.

Note: The methods in _eventContextInformation are provided through the interface ExposedEventContextInformation. Since ExposedEventContextInformation inherits methods in the core Identity Management API, _eventContextInformation can also call these methods from an email template, along with the methods in the above table. For more information about these inherited methods, see Additional Methods (see page 400).

Example--Email notification about a Pending event:

```
<%
_cc = "" ;
_bcc = "";
_subject = _eventContextInformation.getEventName() +
                                                " Approval Request";
%>
<!--- Start of Body --->
<html>
<body text="Navy">

The following item has been added to your work list for approval:
<br><br><br>
Event: <b><%=_eventContextInformation.getEventName()%></b> <br>
<%=_eventContextInformation.getPrimaryObjectTypeName()%>:
<b><%=_eventContextInformation.getPrimaryObjectName()%></b><br>
In <%=_eventContextInformation.getSecondaryObjectTypeName()%>:
<b><%=_eventContextInformation.getSecondaryObjectName()%></b><br>
</body>
</html>
```

Possible email body:

From: lsmith@security.com [mailto:lsmith@security.com]
To: vimperioso@security.com
Subject: CreateUserEvent Approval Request

The following item has been added to your work list for approval:

Event: **CreateUserEvent**
USER: **Richard Ferrigamo**
In ORGANIZATION: **Mortgages & Loans**

Note: The value of the From field is derived from the email.properties file. To change the value, edit the following file:

<*iam_im.ear*>\config\com\netegrity\config\email.properties

where <*iam_im.ear*> is the installed location of Identity Management in the application server domain--for example:

For WebLogic:

<*WebLogic_home*>\user_projects\<domain>\applications\iam_im.ear

For JBoss:
<*Identity Manager_home*>\jboss-3.2.2\server\default\deploy\iam_im.ear

For WebSphere:
<*im_admin_tools_dir* >\WebSphere-ear\iam_im.ear

To add additional information about the user affected by the event to the email in the previous example, add text that resembles the following:

```
<% user = _eventContextInformation.getEvent().getUser(); %>
<b>User information:</b><br>
Last Name: <b><%=user.getAttribute("%LAST_NAME%")%></b><br>
First Name: <b><%=user.getAttribute("%FIRST_NAME%")%></b><br>
Full Name: <b><%=user.getAttribute("%FULL_NAME%")%></b><br>
Email: <b><%=user.getAttribute("%EMAIL%")%></b><br>
Organization Membership: <b><%=user.getAttribute("%ORG_MEMBERSHIP%")%></b><br>
```

Possible email body:

From: lsmith@security.com [mailto:lsmith@security.com]
To: vimperioso@security.com
Subject: CreateUserEvent Approval Request

The following item has been added to your work list for approval:

Event: **CreateUserEvent**
USER: **Richard Ferrigamo**
In ORGANIZATION: **Mortgages & Loans**
User information:
Last Name: Ferrigamo
First Name: Richard
Full Name: Richard Ferrigamo
Email: rferrigamo@mybank.org
Organization Membership: **Mortgages & Loans**

taskContextInformation

Contains a collection of information about the current task, such as task name, organization name, and constituent events. This information is called *context* information for the task.

This object is available for email messages based on Completed templates. For information about this template, see Email Templates (see page 382).

Methods: All the methods below return a String except for the method getExposedEventContexts(), which returns a Java Vector.

Method	Description
getAdminName()	Returns the name of the person submitting the task. Deprecated in Identity Management 5.6. Use one of the following inherited methods: ■ getAdministrator() ■ getAdminFriendlyName()
getExposedEventContexts()	Returns a Java Vector of all events associated with the task. Each object in the Vector is an event context object. You can use the methods listed in _eventContextInformation to retrieve context information for a given event object. The return object is a standard Java Vector object. You can use any of the Vector object's methods--for example, get() and size()--to manage the elements in the Vector.
getOrgName()	Returns the name of the organization where the task is being executed. Deprecated in Identity Management 5.6. Use the inherited method getObjectOrganizationFriendlyName().
getTaskName()	Returns the name of the task being executed. Deprecated in Identity Management 5.6. Use one of the following inherited methods: ■ getAdminTask() ■ getTaskFriendlyName()

Note: The methods in _taskContextInformation are provided through the interface ExposedTaskContextInformation. Since ExposedTaskContextInformation inherits methods in the core Identity Management API, _taskContextInformation can also call these methods from an email template, along with the methods in the above table. For more information about these inherited methods, see Additional Methods (see page 400).

Example--Body of an email notification template for a password change:

```
<%
var imsEventContexts
=              _taskContextInformation.getExposedEventContexts();
if(imsEventContexts != null)
   {
   for(var i=0;i<imsEventContexts.size();i++)
      {
      var eventContext = imsEventContexts.get(i);
      template.add("Hi "+
eventContext.getPrimaryObjectName()
           + ",");
      template.add("<br>Your new password is: <b>"+
                              eventContext.getPassword());</br>
      template.add("<hr>");
      }
   }
%>
```

Possible email body:

Hi Victor Imperioso,
Your new password is: LFH7F1226

Additional Methods

The methods in _taskContextInformation and _eventContextInformation are provided through the ExposedTaskContextInformation and ExposedEventContextInformation, respectively.

These objects inherit methods in the core Identity Management API. Consequently, the inherited methods are also available to _taskContextInformation and _eventContextInformation.

