CA Identity Manager™

Configuration Guide

12.6.4

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

CA Technologies Product References

This document references the following CA products:

- CA Identity Manager
- CA Siteminder®
- CA Directory
- CA User Activity Reporting
- CA Identity Governance

Contents

Chapter 4: Relational Database Management 97

Chapter 5: CA Identity Manager Directories 145

Chapter 7: Advanced Settings 215

Chapter 8: Auditing 229

Chapter 1: Introduction to CA Identity Manager Environments

This section contains the following topics:

CA Identity Manager Environment Components

A CA Identity Manager *Environment* is a view of a management namespace that lets CA Identity Manager administrators manage objects like users, groups, and organizations. These objects are assigned with a set of associated roles and tasks. The CA Identity Manager Environment controls the management and graphical presentation of a directory.

A single user store can associate multiple CA Identity Manager Environments (see page 14) to define different views of the directory. However, a CA Identity Manager Environment is associated with only one user store.

CA Identity Manager environments contain following elements:

Directory

Describes a user store to CA Identity Manager. Directory element includes:

- A pointer to a user store, which stores managed objects such as users, groups, and organizations.
- Metadata that describes how managed objects are stored in the directory and its representation in CA Identity Manager.

Provisioning Directory (optional)

Stores data relevant to the Provisioning Server to manage additional accounts in managed endpoints. Only one Provisioning Directory can be associated with an Environment.

Note: For more information about the Provisioning Server or the Provisioning Directory, see the *Installation Guide*.

User Console

Enables CA Identity Manager administrators to perform tasks in a CA Identity Manager Environment.

Task and role definitions

Determine user privileges in CA Identity Manager and other applications. These task and role definitions are initially available in the CA Identity Manager Environment where they can be assigned to users.

You can customize the default roles and tasks using the User Console.

Self-service

Lets users create and maintain their own accounts for accessing resources, such as a customer web site. Self-service also lets users request a temporary password in case of forgetting the current password.

Workflow definitions

CA Identity Manager includes default workflow definitions that automate approval and notification of user management tasks, such as creating user profiles or assigning users to roles or groups. You can modify the default workflow processes in CA Identity Manager to support each enterprise requirements.

Skins

Determine the appearance of the CA Identity Manager user interface.

Custom features

You can modify CA Identity Manager to suit your business requirements using the CA Identity Manager APIs. See the *Programming Guide for Java*.

Each CA Identity Manager environment requires one or more system managers to customize the initial roles and tasks using the User Console. Once a system manager creates the initial roles and tasks, that manager can grant administrative privileges to users in that environment. These users become administrators who manage users, groups, and organizations. See the *Administration Guide*.

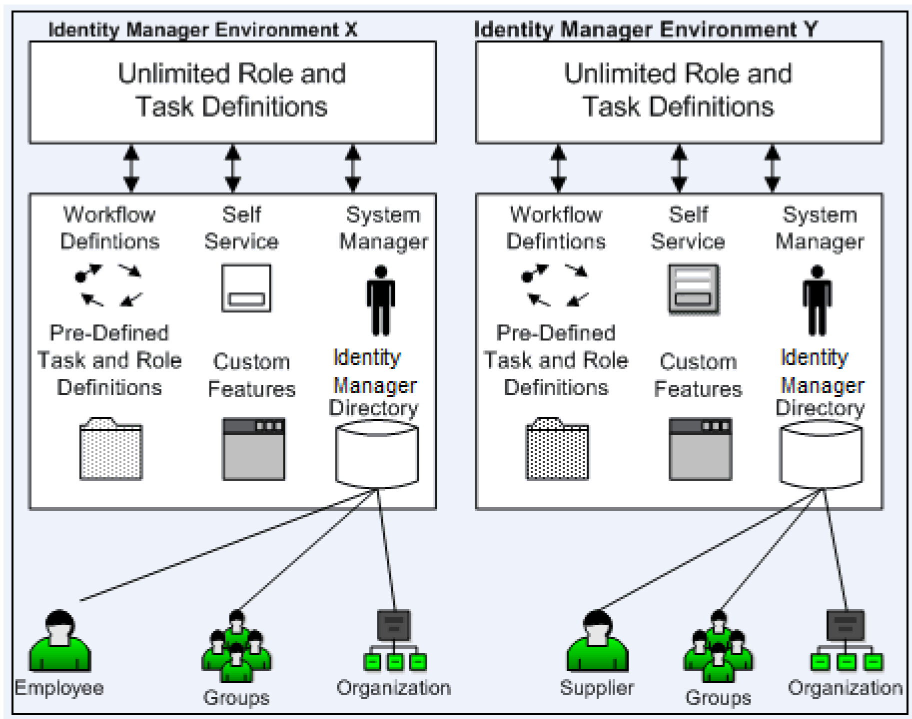
Multiple CA Identity Manager Environments

