IBM Security Identity Manager
Version 6.0.0.18

Planning Topics

IBM Security Identity Manager
Version 6.0.0.18

Planning Topics

Table of contents

Table list

Chapter 1. People planning

For the people in your organization, you can plan how to import identity records that create IBM® Security Identity Manager users and how to provide their passwords.

The following table includes examples of initial conditions and first implementation steps that administrators might take.

Table 1. Summary of potential user planning issues

Topic	Initial condition and questions	Example implementation steps
Identities	1 administrator account named `itim manager` exists with an initial password of `secret`. • Which identity records require IBM Security Identity Manager user IDs? • Does your early implementation need to define more administrators?	At a minimum, create 1 identity to test each group that IBM Security Identity Manager provides. Additionally, create another administrative user ID to guard against accidental loss of access.
Import identity records	A global identity policy exists. • Which data format does your organization plan to use to import identity records? • Are the attributes that the global identity policy specifies appropriate for your use?	Determine which identity feed to use and ensure that the appropriate attributes are specified in an identity policy. For most organizations, manually loading user data is not a practical method to define many users.
Policies (password, identity)	Password policy or forgotten password specifications do not exist. Password synchronization is on. The default identity policy is based on the `uid` attribute of the user. If the `uid` attribute has a null value, the identity policy concatenates the initial of the given name of a person with the surname of a person. What are the requirements of your organization for a password policy, challenge-response authentication, and identity policy?	Determine the password policy of your organization. Also, determine challenge-response authentication, and identity policy and then specify these policies.

Identity feed planning

Planning is required before you populate IBM Security Identity Manager with users by importing the identity records from your human resources repository or from other sources. An *identity feed* is the process of loading identity records into IBM Security Identity Manager.

The identity feed process includes the tasks in Figure 1.

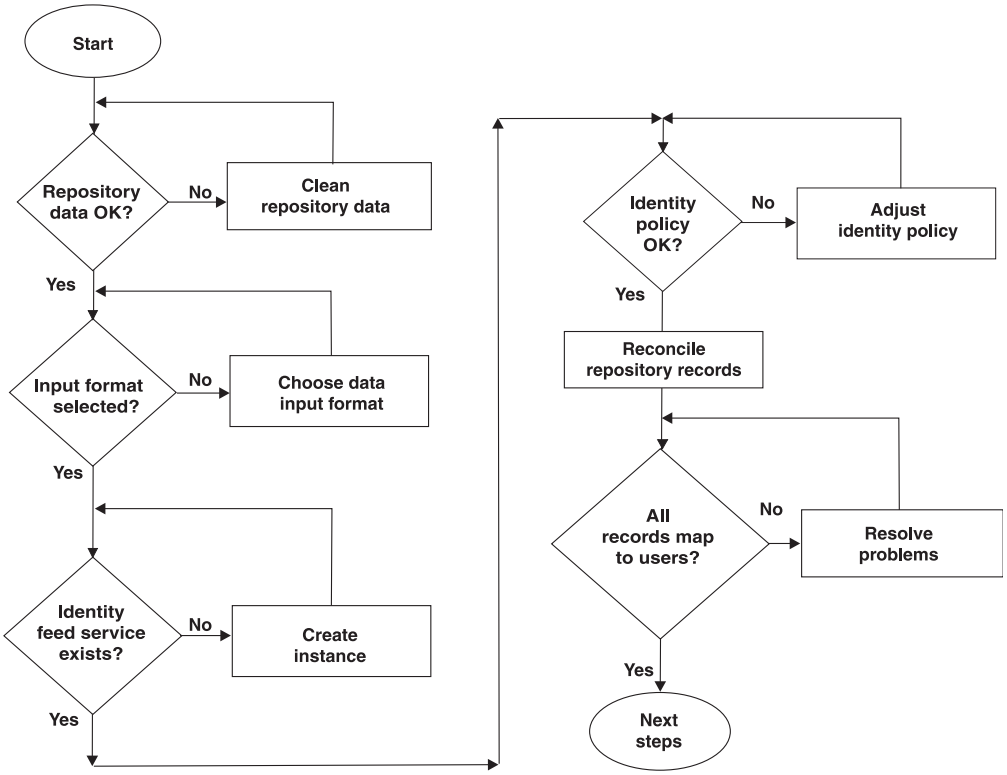

Figure 1. Obtaining identity data

The following identity feed tasks are basic:

1. Prepare the person data for the initial identity feed.

   Determine the best authoritative data sources, such as the human resources (HR) repository. You determine what information to use as the required attributes of a person. For example, data that indicates the title of a person might be required to correctly assign a role to that person as an IBM Security Identity Manager user.

   Minimally, IBM Security Identity Manager requires the following information to manage an identity:

   - Common name (LDAP CN)
   - Last name (LDAP SN)

   Note: Your planning also needs to anticipate the effect of missing information in the user record. If the record that you feed into IBM Security Identity Manager does not include an email address for the user, the user does not receive a password for a new account in an email. The user then must either call the help desk or contact the manager.
2. Determine the format to use to load the data.

For example, you might populate IBM Security Identity Manager people registry by reconciling with one of the following formats:

- Comma-separated value (CSV) identity feed
- DSML identity feed
- AD OrganizationalPerson identity feed
- INetOrgPerson (LDAP) identity feed
- IDI data feed

3. Create a service for the selected data format.
4. If necessary, adjust an identity policy to use in reconciling the repository identity records.
5. Reconcile the service to load the identity information.

   After the initial identity feed is completed, ensure that all the identities are loaded correctly. You might see inconsistencies in person and account data. The amount of cleanup depends on how well your organization prepares the identity data for the initial load.
6. When the initial reconciliation completes successfully, create accounts on the resources that your organization wants to manage with IBM Security Identity Manager.

Password policy and password synchronization planning

Determine the unique requirements for the rules that passwords for a service must meet. These requirements include length, type of characters allowed and disallowed, and whether to keep password synchronization, which is enabled by default.

To ensure that the correct users have access, create a password strength rule.

Password synchronization allows a user to change one password that is synchronized with the passwords for all the accounts that the user has on other resources that IBM Security Identity Manager manages. If you use password synchronization, you need to plan whether to use a single password policy or use several password policies. A single password policy enforces the policy on all accounts that a user owns. If several password policies exist, each policy might apply to a subset of accounts that a user owns.

Note: Password synchronization applies only to individual accounts.

Chapter 2. Security planning

After you install IBM Security Identity Manager, you must take more steps to plan how to grant and manage user access to resources.

Initial security conditions after installation

The initial security conditions of predefined groups, views, and access control items might require that you take more steps to grant and manage user access to resources.

Table 2 describes some of the initial conditions and first planning steps for security that administrators might take after installing IBM Security Identity Manager.

Table 2. Example planning issues

Topic	Initial condition and questions	Example implementation steps
Groups	All default groups initially have no members except the System Administrator group, which contains one user whose account is named *itim manager*. Which individuals in your organization need to belong in which default groups? Do these groups meet your site needs, or do you need to create additional, customized groups?	Complete one or more of these tasks: • Specify another system administrator to ensure that you do not accidentally lose access to IBM Security Identity Manager. • Create additional, customized groups, related views, and access control items.
Group settings in security properties	The check box to automatically populate IBM Security Identity Manager groups is disabled immediately after installation. If you feed identity records into IBM Security Identity Manager, you must manually populate members into predefined groups.	As administrator, you can specify the option to automatically populate IBM Security Identity Manager groups.
Views	All default groups have a set of permitted tasks that members can use.	As administrator, you can specify the view for the user's own accounts and information, and other tasks in the user interface.

Table 2. Example planning issues (continued)

Topic	Initial condition and questions	Example implementation steps
Access control items	Initially, all users have read access to their personal profiles. Other default access control items apply, for example, to the owner of a service and the manager of a subordinate. What are the requirements of your organization to expand or restrict access? Do you want to enable users to modify all or only some fields in a personal profile? Which users are allowed to manage delegation schedules?	Specify one or more access control items that restrict or expand access.
Forms	Initially, forms contain a set of attributes for each available category. The set of attributes can be configured with the form designer. What attributes are required in your business environment to appear on the form for each category? If you customize groups, views, and related access control items after initial installation, you might also want to show or hide some fields. The fields match the expanded or limited permissions you specified during customization.	As administrator, you can customize the forms with the form designer to reflect the attributes that need to appear or be hidden on the form.

Group management issues

Managing groups requires your attention to membership retention in groups. You must be aware of overlapping views if membership occurs in multiple groups, and to the scope of predefined and customized groups.

Automatic group accounts

For some predefined groups, you can configure IBM Security Identity Manager to automatically put the IBM Security Identity Manager accounts of newly named members in the default group. Automatic assignment does not apply to customized groups.

For example, newly named service owners who have IBM Security Identity Manager accounts can automatically receive accounts in the default Service Owner group. Additionally, the accounts of newly named managers are automatically put in the default Manager group. This activity can occur when you create or modify a user who is a subordinate by specifying the manager of the user.

The automatic action is enabled or disabled immediately. You do not need to restart IBM Security Identity Manager.

Upgrading IBM Security Identity Manager retains the group membership from the previous release. All groups that have organization tree access are migrated to the default Help Desk Assistant view.

To populate the default groups in Version 5 of IBM Security Identity Manager, you must manually assign members.

Group membership

A user can be a member of more than one group. The user obtains membership in a group either explicitly or by reference.

An administrator, or another user with the appropriate permissions, can explicitly assign a user to a group. If automatic population of groups is enabled, you can also cause a user to become a group member. You reference the user as the manager of another user or as the owner of a service. You can assign group members by using the Manage Group tasks on the user interface portfolio. You can also edit the IBM Security Identity Manager account profile of a specific user. You must ensure that a member of multiple groups does not receive accidental access to some tasks that fall outside the intended scope for the user.

Not all users gain automatic membership in groups, as described in Table 3.

Table 3. Assignment to groups

User given this relationship	**Automatically assigned to default group***
Manager of a user	Manager
Supervisor of a business unit	Not automatic
Owner of a service	Service owner
*If the property is enabled to automatically populate groups.	

Additional conditions apply:

- Having *no* IBM Security Identity Manager account does not prevent a person from being specified as service owner or supervisor. However, no IBM Security Identity Manager account is created or put in a default group. If that service owner or supervisor later obtains an IBM Security Identity Manager account, the account is not automatically put in a default group. You must create or modify a service or a user who is a subordinate.
- A person who has an IBM Security Identity Manager account becomes a user. Removing the user as manager of all subordinates, or removing the user as owner of all services, does not remove the IBM Security Identity Manager account. Removing the user as manager of all subordinates does not remove the user from a default group, such as the Manager group. Updating the manager attribute in the personal profile of a subordinate to reference a different manager does not remove the previously referenced user from the Manager group. Updating the manager attribute in the personal profile of a subordinate to reference no one, does not remove the previously referenced user from the Manager group. You can explicitly remove a user from the Manager group. The member is automatically removed from the group only when the user record is deleted.
- You can explicitly remove the user from a group and then update a user's personal profile again to reference the user as their manager. The referenced user again becomes a member in the Manager group.

Multiple groups and views

If a user is a member of multiple groups, the user has a merged view of all of the tasks that are provided to both groups. The user has the merged view, even if one of the groups does not grant access to the task.

You must coordinate the views that users see when they have membership in multiple groups. One group to which they belong might permit a task, and another group might not. If a task is permitted in any view, that permission takes precedence. A task might be permitted in the view that one group has. A user in that group can use the task even if the user is also member of a second group with a view that excludes the same task.