The following methods inherited from the TaskInfo object are particularly useful to an email template:

- getAdministrator(). Retrieves a User object for the administrator who is executing the current task.
- getAdminTask(). Retrieves an AdminTask object for the current task.

These retrieved objects allow you to insert administrator-specific and task-specific information into an email. For example:

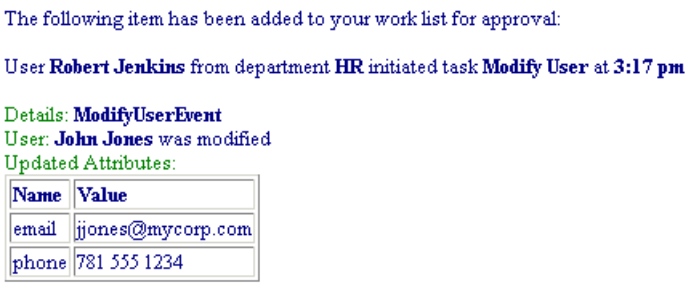
```
<!-- Define the E-mail Properties --->

<%
   _cc = "" ;
   _bcc = "" ;
   _subject = _eventContextInformation.getEventName() +
                                                 " Approval Request";
%>
<!--- Start of Body --->
<html>
<body text="Navy">

The following item has been added to your work list for approval:<br>
<br>
User <b><%= _eventContextInformation.getAdministrator().
               getAttribute(Packages.com.netegrity.llsdk6.imsapi.
                managedobject.User.PROPERTY_FRIENDLY_NAME)%> </b>
               from department <b><%= _eventContextInformation.
               getAdministrator().getOrg(null).getFriendlyName()
               %></b> initiated task <b><%= _eventContextInformation.
               getAdminTask().getFriendlyName() %></b>at
<b><%=                 _eventContextInformation.getSessionCreateTime() %></b>
<br><br>
<font color="green">Details: </font><b><%=_eventContextInformation.
                                          getEventName()%></b><br>
<font color="green"><%=_eventContextInformation.
                               getPrimaryObjectTypeName()%>:</font>
<b><%=_eventContextInformation.getPrimaryObjectName()%></b>
                                                        was modified
<br>
<font color="green">Updated Attributes:</font>
<table border="1">
<tr>
  <td><b>Name</b></td>
  <td><b>Value</b></td>
</tr>
```

```
<%
   var event = _eventContextInformation.getEvent();
   if(event instanceof Packages.com.netegrity.imapi.UserEvent) {
      var user = event.getUser();
      var attributes = user.getAttributes().keys();
      while(attributes.hasMoreElements()) {
         var attr = attributes.nextElement();
         var value = user.getAttribute(attr);
         if(user.hasAttributeChanged(attr)) {
            template.add("<tr><td>" + attr +"</td>");
            template.add("<td>" + value +"</td></tr>");
         }
      }
   }
%>
</table>
<br>
</body>
</html>
```

Possible email body:



For more information about the inherited methods that are available to the Email Template API, see the objects ExposedTaskContextInformation and ExposedEventContextInformation objects in the Identity Management Javadoc.

Java Standard Output Stream

An email message can also make calls to the Java standard output stream from inside the JavaScript tag (<% %>). For example, the following call sends the message Done to the server console:

```
<%
...       // JavaScript processing
out.println("Done.");
%>
```

Javadoc Reference

For information about the ExposedTaskContextInformation and ExposedEventContextInformation objects, including the methods they inherit from the core Identity Management API, see the Identity Management Javadoc.

The Javadoc pages are integrated with an HTML version of the Programming Guide for Java, which is available in the Identity Management Bookshelf.

Email Template Deployment

When Identity Management is about to send email, it searches for templates from which to generate the email in the following root location within your application server:

iam_im.ear\custom\emailTemplates

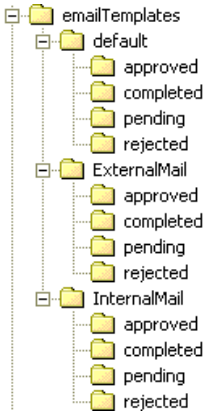
The email templates deployed in this root are contained in template sets that have the same directory structure--that is, there is an approved, completed, pending, and rejected directory in each set.

Template Sets

You can deploy several sets of email templates under emailTemplates. For example, during installation, the following set of email templates is created under iam_im.ear\custom\emailTemplates:

default\approved
default\completed
default\pending
default\rejected

The default email template set contains the installed templates that are described in Email Templates (see page 382). You can add custom templates within the default set. You can also deploy other sets of email templates in directory structures that you define at the same level as the default set. For example, iam_im.ear\custom might contain the following deployed email templates:



Note: For information about how Identity Management chooses a particular email template within a template set, see Template Directories (see page 384).

How to Specify a Template Set for an Environment

When you configure email for a Identity Management environment, you specify the email template set that you want to use for that environment. For information about configuring email for a Identity Management environment, see the *Identity Management Configuration Guide*.

Template Names

The directories in a custom template set should contain default templates with the same name as those that were installed in the default template set. The default names are listed in Email Templates (see page 382). Identity Management uses the default templates when it can find no other template with a name that matches the task or event being executed.

Optionally, you can add additional templates to one or more directories in a template set if you want an email to be generated from a particular template. To do so:

- Assign the template the same name as the task or event for which the email will be generated.
- Place the template in the directory associated with the task or event state for which the email will be generated.