Create multiple CA Identity Manager environments when you want to:

Manage additional user stores—You can manage users in different types of user stores. For example, your company store all of its user profiles in a Sun Java System LDAP directory. You enter into a joint venture with a partner that uses an Oracle database to store user information. You would want a different CA Identity Manager environment for each set of users.

- Manage objects with different LDAP object classes— Consider CA Identity Manager is managing an LDAP directory. Within the same directory, you can manage objects of the same type along with different object classes and attributes. For example, the following illustration shows a directory that contains two types of users:
  - Employees, who have an employee ID number.
  - Suppliers, who are identified with a supplier number.

Equation 1: Diagram showing example of two Identity Manager environments with directories containing employees and suppliers.



CA Identity Manager Management Console

As a CA Identity Manager system administrator, your responsibilities include:

- Creating a CA Identity Manager Directory
- Configuring a Provisioning Directory
- Configuring a CA Identity Manager environment
- Assigning a system manager
- Enabling custom features for initial use

To configure a CA Identity Manager environment, use the Management Console, a web-based application.

The management console is divided into the following two sections:

- Directories—Use this section to create and manage CA Identity Manager Directories and Provisioning Directory, which describe user stores to CA Identity Manager.
- Environments—Use this section to create and manage CA Identity Manager environments, which control the management and graphical presentation of a directory.

How to Access the CA Identity Manager Management Console

To access the Management Console, enter the following URL in a browser:

http://*hostname:port*/iam/immanage

hostname

Defines the fully qualified domain name or IP address for the server where CA Identity Manager is installed.

Note: If you are accessing the Management Console using Internet Explorer 7 and the hostname includes an IPv6 address, incorrect display of the Management Console is expected. To prevent this issue, use the fully qualified hostname or an IPv4 address.

port

Defines the application server port.

Note: If you are using a Web Agent to provide advanced authentication for CA Identity Manager, you do not need to specify the port number.

Note: Enable Javascript in the browser that you use to access the Management Console.

Example paths to the Management Console:

- For Geologic Weblogs:

  http://myserver.mycompany.org:7001/iam/immanage

- For JBoss:

  http://myserver.mycompany.org:8080/iam/immanage

- For WebSphere:

  http://myserver.mycompany.org:9080/iam/immanage

How to Create a CA Identity Manager Environment

To create a CA Identity Manager environment, you complete the following steps in the Management Console:

1. Use the Directory Configuration Wizard (see page 146) to create a CA Identity Manager Directory.
2. If your environment includes provisioning, use the Directory Configuration Wizard again to create a Provisioning Directory (see page 160).
3. Create a CA Identity Manager Environment.
4. Access the Environment (see page 181) to verify that it is running.

Chapter 2: Sample CA Identity Manager Environment

This section contains the following topics:

Overview of Sample CA Identity Manager Environment

CA Identity Manager includes a sample environment that you can use to learn about and test CA Identity Manager.

The sample environment is based on a car trading company named NeteAuto. NeteAuto administrators use CA Identity Manager to manage employees, suppliers, and regional dealerships.