Scope of groups

The default tasks and associated access control items set the scope of activities for members of predefined and customized groups.

Scope of a customized group

A major business reason to customize groups is to increase or limit the scope of activities for group members. You might begin by examining the scope that the predefined groups have.

To enable tasks for customized groups of authorized users, define a new group and a view of tasks for group members. Additionally, specify one or more access control items to grant specific operations and permissions to each group member. You can also change the form for the user interface.

For example, your business needs might require:

- A user group with expanded permissions and activities that a subset of users can exercise.
- A supervisor group with expanded privileges and activities because the default Manager group has a relatively limited scope.
- Service owners or Help Desk Assistant groups with more limited privileges and tasks. The default groups for service owners and help desk assistants have a relatively large scope.

Note: You cannot create an additional group for system administrators.

Users with no membership in a default group

A user who has no membership in a default group can view and change the user's own profile. The user can also view and change other information, activities, and accounts and access.

All IBM Security Identity Manager users that do not explicitly belong to a group are automatically granted the base level of permissions. These users are granted access to the base set of views. See Table 4 on page 9.

Table 4. User permissions and access

Default tasks in view	Default access control items
• Change your passwords and specify information for your forgotten password questions. • View your personal profile. • View and request your accounts. You cannot change, delete, suspend, or restore your own account. • View and request your access. You cannot change, delete, suspend, or restore your access. • View your activities.	• Search and change your password. • Search, add, and change your account password. • View basic data about a service, such as service name and description.

Scope of the Auditor group

The scope of activities for members of the Auditor group is primarily to request reports and search for other information.

An auditor can run all reports and also search for a limited amount of information other than reports, such as roles. See Table 5.

Table 5. Default scope of Auditor group

Default tasks in view	Default access control items
• On the Self-service or Identity Service Center console – Change your passwords and specify information for your forgotten password questions. – View your personal profile. – View and request your accounts. You cannot change, delete, suspend, or restore your own account. – View and request your access. You cannot change, delete, suspend, or restore your access. – View your activities. • On the administrative console: For all users: – Run any report in all major report groups, including requests, users and accounts, audit and security, services, and custom reports. – View all requests, pending requests by user or service, and all requests by user or service.	• Self-care or Identity Service Center – Search and change your password. – Search, add, and change your account password. – View basic data about a service, such as service name and description. • Administrative For all users: – Search for account, dynamic role, location, role, organizational unit, person, and provisioning policy. – Run reports of all kinds, ranging from account to user requests.

Scope of the Help Desk Assistant group

The scope of activities for members of the Help Desk Assistant group is primarily to control the passwords, profiles, and accounts of other users.

These tasks might not be available if the person form is customized to exclude some of the attributes for which the help desk assistant has permission to read or

write. Additionally, help desk assistants can restore accounts, view the requests of others, and both manage and delegate to-do lists. See Table 6.

Table 6. Default scope of Help Desk Assistant group

Default tasks in view	Default access control items
• Self-service or Identity Service Center console – Change your passwords, and specify information for your forgotten password questions. – View your personal profile. – View, request, or delete your accounts. You cannot change, suspend, or restore your own account. – View, request, or delete your access. You cannot change, suspend, or restore your access. – View your activities. – Manage delegation schedules. • Administrative console For all users: – Create, change, delete, suspend, restore, and transfer users. – Change passwords for other users and delegate their activities. – Create, change, delete, suspend, and restore accounts. – Request and delete access. – View pending requests and all requests by user. – View activities for the Help Desk Assistant and for other users. – Manage delegation schedules.	• Self-care or Identity Service Center – Search and change your password. – Search, add, and change your account password. – View basic data about a service, such as service name and description. • Administrative For all users: – Perform all operations with all permissions on non-administrative accounts. – Search business partner organizations, non-administrative groups, locations, roles, and organizational units. – Perform all operations with all permissions on persons and business partner persons. – Delegate to-do lists.

Scope of the Manager group

The scope of activities for members of the Manager group is primarily to manage the accounts, profiles, and passwords of their direct subordinates.

These tasks might be unavailable if the person form is customized. For example, the person form might exclude some of the attributes for which the manager has permission to read or write. Managers can manage and delegate activities on their to-do lists. See Table 7 on page 11.

Table 7. Default scope of Manager group

Default tasks in view	Default access control items
• Self-service or Identity Service Center console – Change your passwords and specify information for your forgotten password questions. – View your personal profile. – View, request, and delete your accounts. You cannot change, suspend, or restore your own account. – View, request, and delete your access. You cannot change, suspend, or restore your access. – View your activities. – Manage delegation schedules. • Administrative console For supervised users: – Suspend and restore users. – Change user passwords and delegate user activities. – Request, suspend, restore, and delete user accounts. – Request and delete user access. – View pending requests or all requests by a user. – View the activities and manage delegation schedules.	• Self-care or Identity Service Center – Search and change your password. – Search, add, and change your account password. – View basic data about a service, such as service name and description. • Administrative For supervised users: – Run reports for: Account Operations Account Operations Performed by an Individual Accounts/Access Pending Recertification Approvals and Rejections Individual Access Individual Accounts Operation Pending Approvals Recertification Change History Recertification Policies Rejected User – Delegate activities to users.

Scope of the Service Owner group

The scope of activities for members of the Service Owner group is to manage a service, including the user accounts and requests for that service.

Additionally, service owners can view requests on services that they own that other users make. A request might be to authorize an account, unless the person form is customized to exclude some of the attributes for which the service owner has permission to read or write. A service owner can manage and delegate activities on their to-do lists. See Table 8 on page 12.

Table 8. Default scope of Service Owner group

Default tasks in view	Default access control items
• Self-service or Identity Service Center console – Change your passwords and specify information for your forgotten password questions. – View your personal profile. – View, request, or delete your accounts. You cannot change, suspend, or restore your own account. – View, request, or delete your access. You cannot change, suspend, or restore your access. – View your activities. – Manage delegation schedules. • Administrative console For the owned service: – Create, change, and delete a service. – Manage groups on a service, including membership, access, and recertification status. A service owner cannot add a group. – Manage accounts on a service, including requesting, changing, deleting, suspending, restoring, assigning, and making orphan accounts. Additionally, manage account defaults, recertification status, and reconciliation. – Manage all policies on a service. – Design workflows for account and access requests. – Request reports for users, accounts, services, and custom reports. – View all requests and pending requests by service. – View activities for the Service Owner. – Manage delegation schedules.	• Self-care or Identity Service Center – Search and change your password. – Search, add, and change your account password. – View basic data about a service, such as service name and description. • Administrative For the owned service: – Add a service owner group. – Run reports for access, individual access, orphan accounts, dormant accounts, pending recertification, recertification history, and recertification policies. – Add and modify accounts and account defaults. – Search accounts, account defaults, admin domains, business partner organizations, and organization units. You can also search groups, users, locations, persons, and service owners. – Use all permissions and operations on all policies. – Use all permissions and operations on workflows.

Access control item management issues

When managing access control items, pay attention to why you want to create them. Recognize potential conflicts between access control items that grant similar rights and resolve them.

Customized access control items

As an administrator, you can create customized access control items. For example, an access control item might limit an operation for members of a customized Service Owner group.

Coordination between access control items

You need to coordinate the outcome when multiple access control items apply to the same operation or attribute. It is possible that a user might be granted permissions by one access control item that are denied by another access control item.

A user might have multiple group memberships. The user's access is based on the widest privilege granted to any of the groups in which the user is a member. However, the user's access is disabled if it is explicitly denied to any of the groups of which the user is a member.

When conflict occurs between two or more access control items, the following rules apply:

- An explicit denial (with a Deny selection) by one access control item overrides an explicit grant by other access control items.
- An explicit grant by one access control item overrides an implied denial (with a None selection) by other access control items.

Use the Deny selection sparingly because an explicit denial overrides all other choices. You might use the None selection instead of the Deny selection.

For an attribute, the permission for a write operation takes precedence over the permission for a read operation. If you explicitly deny read permission and explicitly grant write permission, you are able to see the attribute on the form. The write permission takes precedence over the read permission.

Generally, if a user is granted permission to view or modify an attribute, the user can also see the attribute on the user interface even if read permission is denied. For example, if an access control item grants permission to define an access group, a member of the access control item can also view the access group list, regardless of whether the operation to view group members is granted or denied.

An example focus problem and solution

A problem in focusing an access control item can occur when you create a customized access item for an account object class.

For example, you might intend to prevent Read and Write operations for the Description attribute of an account object class. You might specify a permission value of None for both operations. You select the membership of the access control item as the owner of the service on which the account resides. Testing the new access control item, you then log on as the service owner and begin to request an account for another user. You discover that you are still able to both read and write the account description field.

There are two causes:

- The membership specification of the new access control item applies to accounts that exist. In this case, the membership is for the owner of the service on which an existing account resides. However, the access control item does not prevent Read and Write operations during the account creation process, before the account exists.
- As service owner, you belong to the service owner group, which has an access control item named `Default ACI for Account: Grant All to Supervisor/Domain Admin/Sponsor/Service Owner/Access Owner`.

In your customized access control item, you specified a permission value of None for both operations. However, the default access control item specifies a permission value of Grant. The Grant permission takes precedence over a value of None in any other access control item that applies to the operation.

You might change your customized access control item:

- Change the permission value to Deny for the Read and Write operations. The use of Deny by one access control item overrides an explicit Grant by other access control items. Use the Deny selection sparingly because an explicit denial overrides all other choices.
- Change the membership of the customized access control item to include the service owner group. The change ensures that the access control item applies during account creation.

Entity types used by access control items

Access control items focus on target entity types.

IBM Security Identity Manager provides default access control items that target a protection category. You can also assign a specific object class, such as the `erPosixLinuxAccount` object class, to an access control item.

Access control items focus on an entity such as an account, organization, or role. Some access control items require the selection of an entity subclass. An access control item can focus on these categories of entity types:

Account
Represents a user's access to a managed resource.

Note: Install the service profile for the managed resource on which the accounts that you want reside. Then, create an access control item for the account object class.

Account Default Template
Provides default values for account attributes when requesting an account on a service.

Admin Domain
Identifies a subsidiary part of an organization as a separate entity with its own policies, services, and access control items. Identification includes an administrator whose actions and views are restricted to that domain.

Business Partner Organization
Identifies a business partner organization, which is typically a company outside your organization that has an affiliation, such as a supplier, customer, or contractor.

Business Partner Person
Represents an employee of an outside entity with which your organization is affiliated, such as a supplier or customer.

Credential
Represents a credential in the credential vault.

Credential Lease
Represents the lease for a user to use a credential for a limited time period.

Credential Pool
Provides a way to group credentials with similar access privileges. This grouping can be defined as a service group or a set of service groups.

Dynamic Organizational Role
Selects users based on the attributes in an LDAP filter, such as the title of a user. When a user is added to the system and the LDAP filter parameters are met, the user is automatically added to the dynamic role.

Identity Manager User
Represents an account of the ITIM Service.

Identity Policy
Defines who has access based on an identity policy.

ITIM Group
Specifies a collection of users with accounts on the IBM Security Identity Manager service.

Location
Specifies a container that is different geographically but contained within an organization entity.

Organizational Unit
Identifies a subsidiary part of an organization, such as a division or department. An organizational unit can be subordinate to any other container, such as organization, organizational unit, location, and business partner organization.

Password Policy
Defines who has access based on a password policy.

Person
Specifies a person whose identity record is managed as an account by IBM Security Identity Manager.

Provisioning Policy
Defines who has access based on a provisioning policy.

Recertification Policy
Defines who has access based on a recertification policy.