For example, if you want emails to be generated from a particular template when a CreateUserEvent is rejected, place a template named CreateUserEvent.tmpl in the rejected directory of the environment's template set.

Chapter 14: System Tasks

This section contains the following topics:

Task Status in Identity Management

Administrators may want to track the status of Identity Management tasks once they are submitted for processing. Identity Management provides the following methods for viewing task status:

- **View Submitted Tasks Tab**

  This tab allows you to search for and display Identity Management tasks that have been submitted for processing.

  Administrators can view task details at a high level or view additional levels of detail.

  The View Submitted Tasks tab is included in two default tasks:

  - View My Submitted Tasks

    Allows administrators to search for and display information about tasks that they submitted for processing.

  - View Submitted Tasks

    Allows administrators to search for and display information about tasks that other administrators have submitted for processing.

- **User History Tab**

  This tab, which you can add to user tasks, such as View or Modify User, lets administrators view the following information for a selected user:

  - Tasks performed on the user
  - Tasks performed by the user
  - Workflow approvals by the user

- **Reports**

  Identity Management reports enable you to see the current state of a Identity Management environment. You can use this information to ensure compliance with internal business policies or external regulations.

  Reporting (see page 235) provides additional information about setting up and using reports.

- **Logs**

  Display information about all of the components in a Identity Management installation, and provide details about all operations in Identity Management.

  See the *Configuration Guide* for more information about Identity Management logs.

How Identity Management Determines Task Status

A *task* is an administrative function that a user can perform in Identity Management. Tasks include *events*, actions that Identity Management performs to complete the task. A task may include multiple events. For example, the Create User task may include events that create the user's profile, adds the user to a group, and assigns roles.

Identity Management tasks and events can be associated with a workflow process, which determines how Identity Management performs the required actions, and other custom business logic. Tasks may also be associated with other tasks, called nested tasks. In this case, Identity Management processes the nested tasks with the original task.

The status of a task depends on the status of its associated events, workflow processes, nested tasks, and custom business logic.

View Submitted Tasks

Identity Management includes a View Submitted Tasks feature that provides information about tasks in a Identity Management environment. You can use this feature to search for and view high-level details about actions that Identity Management performs. Detail screens provide additional information about each task and event.

Depending on the status of the task, you can use View Submitted Tasks to abort or resubmit a task.

View Submitted Tasks allows you to track the processing of a task from beginning to end. For example, if a Identity Management task includes provisioning role assignment, and that assignment triggers the creation of accounts in other systems, The View Submitted Tasks tab displays all of the details of the original task and the details of the account creations.

View Submitted Tasks includes details of operations performed in the system. These operations could be the result of a Identity Management event, such as EnableUserEvent. The notifications sent by the system are grouped under this event. View Submitted Tasks displays a message indicating the notifications are In Progress until the End Detail notification is sent. Then, the message changes to Completed.

By default, Identity Management includes the View Submitted Tasks tab in two tasks:

- View Submitted Tasks
- View My Submitted Tasks

Search for Submitted Tasks

Perform the following steps to search for submitted tasks.

To search for submitted tasks

1. Navigate to System, View Submitted Tasks.

   The View Submitted Tasks page appears.

2. Specify search criteria, enter the number of rows to be displayed, and click Search.

   The tasks that satisfy your search criteria are displayed.

   Note: For more information on specifying attributes in the search criteria, see Search Attributes for Viewing Submitted Tasks (see page 410).

Search Attributes for Viewing Submitted Tasks

To review tasks that have been submitted for processing, you can use the search feature in View Submitted Tasks. You can search for tasks based on the following criteria:

Initiated By

Identifies the name of the user who has initiated a task as the search criteria. Searches are based on the user name. To ensure that you entered a valid user name, use the Validate button.

Approval Tasks Performed By

Identifies the name of the task approver as the search criteria. Searches are based on the user name. To ensure that you entered a valid user name, use the Validate button.

Note: If you select Approval Tasks Performed By criteria to filter the tasks, the Show approval tasks criteria is also enabled by default.

Task Name

Identifies the task name as the search criteria. You can refine the search by specifying conditions such as equals, contains, starts with, or ends with the value of the Where Task Name field. For example, you can specify the search criteria "task name equals Create User" by selecting the equals condition, and entering Create User in the text field.

Task Status

Identifies task status as the search criteria. You can select the task status by enabling Where task status equals, and selecting the condition. You can further refine the search based on the following conditions:

- Completed
- In Progress
- Failed
- Rejected
- Partially Completed
- Aborted
- Scheduled

Note: See Task Status Description (see page 412) for more information.

Task Priority

Identifies task priority as the search criteria. You can select the task priority by enabling Where task priority equals, and selecting the condition. You can further refine the search based on the following conditions:

Low

Specifies that you can search for tasks that have a low priority.

Medium

Specifies that you can search for tasks that have a medium priority.

High

Specifies that you can search for tasks that have a high priority.

Performed On

Identifies tasks that are performed on the selected instance of the object. If you do not select an instance of the object, the tasks that were performed on all the instances of that object will be displayed.

Note: This field appears only when the Configure Performed On field is populated in the Configure Submitted Tasks screen. You use this screen to configure the Submitted Tasks tab. See the online help for that screen for more information.

Date range

Identifies the dates between which you want to search submitted tasks. You must provide the From and To dates.

Show unsubmitted tasks

Identifies the tasks in the Audited state. Identifies the tasks that have initiated other tasks or tasks that have not been submitted. All such tasks will be audited and displayed if you select this tab.