User store configurations to use sample NeteAuto environments are:

- LDAP user stores that support organizations
- LDAP user stores that do not support organizations.
- Relational Database user stores that support organizations
- Relational Database user stores that do not support organizations.

Note: Provisioning capabilities are unavailable since this environment has no provisioning directory.

How to Configure the NeteAuto Sample with Organization Support

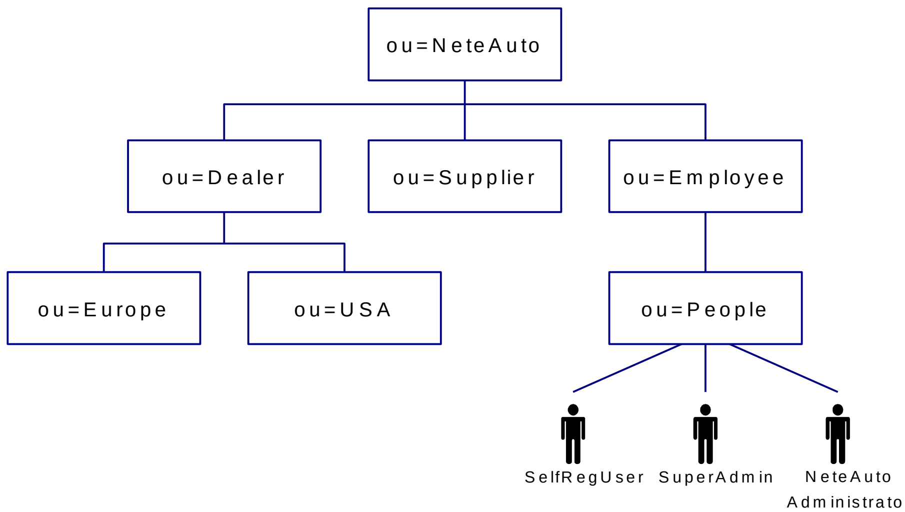
Configuring the NeteAuto sample with organization support involves the following steps:

- Installing the prerequisite software
- Installing the sample CA Identity Manager environment

- Configuring an LDAP user directory
- Configuring a relational database
- Creating the CA Identity Manager directory
- Creating the NeteAuto CA Identity Manager environment

LDAP Directory Structure for NeteAuto

The following illustration describes the NeteAuto sample for LDAP directories:



The sample CA Identity Manager Environment includes the following users:

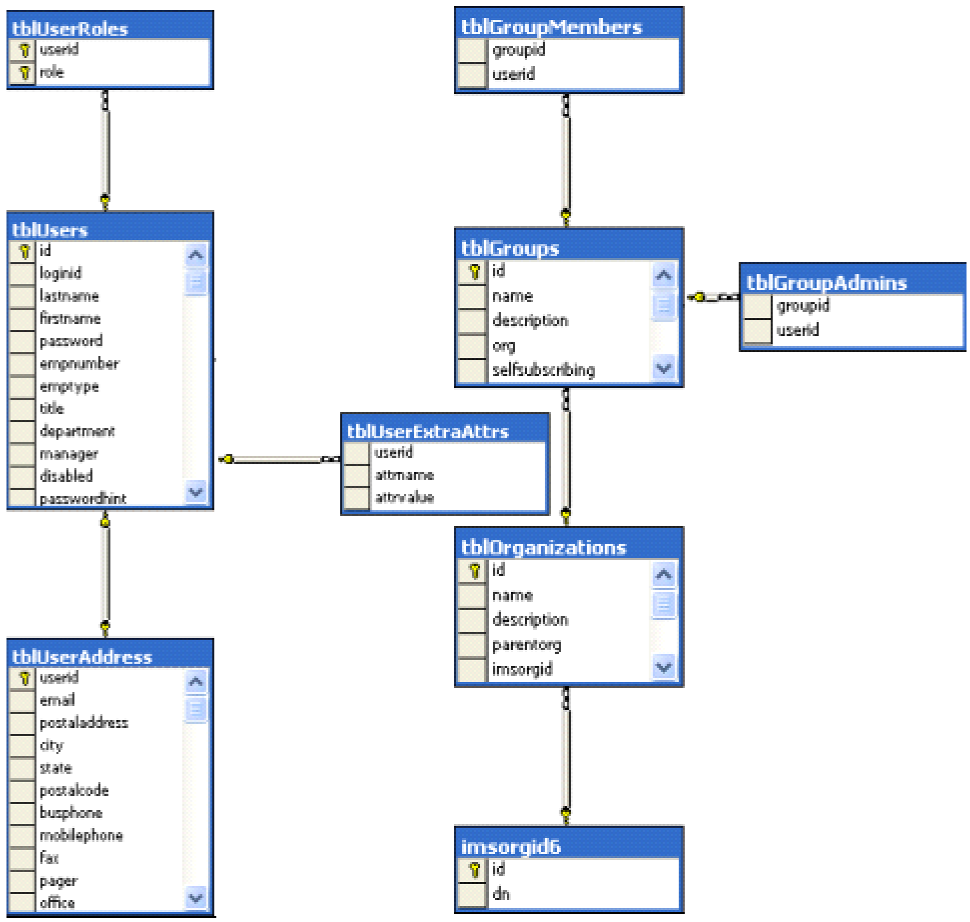
- Superadmin is the administrator account with the System Manager role for this CA Identity Manager Environment. As superadmin, you can perform all default admin tasks.