Report
Specifies report access control items for groups that are allowed to run a specific type of report. For example, the service owner group might have access to run the Orphan Accounts Report. The auditor group might have access to run the Recertification Change History Report.

Separation of Duty Policy
Represents a logical container of separation rules that define mutually exclusive relationships among roles.

Service
Identifies a managed resource, such as the Windows Service, and IBM Security Identity Manager itself.

Service Group
Specifies a collection of users with user accounts on a specific service, such as an accounting application. A service group is related to groups, not services. In other words, a service group is *not* a set of services.

Credential Service
Specifies information about the resource for a credential in the vault.

Service Selection Policy
Defines who has access based on a service selection policy.

Shared Access Policy
Defines who has access to the credentials or credential pools.

Static Organizational Role
Specifies a subset of one or more privileges that can be assigned to users. For example, the ITIM Administrators role is a predefined role.

If a role is a member of another organizational role, then that role member inherits the permissions of the organizational role. All members of the organizational role and its role members have the same set of privileges.

Workflow Design
Defines who can create or modify account and access entitlement workflows.

Access entitlements and access control items

Access control items defined for Service, Service Group, and Account control a user's access privilege for access configuration and user access management that is based on service group.

Access control items defined for role, dynamic role, and Person control the access privilege for access management and user access management for access based on an organizational role.

IBM Security Identity Manager provides default access control items that target access entitlements, as described in Table 9.

For more infomation on default access control items for the shared access module, see the IBM Security Privileged Identity Manager product documentation.

Table 9. Access entitlements and access control items

Who is permitted	Default access control items related to access management	Effect
All users	Service group - read all access attributes Static role - search and modify attributes Dynamic role - search attributes Person - modify and use the erroles attribute read/write for self Account - search, add, view, and remove group member for self	Allow users to request that new access authorization and to view and remove their own access.

Table 9. Access entitlements and access control items (continued)

Who is permitted	Default access control items related to access management	Effect
Manager or supervisor or the account owner	Service group - search and read all access attributes Static role - search and modify attributes Dynamic role - search attributes Person - modify and use the erroles attribute read/write for subordinates Account - search, add, view, and remove group member for subordinates	Allow a manager to view, request, or remove access of a subordinate.
Help desk assistant	Service group - search and read all access attributes Static role - search and modify attributes Dynamic role - search attributes Person - modify and use the erroles attribute read/write for all Account - search, add, view, and remove group member for all	Allow all help desk users to view, request, or remove access for all users in the organization.
Service owner or access owner of the service on which the account resides	Service group - all access control item operations Account - all access control item operations	Allow service owners or access owners to search a group, define access, and recertify access. Allow service owners or access owners to manage accounts and group members for a service or defined access that they own on the service.
Sponsor of the business partner organization in which the account resides	Service group - search or read all access attributes Account - search, and add, view, or remove group member	Allow a sponsor to view, request, or remove access of a subordinate.
Auditor group	Service group - search Account - search and read all access attributes	Allows members of the auditor group to view access reports.
Service owner or auditor group	Reports (access) - run an operation	Allows members of the service owner or auditor groups to view the access report.

Table 9. Access entitlements and access control items (continued)

Who is permitted	Default access control items related to access management	Effect
Auditor, manager, or service owner groups	Reports (individual, access) - run an operation	Allows members of these groups to view the individual access report.
Privileged Administrator group	Static role - all access control item operations Dynamic role - all access control item operations Reports (individual, access) - run an operation	Allow all privileged administrators to view, add, or remove access in the organization. Allows members of these groups to view the individual access report.

Recertification policies and access control items

Recertification policies can also be targets of access control items.

IBM Security Identity Manager provides default access control items that target recertification policies. The default access control items are described in Table 10.

Table 10. Recertification policies and access control items

Who has access	Target object and access control item	Effect
Service owner group	Recertification policy - add, modify, remove, or search	Allow service owners to manage recertification policies.
Auditor or manager group	Recertification policy - search	Allows members of the auditor group or manager group to search or view recertification policies.
Auditor, manager, or service owner groups	Reports (pending recertification, history, and policies) - run operation	Allows members of these groups to view these reports.

Chapter 3. Role planning

Managing roles requires understanding the application of access control and access provisioning models in a customer deployment. Managing roles requires specifying static or dynamic role members, and determining the appropriate scope of a role in an organization tree.

Access control models

There are several commonly found access control models in a centralized identity management solution.

The access control model that an organization uses depends on certain factors. There might be:

- Externally mandated policies
- The maturity of existing identity management processes
- A range of identity management target systems.
- Future requirements
- The number of users managed
- Risk assessment statistics
- Return on investment statistics

In IBM Security Identity Manager, organizational roles can be used to support the following types of access control models:

Role-Based Access Control (RBAC)
This model grants access privileges to users based on the work that they do within an organization. The model allows an administrator to assign a user to single or multiple roles according to their work assignments. Each role enables access to specific resources.

Discretionary Access Control (DAC)
This model enables the owner of a resource to decide whether to allow a specific person access to the owned resource. This system is common in distributed environments that evolved from smaller operations into larger ones.

Mandatory Access Control (MAC)
This model enables grouping or marking resources according to a sensitivity model. This model is most commonly found in military or government environments. An example of this model is the marking of Unclassified, Restricted, Confidential, Secret, and Top Secret resources. The privileges that a user is granted to view certain resources depends on the clearance level of the user.

Access provisioning models

IBM Security Identity Manager provides role-based and request-based access provisioning models.

Role-based
Access to managed services are provisioned automatically based on the user's roles in the organization. To some degree, role-based provisioning

can support a role-based access control model. Role-based provisioning can be used when access control is not centrally managed by a common access control system. The automation between the role and accounts and groups on the target resource, and strict enforcement of role relationships, ensures that access to the IT resource is based on the role of the user.

Request-based
Access entitlements are authorized to a user based on the user's roles in the organization. Entitlements enable the user or other managers or administrator to request the account or access.

Request-based provisioning is often used to support Discretionary Access Control (DAC) and Mandatory Access Control (MAC) access control with a combination of appropriate approval processes. Sometimes, there might be mixed usage of the two models for different sets of users in the organization or for different sets of target services.

Organizational roles and access provisioning

A *user role* is also termed a business role or positional role. A user role represents a group of users with a particular meaning in a business model. The group might be a classification of users who share a business function.

User roles can be modeled with an organizational role in IBM Security Identity Manager and used to support role-based provisioning. A user role can be mapped to a set of access entitlements in the provisioning policy. Access to IT resources is automatically provisioned for the users that belong to the role.

User roles are often modeled to help with user management for the business. User roles can also be used to support role-based access control and role-based provisioning. Access to IT resources might be managed by the following systems:

Central access control system
A role-based access control model grants access to resources based on a user role, such as the user's job title or work responsibility.

Distributed system for a specific resource
A role-based provisioning model automates the access entitlement provisioning process for a specific managed resource, and is based on the roles to which the user belongs.

Consider the following items when you design provisioning policies:

- The target services to manage
- The number of groups on each service
- The number of user roles in the organization
- The pattern of user roles and access entitlement mappings to the target services

An access entitlement can be mapped to an account on a service or to specific group members on a service. A provisioning policy allows a user role to map to multiple entitlements for different services. It allows multiple roles to have the same set of access entitlements. It is also possible to have multiple provisioning policies for the same role, each granting a set of accesses for the role.

An organizational role in Security Identity Manager can also be used to represent access to IT resources. The access can be mapped to one or multiple services that

represent aggregated access to the resources. The accesses are defined by using a Security Identity Manager provisioning policy with both automatic and mandatory entitlement parameters.

This type of organizational role can be directly exposed to the user for access requests. The role can be categorized based on its access type, such as access to an application or a shared folder.

This type of organizational role provides request-based provisioning by enabling requests to aggregated accesses. By giving the appropriate business-oriented name and description to the access and by setting up accesses in a provisioning policy and specifying the appropriate role approval process, you can build a provisioning mechanism to support the access control models that were described in "Access control models" on page 19.

If the role is a child role of another organizational role, which then becomes a parent role, then that child role inherits the permissions of the parent role. In addition, if a role is a child role of another organizational role in a provisioning policy, then that child role also inherits the permissions of provisioning policy.

For more information about how to design these access control systems, see the IBM Redbooks® that describe design activities for Security Identity Manager.

Static and dynamic roles

IBM Security Identity Manager provides static and dynamic roles.

In static organizational roles, assigning a person to a static role is a manual process.

In the case of a dynamic role, the scope of access can be to an organizational unit only or to the organizational unit and its subunits. Dynamic organizational roles use valid LDAP filters to set a user's membership in a specific role. For example, a dynamic role might use an LDAP filter to provide access to specific resources to users who are members of an auditing department named `audit123`. For example, type:

```
(departmentnumber=audit123)
```

Dynamic organizational roles are evaluated at the following times:

- When a new user is created in the Security Identity Manager system
- When a user's information, such as title or department membership, changes
- When a new dynamic organizational role is created

Roles in the organization tree

You can use roles to plan a job title or responsibility, and you can use roles to grant access to accounts and attributes.

Both a static role and a dynamic role can be associated with a business unit in the organization tree. The association can be used to support delegated administration for a role or role assignment. An access control item can specify which user is allowed to create, modify, or delete a role. The specification is based on the association of the role in the organization tree.

- A static role can be located anywhere in the tree. Any user in the same organization can be manually attached to the role.

For a static role, an access control item can specify who is allowed to add or remove users from the role-based association. The specification is for the organization tree of the role and user.

- A dynamic role defines membership based on an LDAP filter and has a scope relative to its position in the tree. Placement of dynamic roles within an organization tree can have performance implications. For more information, see the *IBM Security Identity Manager Performance and Tuning Guide*.

  The scope of a dynamic role can be:

  Single Applies to users in the local business unit

  Subtree
  Applies to the local business unit and all subbusiness units

  For example, suppose that an organization has a depth of containers for the user population, similar to Figure 2.

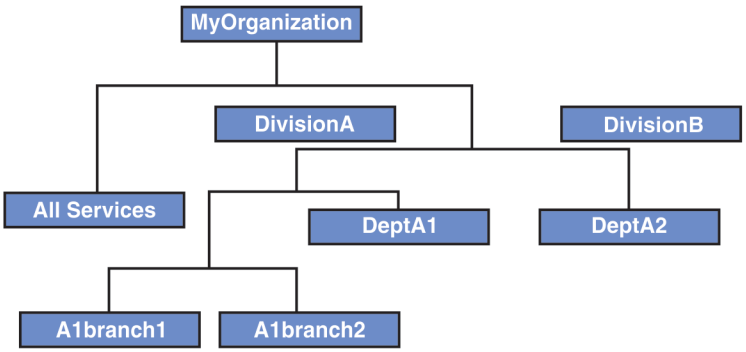

Figure 2. Example roles

Suppose that you configure dynamic roles similar to this list:

– `role_A` for DivisionA
– `role_A1` for department A1
– `role_A1_1` for branch 1 in department A1

Each of these dynamic roles might have a scope of subtree and an LDAP filter such as (objectclass=*). A user in `ou=A1branch1` receives all three roles: `role_A`, `role_A1`, and `role_A1_1`.

To ensure that a discrete dynamic role applies to users based on their location in the tree, you might need to take one of the following actions:

– Place the users in the leaf nodes (containers)
– Make the LDAP filter specific to the location
– Specify the role scope as single.

Chapter 4. Organization tree planning

There are many things to consider when you are planning an organization tree.