Show approval tasks

Identifies the tasks that have to be approved as part of a workflow.

Search archive of submitted tasks

Identifies the submitted tasks that have been archived.

More Information:

Task Status Description (see page 412)

Task Status Description

A submitted task exists in one of the states described below. Based on the state of the task, you can perform actions such as cancelling or resubmitting a task.

Note: To cancel or resubmit a task, you must configure View Submitted Tasks to display the cancel and resubmit buttons based on the tasks status. For more information on cancelling and resubmitting tasks, see Customize the View Submitted Tasks Tab (see page 415).

In progress

Displayed when any of the following occurs:

- Workflow is initiated but not yet completed
- Tasks, which are initiated before the current tasks, are in progress
- Nested tasks are initiated but not yet completed
- The primary event is initiated but not yet completed
- Secondary events are initiated but not yet completed

You can cancel a task in this state.

Note: Cancelling a task will cancel all the incomplete nested tasks and events for the current task.

Cancelled

Displayed when you cancel any of the tasks or events in progress.

Rejected

Displayed when Identity Management rejects an event or task that is part of a workflow process. You can resubmit a rejected task.

Note: When you resubmit a task, Identity Management will resubmit all the failed or rejected nested tasks and events.

Partially Completed

Displayed when you cancel some of the events or nested tasks. You can resubmit a partially completed event or nested task.

Completed

Displayed when a task is completed. A task is completed when the nested tasks and nested events of the current task are completed.

Failed

Displayed when a task, nested task, or event nested in the current task are invalid. This status is displayed when the task fails. You can resubmit a failed task.

Scheduled

Displayed when the task is scheduled to execute at a later date. You can cancel a task in this state.

Audited

Displayed when the current task is audited.

View Task Details

Identity Management provides task details, such as the status of a submitted task, nested tasks, and events associated with a task.

To view details of a submitted task

1. Click the right arrow icon next to the selected task in the View Submitted Tasks tab.

   The task details appear.

   Note: Events and nested tasks (if any) are displayed in the Task Details page. You can view the task details for each of the tasks and events.

2. Click Close.

   The Task Details tab closes and Identity Management displays the View Submitted Tasks tab with the tasks list.

View Event Details

Identity Management provides events details, such as the status of a submitted event, event attributes, and any additional information about the events.

To view details of a submitted event

1. Click the right arrow icon next to an event in the View Task Details page.

   The event details appear.

2. Click Close.

   The Event Details page is closed.

Description of Event Status

Events in Identity Management can be in one of the states described below. Based on the event status, you can cancel or resubmit an event for execution.

Note: To allow administrators to cancel or resubmit an event, you must configure View Submitted Tasks to display the Cancel and Resubmit Events buttons. When you configure the task, you can specify which administrators can cancel and resubmit events. For more information on cancelling and resubmitting events, see Customize the View Submitted Tasks Tab (see page 415).

In progress

Displayed when any of the following occurs:

- Workflow or pre-events are initiated, in progress, or approved
- Identity Management is executing the event
- Identity Management executes post events

You can cancel an event in this state.

Rejected

Displayed when Identity Management rejects an event that is part of the workflow. You can resubmit a rejected event.

Cancelled

Displayed when you cancel any of the events in progress. You can resubmit a cancelled event.

Completed

Displayed when an event is completed.

Failed

Displayed when Identity Management encounters an exception during execution of an event. You can resubmit a cancelled event.

Note: You cannot resubmit a secondary event until the primary event is in the completed state.

Scheduled

Displayed when the event is scheduled to execute at a later date. You can cancel an event in this state.

Audited

Displayed when the current event is audited.

Customize the View Submitted Tasks Tab

You can customize the View Submitted Tasks tab as follows:

- Specify a different task name and tag.
- Change the default display properties. As installed, users see a search screen where they can enter criteria that determine the tasks that appear in the tab. You can configure the tab to automatically display the submitted tasks for a current day, preventing users from having to enter search criteria.
- Determine whether audit events appear in the Task Details page.
- Add an additional column to the task display.
- Specify the criteria for cancelling or resubmitting tasks and events.

Note: Certain task and event details may include data, such as salaries or passwords, that should not be displayed in clear text in the View Submitted Tasks tab. You can hide those attributes by specifying data classification parameters when you define the attributes in the directory.xml file. For more information about the directory.xml file, see the *Configuration Guide*.

You can configure the View Submitted Tasks tab by modifying the corresponding admin task.

To configure the View Submitted Tasks tab

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

   The Select Admin Task page appears.

2. Select Name or Category in the Search Admin Task where field, enter the string you want to search, and click Search.

   Identity Management displays the admin tasks that satisfy the search criteria.

3. Select View Submitted Tasks, and click Select.

   Identity Management displays the task details for the View Submitted Tasks admin task.

4. Click the Tabs tab.

   The tabs that are used for View Submitted Tasks tab are displayed.

5. Click the right arrow icon to edit the Submitted Tasks tab.

   The tab details appear.

6. Edit the fields to customize the View Submitted Tasks tab as needed. See Configuration Settings for the Submitted Tasks tab (see page 416).

Configuration Settings for the View Submitted Tasks Tab

Use the following fields to change the appearance and functionality of the View Submitted Tasks tab.

Name

Defines the name of the task.