  Note: For a description of the default admin tasks, see the *Administration Guide*.

- SelfRegUser is the administrator account that CA Identity Manager uses to enable self-registration for this CA Identity Manager Environment.
- NeteAuto Administrator does not have any privileges when you install the NeteAuto environment. However, you can assign Group Manager as a user role, as described in Assign the Group Manager Role.

Relational Database for NeteAuto

The following illustration describes the relational database for the NeteAuto sample including an organization table:



Prerequisite Software for NeteAuto

The NeteAuto CA Identity Manager environment has the following prerequisites:

- Install CA Identity Manager as described in the *Installation Guide*. Be sure to install the CA Identity Manager Administrative Tools.
- You must have access to a Sun Java system (Sun ONE or iPlanet) Directory Server or a Microsoft SQL Server database.

Installation Files for the NeteAuto Environment

CA Identity Manager includes a set of files that you can use to set up a sample CA Identity Manager environment. CA Identity Manager environment is a view of a management namespace that enables CA Identity Manager administrators to manage objects such as users, groups, and organizations. These objects are managed along with a set of associated roles and tasks. CA Identity Manager environment controls the management and graphical presentation of a directory.

The sample CA Identity Manager environment includes:

- Sample objects, such as users and organizations
- Role, task, and screen definitions

  Tasks appear in the User Console when you click a tab, such as Users or Groups. Based on the assigned roles, the associated tasks appear when the user logs in.

  Note: For more information about roles and tasks, see the *Administration Guide*.

- A sample skin that customizes the User Console for NeteAuto users.
- A directory configuration file that you use to create a CA Identity Manager directory.

The files for creating the sample CA Identity Manager environment are installed in the following location:

`admin_tools\samples\NeteAuto`

In this path, *admin_tools* refers to the Administrative Tools. The Administrative Tools are placed in the following default locations:

- **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

Install the NeteAuto Environment

Perform the following process to install the NeteAuto environment.

Follow these steps:

1. Make sure that the prerequisite software is installed (see page 21).
2. Configure the user store and import the sample data.
   - For LDAP users: Configure an LDAP User Directory (see page 23)
   - For relational database users: Configure a Relational Database

3. Create the NeteAuto CA Identity Manager directory.
4. Create the NeteAuto CA Identity Manager environment.
5. Configure the look and feel of the CA Identity Manager user interface for NeteAuto users (see page 37).

Configure an LDAP User Directory

LDAP directory is available depending upon your installation. You can use the following procedure to check whether the directory exists or to create the directory.