Consider the following items:

- Obtaining agreement on the primary and secondary requirements that a structure must satisfy.
- Clarifying the degrees and levels of delegated administration for subsets of services or users and then specifying admin domains that accomplish these goals.
- Using automated identity feeds to load identity records.
- Determining user access privileges with different degrees of scope within the branches of an organization.
- For organizations in different geographic areas, enabling a flow of changing administrative access to the system for a specific region or time interval.
- Determining effective change control of policies, roles, groups, and other security functions that IBM Security Identity Manager provides.
- Determining the scope of influence within an organization tree for IBM Security Identity Manager policies and workflow participants, such as supervisors and administrators.
- Easing migration from pilot to production-level structures.

Organization tree models

An organization model can use simple or complex branching.

For example, within a branch in an organization tree, a structure of services and users might provide:

- All-in-one organization

  Both the user and the services are within a single organization. An all-in-one model for all services and users has a limited use. Although simple to administer, the model is not scalable to a large population of users.
- Separate branches for services and for users

  Separate branches might provide a services branch and a user branch in an organization. Issues can arise in administration and deployment for different depths of subordinate objects in each branch. Designing the structure must consider the difficulty in moving services and policies within a tree in comparison to the relative ease in moving users.

  Objects that are created within an organizational container must typically be deleted and created again to put the object into a different organizational container. However, existing users can be moved to a different business unit by use of a transfer.

  For example, an organization might manage services as one branch in a resource tree and employees in another branch in a structure similar to Figure 3 on page 24.

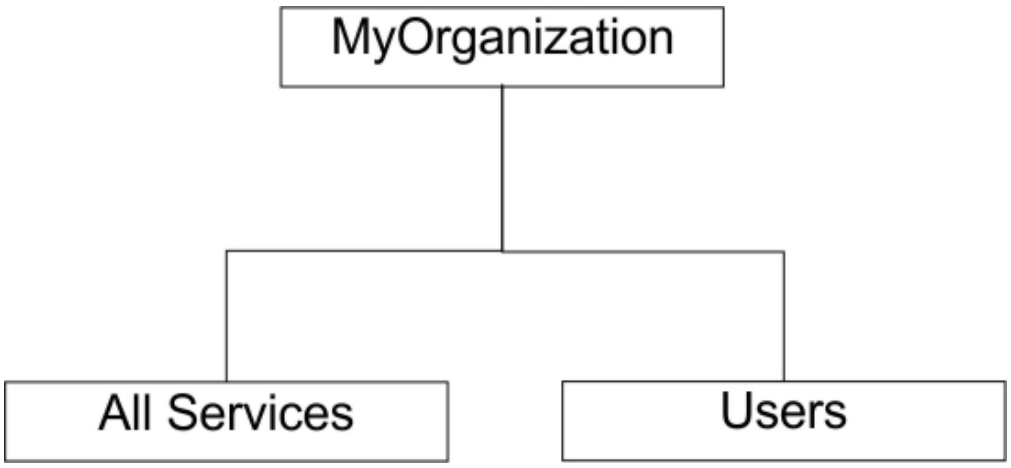

Figure 3. Example configuration

- Exact replication of an entire hierarchy of a continuously changing business

  You might attempt to maintain an ongoing, exact tree that represents your business organization. However, changes that occur in the hierarchy and membership of the persons business unit itself then require exact synchronization of the users within an electronic organization tree.

Organization chart example

An organization might need an extended structure to manage its services and employees.

For example, an organization might require a structure similar to Figure 4.

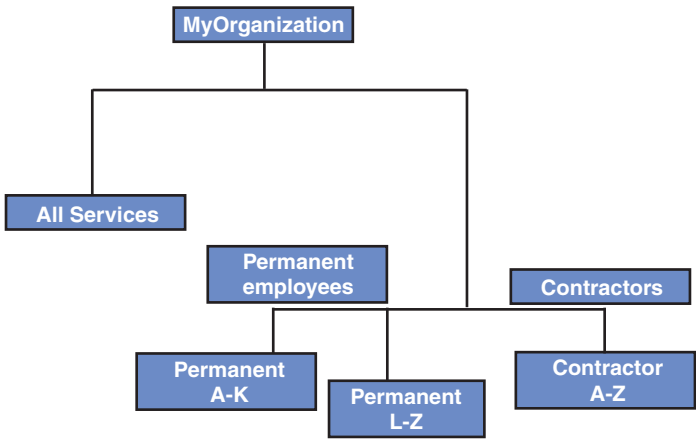

Figure 4. Example organization chart

The organization chart in this example includes the following advantages:

- The high-level split of services from people separates identity management from management of the solution.

  To implement the structure, you would configure admin domains to provide management of objects.
- The user container structure is based on one or more user attributes. There might be structures for a reporting or other line of business hierarchy similar to Figure 5 on page 25.

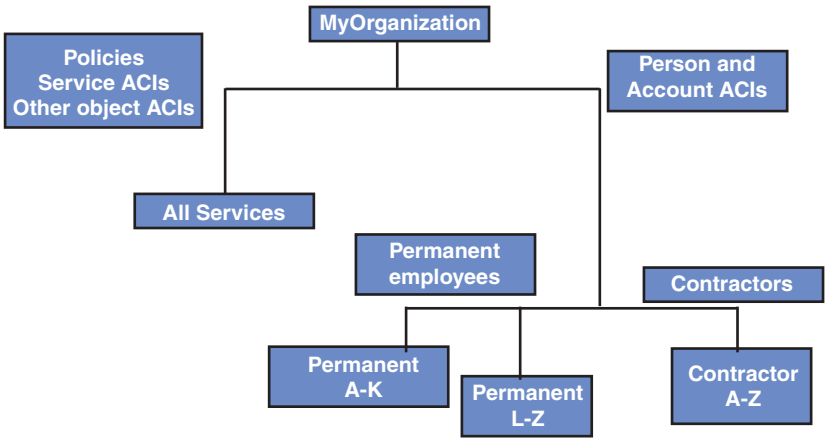

Figure 5. Example organization chart

In this example:

- The user branch is a simple structure that is based on a common piece of data, such as the family name. This model is easy to deploy and requires little placement rule coding for the identity feed.

  The branch structure is largely based on the information available in the identity feed and the number of users in each container. You must analyze the identity data before you confirm the structure.
- The services branch contains all services, policies, and other objects along with their access control items. The structure depends on the requirements for management and ownership of the services and other considerations.

When pilot tests are complete, deploying a more complex structure might keep the service branch as it is and build the new user structure. You then move the users into the new structure.

There are many alternatives other than this example. For an organization with centralized management, you might create a branch that contains services. You might build a separate branch or branches that contain users, each with its own delegated administration. For organizations with distributed management, you might create a container that has both services and people.

For more information about organization design for IBM Security Identity Manager, see Redpapers or Redbooks that might be published for this product. These sources are provided by the IBM International Technical Support Organization. You can also refer to IBM Global Business Services® or another qualified project management consultant.

Scope of governing entities in the organization tree

Most IBM Security Identity Manager policies allow specifying the scope of the governing services in the organization tree. The specification is based on association of the policy with the business unit.

A dynamic role supports the scope of governing users based on its association with the organization tree. An access control item supports the scope for the protecting object types based on its association with the organization tree.

The scope of policies, roles, and workflows differ in their effects in an organization hierarchy. See the summary in Table 11.

Table 11. Scope and inheritance in a tree

Entity Type	Scope of Governing Entity
Identity policy	Services associated with the same business unit or in the subtree.
Password policy	Services associated with the same business unit or in the subtree.
Provisioning policy	Services associated with the same business unit or in the subtree.
Service selection policy	Provisioning policies associated with the same business unit or in the subtree.
Static role	Users in the entire organization to which the role belongs.
Dynamic role	Users associated with the same business unit or in the subtree.
Workflow	Service or Access in the entire organization.
Access control item	Objects (based on the protected object type) associated with the business unit or in the subtree.
Shared access policy	Credentials or credential pools associated with the same business unit or in the subtree.

Provisioning policies

A provisioning policy grants entitlement to accounts on managed services for set of roles.

A provisioning policy governs roles and users in the entire organization, regardless of the association of the policy with the business unit in the organization tree. The policy governs only services that are associated the same business unit or sub tree of the business unit with which the policy is associated. The provisioning policy must be at the same level or higher than the related service.

A role that is referenced in the provisioning policy can be anywhere in the tree of the organization. Any user can be manually attached to the role.

Service selection policies

Service selection policies extend the ability of provisioning policies by provisioning accounts based on user attributes.

Using JavaScript, service selection policies determine a user attribute to select a service. A user attribute might be a geographic location. A service might be a local Windows service in a geographically dispersed business.

The scope of a service selection policy determines which provisioning policies are affected:

- Single

  Provisioning policies only at the same level in the organization tree can reference the service selection policy.
- Subtree scope

  Provisioning policies only at or below the same level in the organization tree can reference the service selection policy.

Identity and password policies

An identity policy determines the user ID for account creation. A password policy sets the password strength rules.

Place identity and password policies at the same level or higher than the services to which they are applied. The scope can be set as single level or subtree. Identity and password policies can cover all services, a specific service profile, or a set of services within the scope.

Workflows

A workflow is a set of steps or activities that define a business process. For example, you can use workflows to customize account provisioning and lifecycle management, such as adding, removing, and modifying users and accounts in IBM Security Identity Manager.

There is no inheritance relationship between a workflow and users or policies in an organization tree. A workflow does not need to be at or above a level in an organization tree. There is no inheritance of workflows to users. There is no placement relationship between workflows and provisioning policies.

Workflow participants

A key component of approval workflow nodes is the participant and escalation participant. The escalation participant is one or more signature authorities that must approve or reject the request.

The participant can be an individual or a group of individuals, including the following individuals:

- Requestee, the person for whom an action is done
- Requestor, who does an action
- Service owner
- System administrator

Some participant types are organization-related:

- Sponsor for a workflow in which a valid requestee is specified.

  You can specify a sponsor for each business partner person or business partner organization. A workflow determines the sponsor that is appropriate to the requestee, based on a BPPerson object or BPOrg object. In the tree, the Sponsor that is selected is the closest to the requestee BPPerson.
- Supervisor for a workflow in which a valid requestee is specified.

  You can specify a supervisor for users, organization units, and locations. A workflow determines the supervisor that is appropriate to the requestee, based on the Person object, or from the closest location or organization unit object that has a supervisor. In the tree, the Supervisor that is selected is the closest to the requestee person.
- Domain administrator for a workflow in which the service subject or requestee is specified. If the service is not specified, it is based on the admin domain that is closest to the requestee.

  You can specify an administrator for an admin domain. A workflow determines the administrator based on the admin domain that has an administrator and is closest to the service.

Workflows and the organization tree

Workflows can use one of the dynamic participant types that are determined at run time. Workflows can also explicitly map an object instance, such as a specific user, as a participant.

The workflow participant in a business unit is termed Supervisor. In a Person profile and also in an access control item, the workflow participant is termed Manager. For example, a Supervisor participant is based on the Manager relationship of a Person, or based on a Supervisor relationship of a business unit.

Participant resolution is based on a relationship to the entity, which has a dependency on the organization tree:

- Manager (approver) is defined on a Person object

  In this case, generic workflows can apply to all parts of the organization tree. Approval for a new account request might go to the line manager first and then to the owner of the service that has the account.
- Manager (approver) is defined on an organization unit, such as a team

  In this case, the leaf nodes of the organization tree need to be the teams to which each user belongs. Their line manager is specified as their supervisor/sponsor.

Access control items

An access control item applies only to entities of protected category within its branch or specified with subtree scope in lower branches of an organization tree.