Tag

Defines a unique identifier for the task. It is used in URLs, web services or properties files. It must consist of letters, numbers, or underscores, beginning with a letter or underscore.

Hide Tab

Identifies that the tab is visible to the users, but will not be executed. If you select this option, Identity Management will display an error to the users.

Show task lists on load

Displays the tasks that have been submitted for the current day.

Note: If you have enabled this option, users clicking on View Submitted Tasks will directly see the tasks that were submitted on the same day.

Show audit events

Specifies if audited events are included in tasks in the View Submitted Tasks page.

Allow custom column

Indicates that you can append a custom column to the tasks table that you can view from the View Submitted Tasks tab and the User History tab. For example, you can append a column "User ID" to the tasks table that is displayed on the User History tab.

Custom column heading

Indicates the display name of the custom column.

Custom column attribute

Indicates the attribute that will be used to populate the custom column in the tasks table. For example, if you are searching for tasks that are performed on employees of an organization, you can append an organization column that displays the organization for each of the employees.

Cancel Tasks and Events

Identifies the criteria for canceling tasks or events. You can set the scope for this field by selecting one of the following options:

Task creator must be current user

Identifies that you can cancel or resubmit tasks or events that you have created.

Task creator must be in scope

Identifies that you can cancel or resubmit tasks that have been initiated by other users who match the user scope rules for the admin role that gives you access to the tab.

For example, you received the User Manager role, which includes View Submitted Tasks, because you met the criteria in a membership rule that includes scope over all users in the Employee organization. You can cancel or resubmit tasks that are submitted by all users in the Employee organization.

No restrictions

Identifies that any user can cancel or resubmit a task or event.

Not allowed

Specifies that a task or event cannot be cancelled or resubmitted.

Resubmit Tasks and Events

Identifies the criteria for resubmitting a task or event. You can set the scope of this field by selecting one of the following options:

Task creator must be current user

Identifies that you can cancel or resubmit tasks or events that you have created.

Task creator must be in scope

Identifies that you can cancel or resubmit tasks that have been initiated by other users who match the user scope rules for the admin role that gives you access to the tab.

For example, you received the User Manager role, which includes View Submitted Tasks, because you met the criteria in a membership rule that includes scope over all users in the Employee organization. You can cancel or resubmit tasks that are submitted by all users in the Employee organization.

No restrictions

Identifies that any user can cancel or resubmit a task or event.

Not allowed

Specifies that a task or event cannot be cancelled or resubmitted.

User History Tab

The User History tab allows you to view tasks that are related to a user. The task details that are displayed in this tab can also be viewed in the View Submitted Tasks tab.

Note: You cannot add this tab to create tasks, such as Create User.

You can use this tab to view a history of the following tasks:

- **Tasks performed on the user**

  Displays all the tasks that are performed on the selected user.

- **Tasks performed by the user**

  Displays all the tasks that are performed by the selected user.

- **Workflow approvals by the user**

  Displays all the tasks that the user has approved as part of a workflow.

Note: The type of tasks that you can view in this tab depend on the tab's configuration. Customize the User History Tab (see page 419) provides more information.

Search Attributes for Viewing User History

To review tasks that have been submitted for processing, you can use the search feature in View Submitted Tasks. You can search for tasks based on the following criteria:

Task Name

Identifies the task name as the search criteria. You can refine the search by specifying conditions such as equals, contains, starts with, or ends with the value of the Where Task Name field. For example, you can specify the search criteria, task name equals Create User by selecting the equals condition, and entering Create User in the text field.

Task Status

Identifies task status as the search criteria. You can select the task status by enabling Where task status equals, and selecting the condition. You can further refine the search based on the following conditions:

- Completed
- In Progress
- Failed
- Rejected
- Partially Completed
- Cancelled
- Scheduled

Note: See Task Status Description (see page 412) for more information.

Task Priority

Identifies task priority as the search criteria. You can select the task priority by enabling Where task priority equals, and selecting the condition. You can further refine the search based on the following conditions:

Low

Specifies that you can search for tasks that have a low priority.

Medium

Specifies that you can search for tasks that have a medium priority.

High

Specifies that you can search for tasks that have a high priority.

Date range

Identifies the dates between which you want to search submitted tasks. You must provide the From and To dates.

Customize the User History Tab

Administrators can customize the User History tab as follows:

- Specify a different task name and tag.
- Change the default display properties. By default, users can enter criteria that determines which tasks appear in the tab. Administrators can configure the tab to display the tasks for a current day automatically, preventing users from having to enter search criteria.

- Determine whether audit events appear in the Task Details page.
- Add a column to the task display.
- Specify the criteria for canceling or resubmitting tasks and events.

Follow these steps:

1. Navigate to Roles and Tasks, Admin Tasks, Manage Admin Tasks.

   The Select Admin Task page appears.

2. Select Name or Category in the Search Admin Task where field, enter the string you want to search, and click Search.

   Identity Management displays the admin tasks that satisfy the search criteria.

3. Select the task that includes the User History tab, and click Select.

   Identity Management displays the task details for the admin task.

4. Click the Tabs tab.

5. Click the Edit icon next to the User History tab.

   The tab details appear.

6. Edit the fields to customize the User History tab.

Configuration Settings for the User History Tab

Use the following fields to change the appearance and functionality of the User History tab.

Name

Defines the name of the task.

Tag

Defines a unique identifier for the task. It is used in URLs, web services or properties files. It must consist of letters, numbers, or underscores, beginning with a letter or underscore.