Follow these steps:

1. In the directory server console, create an instance of LDAP with the following root:

   ```
   dc=security,dc=com
   ```

   Write down the port number for future reference.

2. Import the NeteAuto.ldif file to the directory server from samples\NeteAuto in the Administrative Tools.

   The Administrative Tools are installed in the following default locations:

   - **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
   - **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

Note: If you experience problems importing the LDIF file or creating the CA Identity Manager directory, add the following text to the beginning of the LDIF file:

```
dn: dc=security, dc=com
objectClass: top
objectClass: domain
dc: security
```

Save the file and repeat steps 1 and 2.

Configure a Relational Database

Perform the following procedure to configure a relational database.

Follow these steps:

1. Create a database instance named NeteAuto.

2. Create a user named neteautoadmin with the password test. Grant neteautoadmin rights (such as public and db_owner rights) to NeteAuto by editing the properties of the user.

   Note: To create a NeteAuto database, neteautoadmin role must have at least minimum (select, insert, update, and delete) permissions for all the tables that are created by.sql script. Also, neteautoadmin must be able to execute all of the stored procedures, if any, defined in these scripts.

3. When you edit user properties, make NeteAuto the default database for neteautoadmin.

4. Run the following scripts in the order in which they are listed:

   - *db_type*-rdbuserdirectory.sql—Configures the tables for the NeteAuto sample, and creates the user entries.
   - ims_*db_type*_rdb.sql—Configures support for organizations

     db_type

       Defines the Microsoft SQL or Oracle depending on the type of database that you are configuring.

   These script files are located in the *admin_tools*\samples\NeteAutoRDB\Organization folder. In this example, *admin_tools* refers to the Administrative Tools, which are installed in the following default locations:

   - **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
   - **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

5. Define a JDBC data source named neteautoDS that points to the NeteAuto database.

   The procedure for configuring a data source depends on the type of application server where CA Identity Manager is installed. The section How to Create a JDBC Data Source (see page 101) includes application server-specific instructions for creating a JDBC data source.

Create the CA Identity Manager Directory

Perform the following procedure to create a CA Identity Manager directory.

Follow these steps:

1. Open the Management Console by entering the following URL in a browser:

   http://*im_server:port*/iam/immanage

   im_server

   Defines the fully qualified domain name of the server where CA Identity Manager is installed.

   port

   Defines the application server port number.

2. Click Directories.

3. Click Create from Wizard to start the CA Identity Manager directory wizard.

4. Browse for the appropriate directory configuration .xml file, and click Next.

   The directory configuration file is located in the following folders:

   - For Sun Java System Directory Server user directories:

     admin_tools\samples\NeteAuto\Organization\directory.xml

   - For relational databases:

     admin_tools\samples\NeteAutoRDB\Organization*db_type* directory.xml

     admin_tools

     Defines the installed location of the Administrative Tools.

     The Administrative Tools are installed in the following default locations:

     Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools

     UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

     db_type

     Specifies the type of database that you are configuring: Microsoft SQL or Oracle.

   Status information is displayed in the Directory Configuration Output screen.

5. On the second page of the wizard, supply the following values:

   - Sun Java System Directory Server

     Name

     NeteAuto Directory

     Description

     Sample NeteAuto directory

     Connection Object Name

     NeteAuto Users

     Host

     Computer name or IP address of the system where the user store is installed.

     Port

     Port number for the user store

     Search root

     dc=security, dc=com

     Username

     User name for an account that can access the user store.

     Password and Confirm Password

     Password for the user account

   - Microsoft SQL Server and Oracle Databases

     Name

     NeteAutoRDB Directory

     Description

     Sample NeteAuto directory

     Connection Object Name

     NeteAutoRDB

     JDBC Data Source

     neteautoDS

     Username

     Neteautoadmin

     Password

     Test

6. Click Next.
7. Click Finish to exit the wizard.

Create the NeteAuto CA Identity Manager Environment

Perform the following procedure to create the NeteAuto CA Identity Manager Environment.