An access control item applies to entities and processes only within its branch. If specified with subtree scope, the access control item applies to lower branches of an organization tree. Most of the access control items are defined at the Organization level. There are more considerations, such as:

- Creating access control items at a specific business unit in the Organization to grant operations and permissions for objects within that branch.
- For a business unit object such as a business partner organization, an access control item must be originated one level higher. The access control item has a target entity type of business partner organization, to enable searching.

Customization and bulk loading of identity data

Configuring organization trees must also consider requirements to provide customization and bulk loading of identity data.

Customization with JavaScript APIs

The organization tree might drive some requirements for customization that support your specific business processes.

You might dynamically assign approvers based on the association of the user to the organization tree. JavaScript APIs are often used to look up or search for various types of business units in the organization tree. The search is based on the attributes of the user.

Bulk load of identity data

A small user population that ranges from tens or a few hundreds of users might allow manual loading of identity data. Most projects require a bulk load process. Manual loading is likely to also require a manual refresh of identity data, which adds significant administrative effort to the project.

The IBM Security Identity Manager data load mechanism can include JavaScript to define where each new user is placed in the tree. If you do not include a placement rule, all users are placed into a default organization. You must later move each user into the correct organization container.

To bulk load users into their correct organization container, you must define placement rules to place the user into the appropriate organization container. You can code JavaScript to use some user attribute in an identity feed for incoming Person object data. The process looks up the Organization container in the placement rule definition.

Chapter 5. Workflow planning

To plan workflows for your business needs, you must understand the behavior of the installed workflows.

Planning also includes analyzing the business requirements and policies that are related to account and access provisioning. You need a design that meets these requirements. Your implementation of the design creates more operations, customizes installed operations, and defines appropriate account and access request workflows to support the approval process.

Operation workflows

Operation workflows define the business processes for managing account, user, and business partner user entities.

When an administrator adds, removes, or modifies an entity with the IBM Security Identity Manager graphical user interface or the APIs, the operation workflows complete the request. To implement your specific business processes, you can customize operation workflows by modifying both the flow and the specific activities. You can also extend Security Identity Manager capabilities to call new operations. You create new operations and use the APIs, lifecycle rules, and operation workflow activities.

Operation workflows support both input and output parameters.

Operation levels

Operation workflows are defined at global and also at lower levels in the system.

The available operations depend on which operation level is selected. Click **Configure System > Manage Operations** to access these operation levels in the IBM Security Identity Manager administrative console.

Global
Applies to all entities and entity types. A global operation is always a static operation that does not require a target entity.

Entity type
Applies to all entities of that specific type, such as an Account or Person entity. For static operations, an entity type defines the namespace of the operation; for a nonstatic operation, an entity type defines the type of the target entity. An operation at the entity type level does *not* overwrite an operation at the global level.

Entity Overrides the operations that are defined at the entity type level.

Static and nonstatic operations

Lifecycle operations are either static or nonstatic. Static operations are applied to an entity or entity type. Nonstatic operations are applied to a specific instance of an entity.

Static operation
A static operation is an operation that is defined for an entity or an entity

type that is not started on an instance of the entity. All input parameters of static operations are explicitly defined.

In static operations, the target of the operation is passed as input parameters or is resolved as part of the operation workflow. To ensure that the operation workflow completes successfully, you must pass the appropriate information about the target to the operation.

For example, when the add account operation is started, the account profile must be passed first to the add account operation. This profile information helps distinguish between the entity types (Account and Person) and the operations for entities and entity types. The target of the operation does not have to be an entity that IBM Security Identity Manager manages, because the target of the operation is not used to resolve operation definitions.

If you add an existing operation to a workflow, map the relevant data items to all of the input parameters of the new operation. Mapping consists of matching relevant data items to input parameters between two or more operations. For example, you might change the workflow for a modify Person operation. The change transfers a user when changes are made to their *bu* attribute. You create a relevant data item to hold the new container of the user. You then map this data item to the input of an extension workflow node of the transfer Person operation.

The following list shows the static operations:

- add (Account, Person)
- modify (Account)
- selfRegister (Person)

Nonstatic operation

A nonstatic operation is an operation that is defined for an entity or entity type that must be started on a specific instance of an entity that is currently managed by IBM Security Identity Manager to ensure that the query and the operation workflow succeed.

The target of a nonstatic operation is passed into the operation as entity context information. The entity (Account or Person) that the workflow applies to is passed implicitly to the workflow. The entity is always mapped to a relevant data named Entity rather than an input parameter. The target of the operation, the Entity object, must exist in IBM Security Identity Manager. An entity (Account, Person, Business Partner Person) is added to IBM Security Identity Manager with the appropriate add operation.

All nonstatic operations have a relevant data item with an ID of Entity, and this relevant data item represents the target of the operation. Similar to all other relevant data, this relevant data item can be used to provide input parameters to activities. The relevant data item can be accessed with JavaScript scripting to support additional customization of the operation workflow.

The following list shows the nonstatic operations:

- delete (Account, Person)
- modify (Person)
- suspend (Account, Person)
- restore (Account, Person)
- transfer (Person)

- changePassword (Account)

System-defined operations

IBM Security Identity Manager includes a set of system-defined operations that implement the features of the system.

The system-defined operations are specific to the entity types. Although you can customize these operations, you cannot change the input parameter definitions, the type of operation (static or non-static), or the name of the operation. Click **Configure System** > **Manage Operations** to access these operations in the Security Identity Manager console.

If you directly customize a system operation of an entity type, you cannot delete it and then later restore it back to the default operation. Deleting a system-defined operation for an entity type is not allowed. You must manually remove the customization.

Operations defined for entities override operations that are defined for entity types. If an operation with the same name exists for both an entity and an entity type, the entity operation is the operation that is started by the operation workflow. Because system-defined operations implement the base business processes for Security Identity Manager, exercise caution when customizing these workflows.

For example, if you have specific business process requirements, create a user-defined operation by overriding the system-defined entity type operations. The system-defined delete operation for the Account entity type deprovisions the account and permanently removes the user data from the remote system. To prevent the loss of that user data on AIX® systems, create a delete operation for AIX accounts that sends a request to the service owner of the AIX systems. The request asks them to specify whether to suspend the account or go ahead and deprovision the account. This user-defined entity operation is specific to AIX accounts. All *other* accounts are still managed by the system-defined entity type operation, which deprovisions the account and removes the user data from the remote system.

When you customize an entity type operation for a specific entity, a copy of the system-defined entity type operation is created. You do not change the system-defined entity type operation. If you want to return to the system-defined entity type definition, delete the entity operation that you created.

Security Identity Manager provides the following system-defined entity types:

Global
Specifies all entity types (Account, Business Partner Person, Person).

Account
Specifies all account types, such as Security Identity Manager user accounts, Linux accounts, or Security Identity Manager accounts.

Business Partner Person
Specifies all business partner user types, including the default business partner entity and any custom business partner entities.

Person
Specifies all person types, including the default Person entity and any custom Person entities.

For the Person and Business Partner Person entity types, Security Identity Manager provides the following system-defined operations:

Table 12. Person and Business Partner Person entity type operations

Operation	Description	Type
add	Creates a user in Security Identity Manager and enforces the policy on the new user.	Static
delete	Deletes a user from Identity Manager.	Nonstatic
modify	Modifies a user's attributes and enforces policy on the updated user.	Nonstatic
restore	Restores an inactive user.	Nonstatic
selfRegister	Creates a user in Identity Manager and enforces policy on the new user.	Static
suspend	Suspends an active user.	Nonstatic
transfer	Transfers a user from one business unit to another and then enforces policy when the transfer is complete.	Nonstatic

For the account entity type, Security Identity Manager provides the following system-defined operations:

Table 13. Account entity type operations

Operation	Description	Type
add	Creates an account.	Static
changePassword	Changes the password for an account.	Nonstatic
delete	Deprovisions an account.	Nonstatic
modify	Modifies an account.	Static
restore	Restores an inactive account.	Nonstatic
suspend	Suspends an active account.	Nonstatic

User-defined operations

User-defined operations include new operations that you create.

These operations can be either static or nonstatic, and they can be defined at any one of the following operation levels: Global, Entity Type, and Entity. Click **Configure System** > **Manage Operations** to access these operations in the IBM Security Identity Manager console.

The user-defined operations are not started directly from the Security Identity Manager system. Typically, these user-defined operations are started as an operation activity in system-defined operations to customize the default Security Identity Manager behavior. User-defined operations can also be started with life cycle application APIs directly in custom-built Security Identity Manager applications.

Operation workflow parameters

Operation workflows possess a set of input and output parameters. These parameters depend upon the operation that you select. They can also be modified.

Input and output parameters for operation workflows can be modified as follows:

- For system-defined operations, you can modify output parameters. You cannot modify input parameters.
- For user-defined operations, you can modify both input and output parameters.
- You can map existing output parameters to the relevant data and to the output parameter relevant data identifiers.
- You can add, modify, or delete output parameters. After you add an output parameter, you must map it to relevant data. The relevant data can be modified or deleted, if it is not referenced by any input or output parameters. New relevant data can be added.

User account and access request workflows

Account and access request workflows are invoked during account and access provisioning.

You typically use account and access request workflows to define approval workflows for account and access provisioning. Click **Design Workflows** in the IBM Security Identity Manager console to work on these workflows. These workflow design features are available when you select the advanced method to design workflow activities.

Account request workflows

Account request workflows provide a decision-based process to determine whether to grant the entitlement provided by a provisioning policy.

The entitlement provided by a provisioning policy specifies the account request workflow that applies to the set of users in the provisioning policy membership. There might be multiple provisioning policies that apply to the same user for the same service target. There might be different account request workflows in each provisioning policy. The account request workflow that is started for the user is determined based on the priority of the provisioning policy.

If a provisioning policy has no associated workflow and the policy grants an account entitlement, the operations that are related to the request run immediately. For example, an operation might add an account.

However, if a provisioning policy has an associated workflow, that workflow runs before the policy grants the entitlement. If the workflow returns a result of approved, the policy grants the entitlement. If the workflow has a result of rejected, the entitlement is not granted. For example, a workflow might require a manager's approval. Until the approval is submitted and the workflow completes, the account is not provisioned.

When you design a workflow, consider the intent of the provisioning policy and the purpose of the entitlement itself. For example, a provisioning policy is intended to automatically provision an intranet ID for every new person in an organization. The policy does not need an associated workflow that contains an approval activity. An approval request that might be rejected is in conflict with the intent to assign an intranet account to every new employee.

Workflows for an account contain types of information that are pertinent to the workflow: input parameters and output parameters. Input and output parameters are mapped to relevant data defined in the entitlement workflow. The mapping of input parameters to the relevant data is pre-defined. The input parameters are preselected, read-only parameters.

However, modification can occur for the mapping of the output parameters to the relevant data and the output parameter relevant data identifiers. The relevant data can be modified or deleted if it is not referenced by any input or output parameters. New relevant data can be added. Adding or modifying an account triggers an entitlement workflow if it is associated with the provisioning policy that governs the account.

An account request workflow is started during:

- New account requests from an IBM Security Identity Manager user
- Account modify requests from an IBM Security Identity Manager user
- Automatic account provisioning from the IBM Security Identity Manager system when automatic entitlement is defined in the provisioning policy
- New access requests from an IBM Security Identity Manager user, and there is no access request workflow defined for that access
- Policy enforcement from an IBM Security Identity Manager system. An account modify operation might occur because of a policy enforcement action. The account request workflow for an account is run only if the following property in the enRole.properties file is set to false:

  ```
  enrole.workflow.skipfornoncompliantaccount=false
  ```

  The default value is true.