Hide Tab

Identifies that the tab is visible to the users, but will not be executed. If you select this option, Identity Management will display an error to the users.

Show task lists on load

Displays the tasks that have been submitted for the current day.

Note: If you have enabled this option, users clicking on View Submitted Tasks will directly see the tasks that were submitted on the same day.

Show audit events

Specifies if audited events are included in tasks in the View Submitted Tasks page.

Allow custom column

Indicates that you can append a custom column to the tasks table that you can view from the View Submitted Tasks tab and the User History tab. For example, you can append a column "User ID" to the tasks table that is displayed on the User History tab.

Custom column heading

Indicates the display name of the custom column.

Custom column attribute

Indicates the attribute that will be used to populate the custom column in the tasks table. For example, if you are searching for tasks that are performed on employees of an organization, you can append an organization column that displays the organization for each of the employees.

Cancel Tasks and Events

Identifies the criteria for canceling tasks or events. You can set the scope for this field by selecting one of the following options:

Task creator must be current user

Identifies that you can cancel or resubmit tasks or events that you have created.

Task creator must be in scope

Identifies that you can cancel or resubmit tasks that have been initiated by other users who match the user scope rules for the admin role that gives you access to the tab.

For example, you received the User Manager role, which includes View Submitted Tasks, because you met the criteria in a membership rule that includes scope over all users in the Employee organization. You can cancel or resubmit tasks that are submitted by all users in the Employee organization.

No restrictions

Identifies that any user can cancel or resubmit a task or event.

Not allowed

Specifies that a task or event cannot be cancelled or resubmitted.

Resubmit Tasks and Events

Identifies the criteria for resubmitting a task or event. You can set the scope of this field by selecting one of the following options:

Task creator must be current user

Identifies that you can cancel or resubmit tasks or events that you have created.

Task creator must be in scope

Identifies that you can cancel or resubmit tasks that have been initiated by other users who match the user scope rules for the admin role that gives you access to the tab.

For example, you received the User Manager role, which includes View Submitted Tasks, because you met the criteria in a membership rule that includes scope over all users in the Employee organization. You can cancel or resubmit tasks that are submitted by all users in the Employee organization.

No restrictions

Identifies that any user can cancel or resubmit a task or event.

Not allowed

Specifies that a task or event cannot be cancelled or resubmitted.

Show Tasks

Determines the tasks that appear in the User History tab.

Tasks performed on the user

Displays all the tasks that are performed on the selected user.

Tasks performed by the user

Displays all the tasks that are performed by the selected user.

Workflow approvals by the user

Displays all the tasks that the user has approved as part of a workflow.

The View User Activity Task

User activity is a history of tasks that involve a specific user. Administrators can use the View User Activity task to track the following user information:

- Tasks performed on the user
- Tasks performed by the user
- Workflow approvals performed by the user

To view user activity

1. Navigate to Users, Manage Users, View User Activity.

   The Select User screen appears.

2. Search for a user and click Select.

   The View User Activity screen appears.

Note: For more information on the user activity displayed, see the User Console Online Help.

Configure Correlation Attributes Task Screen

This topic applies only to CA CloudMinder.

Use the Configure Correlation Attributes Task Screen to configure correlation rules for the environment.

Identity Management reads configuration parameters into memory and periodically synchronizes the memory version with the database version of the common DSA. Since the Correlation Attributes is tenant specific, the Provisioning Server reads the Correlation Attributes from the corresponding tenant DSA during an Explore and Correlate operation. The updated Correlation rules take effect immediately with no need to wait for the Parameters Update Time.

Cleanup Submitted Tasks

With each task submitted, the runtime performance of tasks and events slows as the task persistence database grows. The garbage collecting of stored procedures mitigates the potential for performance problems or system outages due to the task persistence database running out of storage space. The ability to archive the tasks, gives the administrator the ability to view both current task and event information, as well as tasks and events that have been deleted.

In the User Console, Identity Management administrators can schedule jobs to automatically perform garbage collection and archive on a recurring basis.

Recurrence Tab

Use this tab to schedule your job. The fields in this tab are as follows:

Execute now

Runs the job immediately.

Schedule new job

Schedules a new job.

Modify existing job

Specifies that you want to modify a job that already exists.

Note: This field appears only if a job has already been scheduled for this task.

Job Name

Specifies the name of the job you want to create or modify.

Time Zone

Specifies the server time zone.

Note: If your time zone is different from the server's time zone, a drop-down box is displayed so you can select either your time zone or the server's time zone when scheduling a new job. You cannot change the time zone when modifying an existing job.

Daily schedule

Specifies that the job runs every certain number of days.

Every (number of days)

Defines how many days between job runs.

Weekly schedule

Specifies that the job runs on a specific day or days and time during a week.

Day of Week

Specifies the day or days of the week the job runs.

Monthly schedule

Specifies a day of week or day of month that the job runs on a monthly basis.

Yearly schedule

Specifies a day of week or day of month that the job runs on a yearly basis.

Advanced schedule

Specifies additional scheduling information.

Cron Expression

For information about filling out this field, see the following:

http://quartz-scheduler.org/api/2.1.0/org/quartz/CronExpression.html

Execution Time

Specifies the time of day, in 24-hour format, that the job is run. For example, 14:15.

Execute a Job Now

To execute a job immediately, use the Cleanup Submitted Tasks wizard.