Follow these steps:

1. In the Management Console, click Environments.
2. In the CA Identity Manager Environments screen, click New.

   The CA Identity Manager Environment wizard appears.
3. In the first page of the wizard, enter the following values:

   Environment name

   NeteAuto Environment

   Description

   Sample Environment

   Alias

   Neteauto

   The alias is added to the URL for accessing the CA Identity Manager Environment. For example, the URL for accessing the neteauto environment is:

   http://*server_name*/iam/im/neteauto

   server_name

   Defines the fully qualified domain name of the server where CA Identity Manager is installed, for example:

   http://myserver.mycompany.org/iam/im/neteauto

   Note: The alias is case-sensitive.

   Click Next.
4. Select the CA Identity Manager Directory to associate with the Environment that you are creating:
   - For Sun Java System Directory Server, use the NeteAuto Directory.
   - For Microsoft SQL Server or Oracle database, use the NeteAutoRDB Directory.

   Click Next.

5. Configure support for public tasks, such as the self-registration and forgotten password tasks, as follows:

   a. Type the following alias for public tasks:

      Neteautopublic

   b. Enter SelfRegUser as the anonymous user account.

   c. Click Validate to view the user unique identifier.

   Note: Users do not need to log in to use public tasks.

6. Select the tasks and roles to create for the NeteAuto Environment:

   a. Select Import roles from the file.

   b. Browse to one of the following locations:

      - For a Sun Java System Directory Server user store:

        admin_tools\samples\NeteAuto\RoleDefinitions.xml

      - For a Microsoft SQL Server user store:

        admin_tools\samples\NeteAutoRDB\Organization\mssqlRoleDefinitions.xml

      - For an Oracle user store:

        admin_tools\samples\NeteAutoRDB\Organization\oracleRoleDefinitions.xml

      admin_tools refers to the Administrative Tools, which are installed in the following location by default:

      Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools

      UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

7. Specify a user to serve as the System Manager for this environment and click Next:

   a. Type SuperAdmin in the System Manager field.

   b. Click Add.

      CA Identity Manager adds the unique identifier of the Superadmin user to the list of users.

   c. Click Next.

8. Review the settings for the environment, and do the following tasks:
   - (Optional) Click Previous to modify.
   - Click Finish to create the CA Identity Manager Environment with the current settings.

     The Environment Configuration Output screen shows the progress of the environment creation.
9. Click Continue to exit the CA Identity Manager Environment wizard.
10. Start the CA Identity Manager Environment.

Once you create the NeteAuto Environment, you can:

- Create a skin for this CA Identity Manager environment (see page 37).
- Access the environment (see page 36)

How to Configure the NeteAuto Sample without Organization Support

Configuring the NeteAuto sample without organization support involves the following steps:

- Installing the prerequisite software (see page 21)
- Installing the sample CA Identity Manager environment
- Configuring the database
- Creating a JDBC data source
- Creating the CA Identity Manager directory
- Creating the NeteAuto CA Identity Manager environment

Sample CA Identity Manager Environment Description

For Microsoft SQL Server and Oracle databases, CA Identity Manager includes a version of the NeteAuto environment that does not include organizations. This CA Identity Manager environment includes the following three users:

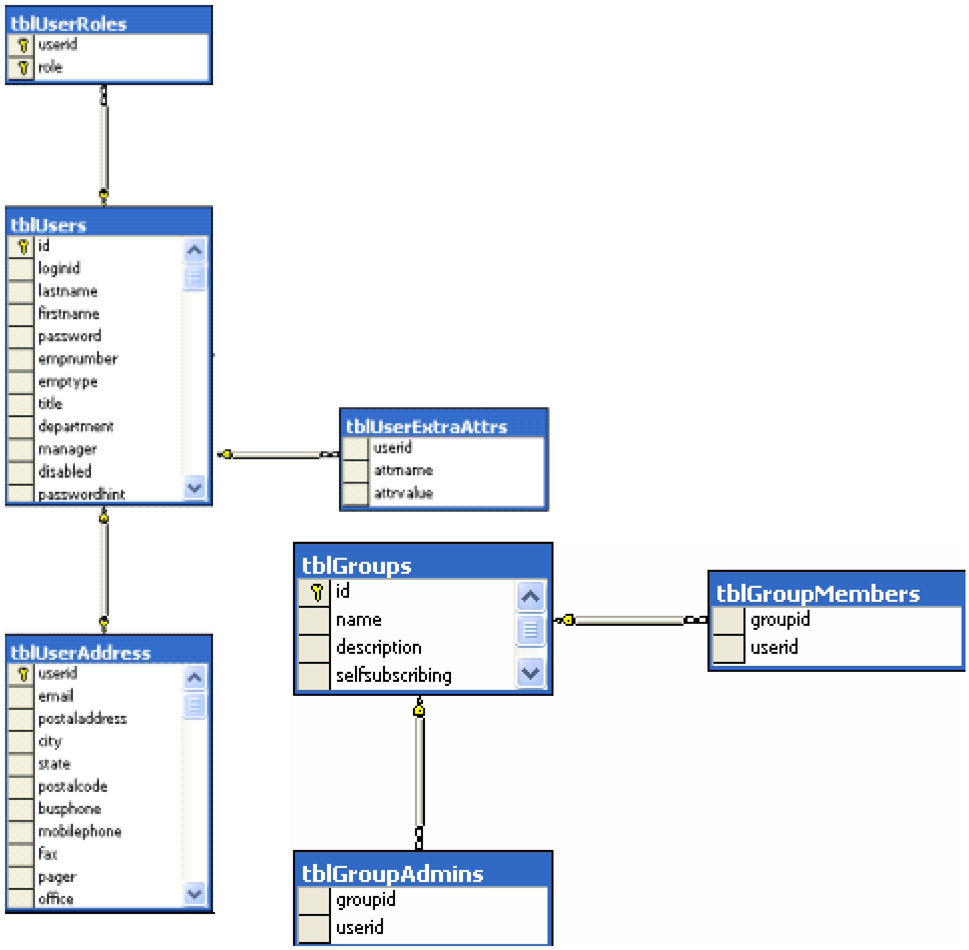
- Superadmin is the administrator account with the System Manager role for this CA Identity Manager environment. As Superadmin, you can perform all of the default admin tasks.

  Note: For a description of the default admin tasks, see the *Administration Guide*.
- SelfRegUser is the administrator account that CA Identity Manager uses to enable self-registration for this CA Identity Manager environment.

- NeteAuto Administrator does not have any privileges when you install the NeteAuto environment.

  However, you can assign the Group Manager role to the NeteAuto Administrator account.

The following illustration describes the NeteAuto sample for a relational database, without organizations:



Installation Files for the Neteauto Environment

CA Identity Manager includes a set of files that you can use to set up a sample CA Identity Manager environment. A CA Identity Manager environment is a view of a management namespace that enables CA Identity Manager administrators to manage objects. These objects such as users and groups are with a set of associated roles and tasks. A CA Identity Manager environment controls the management and graphical presentation of a user store.

The sample CA Identity Manager environment includes:

- Sample users
- Role, task, and screen definitions

  Tasks appear in the User Console when you click a category, such as users or groups. The tasks that appear are based on the roles which are assigned to the user.

  Note: For more information about roles and tasks, see the *Administration Guide*.
- A sample skin that customizes the User Console for NeteAuto users.
- A directory configuration file that you use to create a CA Identity Manager directory.

The files for creating the sample CA Identity Manager environment are installed in the following location:

`admin_tools\samples\NeteAutoRDB\NoOrganization`

In this path, *admin_tools* refers to the Administrative Tools.

The Administrative Tools are placed in the following default locations:

- **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

How To Install the NeteAuto Environment—No Organization Support

Perform the following process to install the NeteAuto environment.