Access request workflows

An access request workflow is started during access provisioning for a request from an IBM Security Identity Manager user.

Use the **Define an Access** task for a service group to assign an access request workflow to a specific access. The access request workflow is started for that access when the access is requested by an IBM Security Identity Manager user.

In the IBM Security Identity Manager system, an access provisioning request is mapped to an account provisioning request for a specific service. Mapping is based on the access definition, such as the service and service group of the access.

If an access request workflow exists, the workflow is run to do approval and validation of the access. The account request workflow is bypassed.

If there is no access request workflow, the applicable account request workflow is started. The workflow is based on the specification in the provisioning policy entitlement for the service to which access is provisioned.

The access request workflow is started only during an access request. It is *not* started during an account request even if the access request workflow gives the user the same access. For example, the access request workflow assigns the account to a specific group.

Entitlement workflow parameters

Entitlement workflows for accounts and access both have a fixed set of input and output parameters. The set cannot be modified by an IBM Security Identity Manager user.

The input parameters are:

Entity A relevant data item that represents the account or access that is acted upon.

For an account request workflow, the entity type for this entry is Account. If the operation is Add, this object contains the data for the new account. If the operation is Modify, this object contains the modified attributes.

For an access request workflow, the entity type for this entry is UserAccess. The entity type is inherited from Account and contains the data for a new account or changes to an existing account to provision the access. It also includes other information for the access:

- identifier
- name
- description
- owner

Service
A relevant data item that represents the service to which the account or access is associated. The entity type for this entity is Service.

Owner
A relevant data item that represents the person to whom the account or access belongs. The entity type for this entity is Person.

A single output represents the account or UserAccess entity for which the request is issued. This output parameter allows changes that are made to the account or UserAccess object that communicates with the provisioning system. For example, an RFI can be used to request that an administrator manually enters account values. The resulting account object is returned to the provisioning system with the values.

Workflow elements

Operation, account request, and access request workflows are made up of processes and activities.

Workflows are made up of one or more of these elements:

Processes
Processes define the activities and flow between activities that are needed to run a business process. Processes include these elements: activities, transitions, input/output parameters, and relevant data.

Activities
Activities represent the business logic for a specific task in a workflow process. An activity is represented in a workflow as a *node*.

IBM Security Identity Manager supports the following types of nodes:

- Approval
- Mail
- Request for Information (RFI)

- Operation
- Loop
- Extension activities
- Script
- Workorder
- Subprocess, which is available only for account request and access request workflows. You cannot include a Subprocess activity in an operation workflow.

In order to support the business logic of an activity, input and output parameters might be required. For activities, input and output parameters represent data that is passed to and returned from the activity. You use relevant data to associate the input and output parameters of an activity with any data that is stored in the process.

When sending notification messages from workflow activities, you can specify dynamic content in those messages to personalize and customize them appropriately. Dynamic content can also be used to customize the action text for a workflow activity.

Transitions

Transitions represent a flow between two activities. When a workflow process is run, the flow from one activity to another activity is controlled by the conditional logic (JavaScript coding) in the transitions and the activity configuration information. You can define both serial and parallel flows with IBM Security Identity Manager.

Input/output parameters

Input and output parameters define the data that is passed into and returned from a workflow process. Some workflow processes in IBM Security Identity Manager might restrict the customization of input and output parameters.

Relevant data

Relevant data defines the global variable data for workflow processes. You can use this variable data to pass data from one activity to the next by associating it with activity parameters. Output parameters that result from an activity are stored as a relevant data item. They are then passed from relevant data and become the input parameter for another activity.

Transitions can also access and use relevant data in their conditional logic.

Activity participants

Activity participants are IBM Security Identity Manager users who are assigned to interact with activities in a workflow process. Activities might include approvals, mail, requests for information, and work orders.

An activity participant can be any of the following users:

- A specific user with an IBM Security Identity Manager account.
- A specific user assigned to a particular organizational role.
- A user with a specific relationship, such as a supervisor or services owner.

Some workflow activities might restrict the list of available participants. Some activities (mail, work order activities) can be configured to not require the participant to be an IBM Security Identity Manager user at all.

JavaScript

Many of the workflow elements, such as transitions and script elements,

integrate the JavaScript scripting language in order to enable customization of workflow processes. In addition to the standard JavaScript extensions, IBM Security Identity Manager provides JavaScript extensions that you can use to access processes, activities, relevant data, and participants.

Common attributes for workflow activities

Each workflow activity has a set of attributes that must be configured in order for the workflow to work properly.

To show the attributes menu for a node, double-click or right-click the node and then click **Properties**. The following attributes are common for multiple node types.

Activity ID
The Activity ID is a unique identifier for the node activity. The Activity ID occurs below the node in the workflow designer and is a required field. It must be a valid JavaScript variable name, and it must start with a letter. The Activity ID cannot contain any spaces or special characters other than an underscore (_).

Activity Name
The Activity Name is an optional, descriptive field that is used for several activities in the audit logs.

Description
The Description is the optional, descriptive explanation of the activity.

Join Type
Each activity in the workflow has an attribute to define the split and the join criteria. This attribute is used to determine how the workflow engine processes multiple paths into and out of an activity. Join types define how the workflow engine processes the transitions into an activity. The Join Type is a workflow directive that synchronizes incoming transitions. The Join Type must match with a Split Type previously defined in the workflow. The following are the Join Type values that you can select:

AND The activity must wait for all active incoming transition paths to be completed before initiating the activity.

Example: You have a split defined that leads to three parallel approval activities and one of the split transition conditions evaluates to “false.” Because only two of the approvals become active, only those two paths must be completed in order for the activity that is the point of join to run.

OR The first path that leads to the activity that evaluates to true causes this activity to run. Because the order of transition evaluation is undefined, use the OR condition when only one condition evaluates to true.

Split Type
Split types define how the workflow engine processes the transitions out of an activity. The Split Type is a workflow directive that synchronizes diverging transitions. The Split Type must have a corresponding Join Type that matches its condition as the workflow progresses. The following are the Split Type values that you can select:

AND All paths that leave the activity are evaluated and all paths that resolve to true are followed.

OR The workflow engine evaluates transitions until it finds one that has a condition of true. That path is then followed. The remaining paths are not evaluated.

Manual activities (approval, RFI, work order) have the following common attributes:

Participant
This attribute defines the user that is notified to respond to a request for a manual workflow activity. A Mail activity has a **Recipient** attribute instead of **Participant**. The attribute defines the recipient type for mail. For example, a recipient might be a Person entity that has an email account, an organizational role, or a group. When defining a custom recipient, the Participant object must be used within the script for the recipient field of the mail node.

Escalation Participant
This attribute defines the user that is notified to respond to a request for a manual workflow activity if the original participant does not respond by the time period specified in the Escalation Limit attribute. Escalation participant is an optional attribute. The escalation participant is also used if there is a failure to resolve the original participant. A Mail activity has no escalation participant.

Escalation Limit
This attribute specifies the time before escalation to the escalation participant. A Mail activity has no escalation limit.

Note: Participant, escalation participant, and escalation limit are only applicable for human interactive activity types like approval, mail, RFI, and work order.

Skip Escalation
When set, the activity does not trigger an escalation when it runs. When the participant does not respond and the escalation limit is reached, the value of **Complete on Timeout** determines how the activity proceeds:

- Not set: The activity status is set to `terminated/FAILED`
- Set: The activity status is set to `completed/TIMEOUT`

.

When you select this option, any existing entries for **Escalation Participant** are disabled.

No Timeout Action
When set, there is no action performed when the activity times out. It remains in PENDING state until manually completed, the process level timeout is reached, or the process is cancelled or interrupted.

This option takes precedence over **Complete on Timeout**.

Complete on Timeout
When set, the activity runs to completion even if it reaches the timeout threshold. No timeout notification is sent. The postscript of the activity is run.

Transitions

Transitions are the pathways that connect various workflow activities. Nodes are connected with transition lines to form a workflow.

Each transition has a JavaScript component (condition) that is used to determine which transition or path to follow. The JavaScript condition must evaluate to true or false. If the condition is true, then the transition is followed, depending on the split criteria.

When two or more of transition lines connect to the same node, the join type of the node is evaluated. The join type can either be an And clause, indicating that all transitions must be active to trigger the node. The join type can be an Or clause, in which case, any one of the transitions that are active trigger the node. The join type is indicated by a symbol on the left of the node. The symbol is a triangle for the Or clause and a circle for the And clause.

The only required property for a transition line is the Condition.

Start and end nodes

The start node defines the beginning of a workflow and the end node defines the end of a workflow.

Start and end nodes are always included in a workflow and cannot be deleted. These nodes each contain a JavaScript window in which you can add JavaScript code to run at the beginning or end of the workflow. Start nodes have out transitions only. End nodes have transitions in only.

Approval node

Use the approval node to add a request for approval when adding or modifying people, accounts, and access.

The approver must be an IBM Security Identity Manager user. The approver is required to log in to IBM Security Identity Manager to approve or reject the request.

In entitlement workflows, use approval nodes to request authorization to continue with a provisioning request. In operation workflows, use an approval node as a switch to follow a specific workflow path.

Approval text and labels can also be customized to allow approvals to be used for most Yes/No decision activities.

Mail node

Use the mail node to specify the recipient type and content to be emailed to a user in an email notification.

The content can be specified directly or copied from a template used by mail activities in other workflows.

Request for information node

Use the request for information (RFI) node within entitlement and operation workflows to solicit account or user-related information from a user with an IBM Security Identity Manager account.

Within the RFI, you specify the attributes for which the participant is being asked to provide values. Only the attributes that you select are editable by the participant. All other form attributes are read-only. The page in the Activities to-do list matches the form specified with the form designer. Attributes that are listed as

mandatory on the account form are also mandatory for the RFI. ACI definitions do not need to be created for the fields that the participant is asked to respond to.

You must select an entity type and entity for the RFI to be used to request information about attributes for a specific entity. After the entity type is selected, you select from a list of attributes. The attributes selected occur on the RFI page when the participant logs in and accesses the RFI activity list item.

For example, if you select Account, then the account parameter must contain an entity for an account, such as ITIMAccount. If you select Person for the Entity type, the parameter must be for an Entity, such as Person or BPPerson. The list of entity or account objects vary based on your implementation. For each account profile, you have an entity listed when the account type is selected. For each defined user, you have an entity listed when the Person or BPPerson type is selected.

Operation node

Use the operation node to invoke an existing operation from within a workflow.

The operation activity type can be Global, which requires that you specify an operation. If the activity type is Data Reference, you must specify the data as Entity, Service, or Owner. If the activity type is Expression, Static, or Dynamic, select an Account, Business Partner Person, or Person entity type. Depending on your choice, also specify subordinate attributes, such as a specific entity, expression, and operation.

The operation must be predefined for an entity type or entity. Operations take input parameters, but they do not return anything to the calling workflow.

The Activity ID and Operation properties are required.

Loop node

Use the loop node to execute one or more nodes in a loop. Nested loops are not supported.

There are two types of loops:

Do while
: This loop evaluates the condition before running. If the condition is true, the loop runs, otherwise it continues with the next node. This loop type is used when the condition is dependent on the workflow processes that occur before the loop node. The process defined by the loop does not run if the condition is already met.

Do until
: This loop will evaluate the condition after each execution of the loop nodes. The workflow completes the process defined in the loop before checking the loop condition. If the condition is true, the loop runs again, otherwise it continues with the next node after the loop. This loop type is used if the process defined by the loop must run at least one time regardless of any previous activities.