Follow these steps:

1. Navigate to System, Cleanup Submitted Tasks.

   The Recurrence step of the wizard appears.

2. Select Execute Now and Next.

   The Cleanup Submitted Tasks step of the wizard appears.

3. Enter the minimum age, archive, audit timeout, time limit, and task limit information and click Finish.

   The job is submitted immediately.

Schedule a New Job

To schedule a new job, use the Cleanup Submitted Tasks wizard.

Follow these steps:

1. Navigate to System, Cleanup Submitted Tasks.

   The Recurrence step appears.

2. Select Schedule a new job, enter the job name and scheduling information for the job and click Next.

   The Cleanup Submitted Tasks step appears.

3. Enter the minimum age, archive, audit timeout, time limit, and task limit information and click Finish.

   The new job is scheduled.

Modify an Existing Job

To modify an existing job, use the Cleanup Submitted Tasks wizard.

Follow these steps:

1. Navigate to System, Clean Up Submitted Tasks.

   The Recurrence step appears.

2. Select Modify an existing job and choose an existing job, modify the scheduling information, and click Next.

   The Cleanup Submitted Tasks step appears.

3. Modify the minimum age, archive, audit timeout, time limit, and task limit information and click Finish.

   The existing job is modified.

Delete a Recurring Task

To delete a recurring task, follow this procedure.

Follow these steps:

1. Navigate to System, Select Delete Recurring Task.
2. Select the task you want to delete.
3. Click Submit.

Cleanup Submitted Tasks Tab

Use this tab to specify the minimum age, archive, audit timeout, time limit, and task limit of the task. Click Finish once you have completed all required fields. The fields in this tab are as follows:

Minimum Age

Specifies the minimum age of tasks that are in a final state (Completed, Failed, Rejected, Cancelled, or Aborted) to be cleaned up. For example, if 1 month is specified, any tasks that have reached a final state in the last month are retained. Any tasks that have reached a final state more than a month ago are subject to cleanup and archiving.

This is a required field.

Archive

Backs up tasks to the archive database before deleting them from the runtime database.

Once the job is run, if archive is selected, the data is committed to the archive database and removed from the runtime task persistence database. Data is not removed until a successful commit to the archive database happens.

Audit Timeout

Specifies the length of time before tasks in the audit state are subject to cleanup. Tasks in the audit state are not considered to be in a final state until this length of time has elapsed. Tasks in the audit state have not been submitted

Time Limit

Limits cleanup to a specific amount of time.

Task Limit

Limits cleanup to a specific number of tasks.

Delete Recurring Tasks

When a task no longer needs to be run on a recurring basis, the Identity Management administrator has the ability to delete the task. Once the task has been deleted, garbage collection and archiving is not performed for that task.

All tasks scheduled using the Cleanup Submitted Tasks:Recurrence wizard are listed on this page and the Identity Management administrator can choose which tasks to delete.

Note: The tasks are still present in the database, only the scheduling recurrence is deleted.

Manage Connector Servers

CA CloudMinder tenants can manage endpoints in their enterprise, called on-premise endpoints, by installing a local CA IAM CS.

Administrators who have the Manage Connector Server task configure a connection to a local CA IAM CS to allow tenants to manage endpoint accounts in CA CloudMinder.

Follow these steps:

1. Log in to the User Console as an administrator who can use the Manage Connector Server task.
2. Click System Tasks, Manage Connector Server.

   A diagram displays the cloud Connector Server and the local connector servers that it connects to.
3. Click Add to create a connector server definition.

4. Complete the following fields:

   Name

   Specifies the display name for the connector server.

   URL

   Specifies the Connector Server message broker URL.

   Domain

   Specifies the LDAP domain name.

   The Domain is included for backward compatibility. In most cases, do not change the domain.

   Username

   Specifies the user account that logs in to the connector server.

   Password

   Specifies the password for the account specified in the Username field.

   Allowed Tenant

   Specifies the name of the tenant who installed the connector server.

   Tenant Host ID

   Specifies an optional identifier in cases where multiple connector servers are installed at a tenant site to handle different sets of connectors. For example, one connector server supports Active Directory connectors and another connector supports Oracle connectors. The on-premise connector servers in this scenario are not peers since they are separate servers. However, they share a tenant name and authentication credentials for the Cloud connector server.

5. Click OK.
6. Drag the connector server that you created to the local connector server in the diagram above the list of connector servers to create the connection.

7. Specify which endpoint operations are handled by the on-premise connector server as follows:

   a. Right-click the connector server definition, and select Modify Route.

      You route any connector types that need to communicate with on-premise endpoints, such as a corporate Active Directory, to an on-premise connector server. You do not need to route connector types that operate as a cloud service, such as Google Apps, to an on-premise connector server.

   b. Select one or more endpoint type, then click OK.

      Note: For each tenant, a connector type can be assigned to only one on-premise connector server. Any connector types that are used in a route for another on-premise connector server with the same Tenant Name, but different Tenant Host ID, appear as unavailable for selection.

      A Connector Type can be shared however, if a filter is used. The filter can be an exact endpoint name match, or it can include the wildcard (*) character at the start and end of the name. If any endpoints have already been acquired, they can be selected from a list.

Logical Attribute Handlers

Identity Management logical attributes allow you to display user store attributes (called physical attributes) in a user-friendly format on task screens. Identity Management administrators use task screens to perform functions in Identity Management. Logical attributes do not exist in a user store. Typically, they represent one or more physical attributes to simplify presentation. For example, the logical attribute date may represent the physical attributes month, day, and year.

Logical attributes are processed by logical attribute handlers, which are Java objects that are written using the Logical Attribute API. (See the *Programming Guide for Java*.) For example, when a task screen is displayed, a logical attribute handler might convert physical attribute data from the user store into logical attribute data, which is displayed on the task screen. You can use pre-defined logical attributes and logical attribute handlers included with Identity Management, or create new ones using the Logical Attribute API.

Note: For more information on logical attributes, see the *Programming Guide for Java*.

In the User Console, the Environment category contains tasks for managing logical attribute handlers. The list includes pre-defined handlers shipped with Identity Management and any custom handlers defined at your site.

From the Environment task category, you can do the following:

- Create a new logical attribute handler with Identity Management
- Copy a handler
- Delete a handler
- Modify an existing handler configuration

Note: To change the order of execution for logical attribute handlers, use the Management Console.

Create a Logical Attribute Handler

To create a logical attribute handler

1. Navigate to System, Logical Attributes, Create Logical Attribute Handler.
2. In the Create Logical Attribute Handler screen, select Create Standard Logical Attribute Handler and click OK.
3. In the Create Logical Attribute Handler screen, configure the settings for the logical attribute handler.

   For a description of each field, click the Help link from this screen.
4. Click Submit.

   The handler is added to the list of handlers on the Logical Attribute Handlers screen.

Note: You do not need to restart the application server after configuring logical attribute handlers using the User Console.

Copy a Logical Attribute Handler

To copy a logical attribute handler

1. Navigate to System, Logical Attributes, Create Logical Attribute Handler.
2. In the Create Logical Attribute Handler screen, select Create a copy of a logical attribute handler definition and click Search.
3. Select a logical attribute handler (for example, ConfirmPasswordHandler) and click OK.

4. In the Create Logical Attribute Handler screen, configure the settings for the logical attribute handler.

   For a description of each field, click the Help link from this screen.

5. Click Submit.

   The handler is added to the list of handlers on the Logical Attribute Handlers screen.

Note: You do not need to restart the application server after configuring logical attribute handlers using the User Console.

Create a ForgottenPasswordHandler Logical Attribute Handler

The ForgottenPasswordHandler logical attribute handler uses separate logical attributes for the following:

- configuration
- runtime questions and answers

To create a ForgottenPasswordHandler logical attribute handler

1. Navigate to System, Logical Attributes, Create Logical Attribute Handler.
2. In the Create Logical Attribute Handler screen, select Create Standard Logical Attribute Handler and click Search.
3. Select the ForgottenPasswordHandler and click OK.
4. In the Create Logical Attribute Handler: ForgottenPasswordHandler screen, configure the settings for the logical attribute handler.

   For a description of each field, click the Help link from this screen.

5. Click Submit.

   The handler is added to the list of handlers on the Logical Attribute Handlers screen.

Note: You do not need to restart the application server after configuring logical attribute handlers using the User Console.

Delete a Logical Attribute Handler

To delete a logical attribute handler

1. Navigate to System, Logical Attributes, Create Logical Attribute Handler.
2. In the Delete Logical Attribute Handler screen, select the check box to the left of each logical attribute to delete.

3. Click Select.

   Identity Management displays a confirmation message.

4. Click Yes to confirm the deletion.

Modify a Logical Attribute Handler

To modify a logical attribute handler

1. Navigate to System, Logical Attributes, Create Logical Attribute Handler.
2. In the Modify Logical Attribute Handler screen, select the handler that you want to modify and click Select.
3. Select a logical attribute handler (for example, ConfirmPasswordHandler) and click OK.
4. In the Modify Logical Attribute Handler screen, configure the settings for the logical attribute handler.

   For a description of each field, click the Help link from this screen.

5. Click Submit.

Note: You do not need to restart the application server after configuring logical attribute handlers using the User Console.

View a Logical Attribute Handler

To view a logical attribute handler

1. Navigate to System, Logical Attributes, Create Logical Attribute Handler.
2. In the View Logical Attribute Handler screen, select the handler that you want to view and click Select.
3. View the logical attribute handler's properties and click Close.

Manage Secret Keys

Use Secret Keys to manage dynamic keys that encrypt or decrypt data. If you suspect that a user gained unauthorized access to a key, you can change the password for the keystore. The keystore is the database that stores secret keys. Once you change this password, Identity Management re-encrypts the values of the keys.

Each environment has a set of dynamic keys and a keystore password. If environments share a user directory, use the same dynamic keys and keystore password for each environment.

Keystore passwords are encrypted using keys embedded in encryption code or the parameters that are entered during installation of the Identity Management server. In a cluster, all nodes share the values for dynamic keys and the keystore password.

Encryption operations use the latest dynamic key for the correspondent algorithm and environment. Decryption operations check if a Key ID exists in the encrypted data, so that the right key is used. The Encrypted Text Formats section of the *Configuration Guide* provides more details.

Follow these steps:

1. Enter or modify the password to the Keystore.
2. Click Add a Key if you need another key.
3. Select an algorithm.
4. Enter a password for the key.

   For PBE and RC2, the maximum key length is 128 bytes.

   For AES, the valid key sizes are 16, 24, and 32 bytes.
5. Click Submit.
6. If you modified the Keystore Password, click Submit.

   Identity Management encrypts the values of the keys again.