You can also set the following option:

Asynchronous Processing of the Loop Body
: When set, the loop runs unblocked. Each iteration of the loop runs without

waiting for the previous iteration to finish. If not set, then each iteration of the loop must finish before the next iteration of the loop can start.

Nodes contained within the loop must not change to any activities outside the loop. All transitions must originate to and from the loop node. The standard rules for split and join apply for loops in terms of multiple transitions in and multiple transitions out. The loop node does not specify the results of the nodes in the loop. You must check the status of nodes in the loop in a script that follows the loop, if required.

An activity in the loop can run multiple times (one time for each loop iteration). The workflow engine tracks the activities in a loop by giving them an index that represents to which iteration of the loop the instance of the activity applies. This index is stored in the activity object as a member variable called **index**. For activities that are not in a loop, this index is set to 0. For activities in a loop, this value is 1-*n*, where *n* represents the number of actual loop iterations that run.

To retrieve an instance of an activity in a loop, use the process.getActivity() method. This method takes two arguments: the activity ID and the index of the target activity instance. If you use this method to retrieve an activity that is not part of a loop, you can either omit the index argument or set it to 0. The method returns an activity object.

The condition defined in the loop specifies the number of loop iterations. The global variable, **loopcount**, can be used to identify the current iteration of the loop. The loopcount variable starts with 1 (loopcount=1) and increments each time that the loop is run. This variable can be useful for creating loop conditions, for example, `loopcount<x`.

Nodes placed in the loop determine the amount of time each loop takes. The loop node does not have a built-in wait mechanism.

The activity ID and the loop condition are the only required properties for the loop node.

Get activities within a loop

Use the getActivity() call to retrieve an activity.

For example, the following call retrieves the supervisor approval activity associated with the third iteration of the loop:

```
theActivity = process.getActivity("SupervisorApproval",3);
```

The following call retrieves the service owner approval activity that is not part of a loop:

```
theActivity = process.getActivity("ServiceOwnerApproval");
```

An instance of an activity is not available for retrieval with process.getActivity() until the activity actually runs in the loop. You might have a loop condition that specifies that a loop runs three times *or* until another condition is met. It is possible that the third iteration of the loop does not run. In this case, a script that runs the following line of JavaScript code after the loop returns a null value.

```
theActivity = process.getActivity("SupervisorApproval",3);
```

Accessing the Activity with a null value causes errors to occur.

Extension node

Use the extension node to start an application extension from within the workflow.

The application extension is a Java™ class that is configured for the workflow environment. Extensions can accept input parameters and return output parameters back to the workflow. Only extensions that are properly registered occur in the extension window.

System-defined extensions define basic, atomic services provided by IBM Security Identity Manager. These building blocks are the components with which standard operation workflows are built. Workflow extensions make it easy to incorporate core IBM Security Identity Manager services and functions into various IBM Security Identity Manager workflows.

Although operation extensions are components of operation workflows, they can also be used outside of an operation workflow. They can also be called in entitlement workflows.

For example, assume that during account provisioning that you want to run a change to the user record of the account owner in the IBM Security Identity Manager user store. You can include the **modifyPerson** extension into the entitlement workflow to accomplish this change. If you want to have the business logic associated with the modify user operation, you can also include the **modifyPerson** operation in the entitlement workflow. For more information about including an operation into an entitlement workflow, see the operation node.

You can also create your own custom application extensions and add them to a workflow.

Script node

Use the script node to add logic to the workflow through the use of JavaScript code.

The Script node makes clear to anyone who views the workflow that scripting is present in the workflow.

JavaScript code is used within workflows to dynamically define and retrieve parameter and attribute values. JavaScript code is also used to store and forward these values as variables for use by logic or code within a single workflow activity.

The JavaScript code can be extended by defining custom JavaScript objects through a Java extension.

When using the Script node, carefully consider the overhead associated with running additional activities in a workflow versus using post-activity execution scripts on existing activities. Also consider combining multiple sequential script activities into a single script node to avoid additional workflow transaction overhead.

The Activity ID is the only required property for the Script node.

Work order node

Use the Work order node to send email to an IBM Security Identity Manager user. The email either requests some type of manual activity or is a simple notification.

The work order activity supports two execution modes: send mode and send and wait for completion. The send mode completes the activity when the work order request messages are successfully sent to the mail server for forwarding to participants. The send and wait for completion mode sends the email and then waits for notification of the completion of a manual activity.

The activity ID and Participant properties are the only required properties for the work order node.

Work order workflow participant resolution

Work orders interact with users differently, based on the setting for send or send and wait for completion. The setting also changes the behavior of participant resolution. In the send-only mode, only participants with email addresses are resolved. Send mode emails only a notification. If no email address is specified for the user, the user is not a valid workflow participant.

If the send and wait for completion option is specified, an email is sent. The work order waits to be completed through the activity list or the external API. Participant resolution considers users who own IBM Security Identity Manager accounts and users who do not own Security Identity Manager accounts but have email addresses. If a resolved user has a Security Identity Manager account, a work order entry is added to the user's activity list. Email notifications are sent to users with Security Identity Manager accounts and an email address. Notifications are also sent to users with only an email address, but no Security Identity Manager accounts.

The **Continue on Participant Resolution Failure** option is available in the Work Order node of the user recertification workflow designer. If you select this option, the workflow process continues in the event of a participant resolution failure.

Subprocess node

Use the subprocess node to execute one entitlement workflow from another.

Subprocesses simplify the workflow by using one node to represent a previously defined workflow sequence.

Note: The Subprocess node is not available in Operation workflows.

Subprocess nodes are typically used for the following reasons:

Organization
Workflows can become complicated when there are many parallel branches that must run. Placing large blocks of processing into a subprocess can sometimes help.

Reusability
Subprocess workflows that define generic processing can be included in multiple workflows.

A subprocess can use any predefined entitlement workflow of the same service type or global workflows. However, the workflow must be located within the same organization.

You must map the parameters required by the child workflow to relevant data items in the parent workflow. The subprocess node provides the set of input and output parameters for the Subprocess. While the parameter names and types are fixed by the subprocess node, you can configure the value of each parameter.

The Activity ID and the Subprocess property are the only required properties for the subprocess node.

Workflow data

Workflow data consists of the information that is input to and output from the activities that comprise the workflow.

The following types of data are used to build workflows:

- Relevant data objects
- JavaScript variables
- Workflow context objects

Relevant data

Relevant data defines global variable data for the workflow process.

Relevant data is used to pass data from one activity to the next by associating it with activity parameters. Relevant data can be accessed through JavaScript for use in transition conditional logic, script activities, and post-activity scripts.

Relevant data consists of the following attributes:

ID The relevant data ID is its name, for user. This ID is used to reference the relevant data item when associating it to activity parameters or accessing it in JavaScript. The ID cannot contain any spaces or special characters other than an underscore (_).

Type The relevant data container is the type of data. For example: Person, Account, Distinguished Name, List, Integer, and String.

Entity Used with certain relevant data types to specify the profile type. For example, with Account-type relevant data the Entity might be ITIMAccount.

Element Type
Used with the List type to specify what type of items are stored in the list.

You might want to enhance lifecycle operations with additional functionality. You might add additional relevant data items and JavaScript code to retrieve values for those items from the directory. For example, assume that you want to transfer a user to another location within the organization whenever a **modifyPerson** operation is done that changes an attribute related to the user's job function. You would need to define relevant data that applied to the organizational container in which you were placing the user.

Input and output parameters

Workflow input parameters are data items that are used to do a function. Workflow output parameters are data items that result from an activity.

Workflows are processes that are a series of activities. Activities require input data on which to act. Activity output can be mapped to the input of another activity through relevant data.

Both input and output parameters are supported in operational workflow and entitlement workflow processes.

Parameters for workflows must be mapped to a workflow or relevant data item. The values for these parameters can then be used as parameters for a subsequent workflow activity.

System-defined data

System-defined data are default global data items (input parameters and relevant data) that are defined for the workflow engine.

For example, the add user operation defines input parameters to create the user and specify the organization tree level container in which to create them. These data items are available to the entire workflow. Any activities and nodes in the workflow can use them.

System-defined input parameters are predefined for each default operation, and values are set for you before execution of the workflow. However, while you can use JavaScript code to modify these values, you are not allowed to create new input parameters for the entire workflow.

System-defined relevant data can be predefined, and values are set for you before execution of the workflow. You can use JavaScript code to modify these values and you are allowed to create new relevant data for the entire workflow. However, you cannot remove existing system-defined relevant data items or you risk corrupting the default business logic behind the workflow.

User-defined data

You can create new relevant data items to use within a workflow to hold data that might be used in more than one activity. Relevant data items might receive the output of an application extension.

To store data in the relevant data items, you must first create the relevant data item with the appropriate data type. You can then optionally initialize the data item with JavaScript code, but you are not required to do so. Relevant data can also be populated by mapping it to the output parameter of an activity.

Context of relevant data items

When you define a relevant data item, you can associate it with a context.

The context of a relevant data item specifies the entity to which the item applies. The context of a relevant data item can be used to determine workflow participants and is useful for audit purposes. You define the context of a relevant data item when you define the item. The possible values for context are:

Not Applicable
Specifies that this data item has no context.

Subject
Identifies that this data item is the subject of the workflow. You can define only one subject per workflow. For example, in an entitlement workflow for:

- Account, the account is the subject.
- Access, the access is the subject.

Requestee
Identifies that this data item is the user to whom this workflow applies. The requestee for an entitlement workflow is the account owner. You can define only one requestee for a workflow.

Both Identifies that this data item applies to both the subject and requestee context.

Workflow data in JavaScript code

Data items that represent IBM Security Identity Manager entities are represented in JavaScript code as DirectoryObject objects.

DirectoryObject objects have methods and instance variables that can be accessed in JavaScript code.

The data item types that represent Security Identity Manager entities include the following types:

- Account
- Admin Domain
- Business Partner
- Business Partner Organization
- Business Partner Person
- Business Unit
- DirectoryObject
- Host Selection Policy
- Organization
- Organizational Container
- Organizational Role
- Person
- Provisioning Policy
- Service

Workflow data persistence

When you call a get method on a relevant data item, you load the data from the persistent data store into an in-memory cache. If another activity in the workflow calls a get method on the same data item, it uses the same cached copy.

When you use the addProperty, setProperty, or removeProperty methods, you update the cached copy of the data. The set method writes your changes from the cache back to the persistent data store.

The addProperty, setProperty, and removeProperty methods do not automatically write their changes to persistent storage for two reasons: performance and data integrity. There is a performance cost associated with each write process. To minimize overhead, a single write process can be executed after all attributes have been modified. Think of the set method as a database transaction commit. You would not want the persistent storage to be updated with only a portion of your changes if the server suddenly became unavailable. Using the set method as a transaction commit allows all or none of the changes to be committed to persistent storage.

Workflow context objects

You can access a number of system objects directly from JavaScript code without using a get method. These system objects are referred to as context objects.

Workflow context objects are exposed to the workflow scripting environment as global JavaScript objects. Other JavaScript contexts have global data available to them, but only the following global objects are related to workflow:

Activity
This object contains the information related to the current activity.

Process
This object contains the information related to the current process.

Workflow participants

A workflow might contain activities that require manual intervention or input by a user before it can be completed. This user is called a workflow *participant*.

When an activity requires an action before it can continue, an activity to-do item is assigned to the participant. The item occurs in the Activities to-do list of the participant, who can view the item the next time the participant logs on to IBM Security Identity Manager. This to-do item might consist of a participant who approves or rejects a request. The to-do item might provide the activity with information for it to process, or complete a manual task outside of the system. When a participant completes the to-do item, the activity can proceed.

Workflow participants must have an IBM Security Identity Manager account to respond to approval requests and requests for information.

To respond to work order requests that require completion before the workflow continues, a participant must have a valid email address or an IBM Security Identity Manager account.

For each activity that requires a participant, you can also specify an escalation participant.

When you specify an escalation participant, you define a time limit that is used to determine when the request is escalated. If the original participant cannot be resolved or does not respond within the specified time, the escalation participant is notified. If for any reason the participant and the escalation participant cannot be resolved, the System Administrator group becomes the workflow participant.

A manual activity has performance considerations if you select a participant of type ITIM Group or Organizational Role. A group or role with a large membership, greater than 1000 members, might cause problems when the system creates activities for each individual.

Additionally, you might redesign a workflow process that authorizes many individuals to approve an activity. For example, you might select the participant for a workflow activity. An approval that goes to an ITIM Group or an Organizational Role sends the approval to each member of that group or role. The activity is completed as soon as any one member of the group or role (in this example, containing over 1000 members) acts on the approval.

Workflow participant types

Workflow participants can be specified by the participant type, including system-defined participant types and custom (user-defined) participant types.

IBM Security Identity Manager provides the following types of workflow participants:

Person (with email account)
The participant is a specific user who has an email account and must respond to the request.

Person (with ITIM Account)
The participant is a specific user who has an ITIM Account and must respond to the request.

Organizational Role
The participant is a specific organizational role.

The role might be a child role of another organizational role, which then becomes a parent role. The child role inherits the permissions of the parent role. Therefore, the notification is sent to both the organizational role and its child roles. All user members of the organizational role and its child roles are eligible to respond. The first response from any user member or child role triggers the workflow to continue.

If you need to delete an organizational role, first check that no workflow definitions use the role. This check avoids any possible problems with participant resolution.

ITIM Group
The participant is a Security Identity Manager group. All members of the group are notified, and all members are eligible to respond. The first response triggers the workflow to continue.

Requestor
The participant is the user that initiated the request.

Requestee
The participant is the user or account that is acted upon by the request. For accounts, the account owner is generally the requester.

Service Owner
The participant is the owner of the service, as specified in the relevant data account object.

Access Owner
The participant is the owner of the access, as specified in the relevant data account object. This type is only available in the access request entitlement workflow designer.

Role Owner
The participant is the owner of the role, as specified in the corresponding relevant data. This type is only available in the workflow designer for operations.

Sponsor
The participant is the person who is designated as the sponsor for the requester. Sponsor is only applicable to business partners.

Manager
The participant is the supervisor or manager of the requester. If a manager is not specified for the requester, the manager who is designated on the organizational container of the requester is the participant. If no manager is specified for the organizational container, the next level up in the organizational tree is checked for a manager. The tree is checked until the top of the organization is reached.

System Administrator
The participant is the System Administrator group. One of the members of the System Administrator group is required to act before the activity can continue.

Domain Administrator
The participant is the domain administrator of the organizational container that is associated with the subject of the request.

- If the subject is an Account, UserAccess, or Service, then the service to which the subject refers is used to determine the organizational container.
- If the subject is not Account, UserAccess, or Service, then the requester of the workflow process is used to determine the organizational container.

Custom Participant
The participant is determined with a JavaScript expression. The JavaScript expression must return one of the system-defined participant types.

Custom workflow participants

You can define custom workflow participants with JavaScript code.

A user-defined workflow participant is a custom participant type. To implement a custom participant or a custom recipient in the case of a mail node, you must create an object of type Participant.

Self-approval for requester

When the `enrole.workflow.selfapproval` is set to true, requesters can approve their own requests.

The configuration property `enrole.workflow.selfapproval` was added to the *ITM_HOME*`/data/enrole.properties` file. If this property is set to `true`, the workflow routes an approval request to the requester. The requester must be listed as an approver or as a member of an approval group. The default value is `false`.

This property works with two other properties.

`enrole.workflow.skipapprovalforrequester`

`enrole.workflow.disablerequesterapproval`

The following tables illustrate how the properties interact. Because IBM Security Identity Manager evaluates `enrole.workflow.disablerequesterapproval` only when both `enrole.workflow.skipapprovalforrequester` and `enrole.workflow.selfapproval` are false, only five combinations exist.

Table 14. Default settings

Property	Value	Results
enrole.workflow.selfapproval	False	When the requester is the sole approver, the request is automatically approved. When there are other approvers, the request is routed to the other approvers, and the requester is skipped. If the requester has a delegate, the delegate is not skipped. The request is routed to the delegate and other approvers.
enrole.workflow.skipapprovalforrequester	False	
enrole.workflow.disablerequesterapproval	False	

Table 15. Variation 1

Property	Value	Results
enrole.workflow.selfapproval	False	When the requester is the sole approver, the request is automatically approved. When there are other approvers, the request is routed to other approvers and the requester is skipped. When the requester has a delegate, the delegate is skipped. If there are other approvers, the request is routed to other approvers. If the delegate is the sole approver, the delegate is skipped, which results in an "Unresolved Participant" condition. The request is escalated.
enrole.workflow.skipapprovalforrequester	False	
enrole.workflow.disablerequesterapproval	True	

Table 16. Variation 2

Property	Value	Results
enrole.workflow.selfapproval	True	When the requester is the sole approver, the request is routed to the requester for approval. If there is more than one approver, the request is routed to each approver separately. If the requester has a delegate, the request is routed to the delegate instead of the requester and to other approvers.
enrole.workflow.skipapprovalforrequester	False	

Table 17. Variation 3

Property	Value	Results
enrole.workflow.selfapproval	False	When the requester is the sole approver, the request is automatically approved. When there are other approvers, they are skipped and the request is automatically approved. If the requester has a delegate, the delegate is not skipped. The request is routed to the delegate and other approvers.
enrole.workflow.skipapprovalforrequester	True	

Table 18. Variation 4

Property	Value	Results
enrole.workflow.selfapproval	True	When the requester is the sole approver, the request is routed to the requester. When there are other approvers, they are skipped. If the requester has a delegate, the request is routed to the delegate.
enrole.workflow.skipapprovalforrequester	True	

Customized self-approval for requestee and requester

Administrator can define the customized self-approval by requester and requestee by using JavaScript code.

In a typical deployment, self-approval by requester and requestee is disabled by using the configuration setting in the `enrole.properties` file. For more information about how to disable or enable self-approval, see "Disable requestee or requester approval" on page 54.

Using JavaScript code, administrator can enable self-approval by requestee or requester for specific workflow even though the global configuration is set to disable the self-approval. For more information about method to enable self-approval for specific workflow, see Participant.

Skip approval for requester property

In some cases, the requester of an activity can also serve as a workflow participant.

If the requester of an activity is also designated as a workflow participant, the approval request for the requester is skipped by the workflow unless the `enrole.workflow.selfapproval` property is set to true.

Use the `enrole.workflow.skipapprovalforrequester` property to specify whether other participants are skipped. The `enrole.workflow.skipapprovalforrequester` property is defined in the `enRole.properties` file.

If the property is set to `true`, IBM Security Identity Manager skips approval for other resolved participant users. Skipping approval occurs if the following conditions are met:

- The requester, the user who submitted the request, is identified as one of the participant users as the result of participant resolution.
- The `enrole.workflow.selfapproval` property is set to false.

The approval is skipped.

If the property is set to `false` and the `enrole.workflow.selfapproval` property is set to false, the requester is skipped. The approval still goes to the other resolved participants. If the requester is identified as single approver as the result of participant resolution, then the approval is skipped.

Disable requestee or requester approval

In some cases, the requester or requestee of an activity can also serve as a workflow participant for an approval.

The `enrole.workflow.disablerequesteeapproval` and `enrole.workflow.disablerequesterapproval` properties are defined in the `enRole.properties` file. Use these properties to specify whether to skip the requestee or requester from the approver list, if the requestee or requester is also an approver. These properties are applicable only for normal and recertification approvals.

enrole.workflow.disablerequesteeapproval

Use the `enrole.workflow.disablerequesteeapproval` property to specify whether to remove the requestee and its delegate from the approver list. This property is applicable if the requestee (the user for whom the request is initiated) is also an approver. The default value of the property is `false`.

When the value is `false`, the approval request then goes to the approvers, including the requestee, if the requestee is also an approver.

If you set the property to `true`, IBM Security Identity Manager does the following actions:

- Removes the requestee and its delegate from the approver list if the requestee is identified as an approver as a result of participant resolution. The approval goes to the other approvers.
- Considers the participant resolution as "Unresolved Participant" if the requestee or its delegate is a single approver or a single escalation approver. The request is then sent to the escalation participant if it is defined and is not escalated already. Otherwise, the approval request is sent to a system administrator group member.
- Removes the requestee and its delegate from the approver list even when the user is a system administrator group member.
- Precedes over the `enrole.workflow.skipapprovalforrequester` property if the `enrole.workflow.skipapprovalforrequester` property is also set to `true`. The requestee and its delegate are removed from the approver list if the requester and requestee are the same, including the request approver. The approval process is not skipped.

Example

```
enrole.workflow.disablerequesteeapproval=true
```

enrole.workflow.disablerequesterapproval

IBM Security Identity Manager considers the `enrole.workflow.disablerequesterapproval` property value only when the `enrole.workflow.skipapprovalforrequester` and the `enrole.workflow.selfapproval` property values are set to `false`. The `enrole.workflow.disablerequesterapproval` property specifies whether to disable the requester approval when the activity requester is the single approver. The default value of this property is `false`.

When the value is `false`, the Security Identity Manager works in accordance with the value that you set for the `enrole.workflow.skipapprovalforrequester` property. The `enrole.workflow.skipapprovalforrequester` property behavior does not change.

Use the `enrole.workflow.disablerequesterapproval` property to specify whether to remove the requester and its delegate from the approver list. This property is applicable if the requester (the user who submitted the request) is also an approver.

If you set the `enrole.workflow.skipapprovalforrequester` property to `false` and the `enrole.workflow.disablerequesterapproval` property to `true`, the Security Identity Manager does the following actions:

- Removes the requester and its delegate from the approver list if the requester is an approver as a result of participant resolution. The approval goes to the other approvers.
- Considers the participant resolution as "Unresolved Participant" if the requester or its delegate is a single approver or a single escalation approver. The request is then sent to the escalation participant if it is defined and is not escalated already. Otherwise, the approval request is sent to a system administrator group member.
- Removes the requester and its delegate from the approver list even when the user is a system administrator group member.

Example

```
enrole.workflow.skipapprovalforrequester=false
enrole.workflow.disablerequesterapproval=true
```

Skip delegation when requestee is the delegated approver

When an approver delegates the approval activity, the delegatee in some cases might end up to be the requestee in the request.

Use the **enrole.workflow.skipdelegateforrequestee** property to specify whether the delegation is skipped. The **enrole.workflow.skipdelegateforrequestee** property is defined in the `enRole.properties` file.

If the property is set to `true`, IBM Security Identity Manager skips the delegation if the following conditions are met:

- The delegatee, the user whom the approval is delegated, is the requestee .
- The **enrole.workflow.skipdelegateforrequestee** property is set to true.

If the property is set to `false`, the delegation is not skipped.

Notification of failed workflow requests

IBM Security Identity Manager sends email alerts for all failed workflow requests, including reconciliation, to the requester.

The email notification is sent to the system administrator for system-generated processes, such as scheduled reconciliations.

Index

S

T

U

V

W

Printed in USA