Follow these steps:

1. Verify that the prerequisite software (see page 32) is installed.
2. Configure the database (see page 23).
3. Create the CA Identity Manager Directory. (see page 33)
4. Create the NeteAuto CA Identity Manager Environment (see page 34).
5. Configure the look and feel of the CA Identity Manager user interface (see page 37) for NeteAuto users.

Prerequisite Software

The NeteAuto CA Identity Manager environment has the following prerequisites:

- Install CA Identity Manager as described in the *Installation Guide*. Verify to install the CA Identity Manager Administrative Tools.
- You must have access to a Microsoft SQL Server or Oracle database.

Configure a Relational Database

Perform the following procedure to configure a relational database.

Follow these steps:

1. Create a database instance named NeteAuto.
2. Create a user named neteautoadmin with the password test. Grant neteautoadmin rights (such as public and db_owner rights) to NeteAuto by editing the properties of the user.

   Note: To create a NeteAuto database, neteautoadmin role must have at least minimum (select, insert, update, and delete) permissions for all the tables that are created by.sql script. Also, neteautoadmin must be able to execute all of the stored procedures, if any, defined in these scripts.
3. When you edit user properties, make NeteAuto the default database for neteautoadmin.
4. Run the following scripts in the order in which they are listed:
   - *db_type*-rdbuserdirectory.sql—Configures the tables for the NeteAuto sample, and creates the user entries.
   - ims_*db_type*_rdb.sql—Configures support for organizations

     db_type

         Defines the Microsoft SQL or Oracle depending on the type of database that you are configuring.

   These script files are located in the *admin_tools*\samples\NeteAutoRDB\Organization folder. In this example, *admin_tools* refers to the Administrative Tools, which are installed in the following default locations:
   - **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
   - **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

5. Define a JDBC data source named neteautoDS that points to the NeteAuto database.

   The procedure for configuring a data source depends on the type of application server where CA Identity Manager is installed. The section How to Create a JDBC Data Source (see page 101) includes application server-specific instructions for creating a JDBC data source.

Create the CA Identity Manager Directory

Perform the following procedure to create the CA Identity Manager directory.

Follow these steps:

1. Open the Management Console by entering the following URL in a browser:

   http://*im_server*:*port*/iam/immanage

   im_server

   Defines the fully qualified domain name of the server where CA Identity Manager is installed.

   port

   Defines the application server port number.

2. Click Directories.

   The CA Identity Manager directories screen appears.

3. Click New to start the CA Identity Manager directory wizard.

4. Browse for one of the following directory configuration XML files, and click Next:

   - Sun Java Systems:

     admin_tools\samples\NeteAuto\NoOrganization\directory.xml

   - SQL Server databases:

     admin_tools\samples\NeteAuto\NoOrganization\mssql-directory.xml

   - Oracle databases:

     admin_tools\samples\NeteAuto\NoOrganization\oracle-directory.xml

   admin_tools refers to the Administrative Tools, which are installed by default in the following location:

   - **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
   - **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

   Status information is displayed in the Directory Configuration Output screen.

5. In the second page of the wizard, supply the following values:

   Name

   NeteAutoRDB Directory

   Description

   Sample NeteAuto directory with no organization support

   Connection Object Name

   NeteAutoRDB

   JDBC Data Source

   neteautoDS

   Username

   neteautoadmin

   Password

   test

6. Click Next.
7. Click Finish to exit the wizard.

Create the NeteAuto CA Identity Manager Environment

Perform the following procedure to create the NeteAuto CA Identity Manager environment.

Follow these steps:

1. In the Management Console, click Environments.
2. In the CA Identity Manager environments screen, click New.

   The CA Identity Manager environment wizard opens.
3. In the first page of the wizard, type the following values:
   - Environment name—NeteAuto environment
   - Description—NeteAuto is a sample environment.

- Alias—neteautoRDB

  The alias is added to the URL for accessing the CA Identity Manager environment. For example, the URL for accessing the neteauto environment is:

  `http://`*`domain`*`/iam/im/neteautoRDB`

  In this path, *domain* defines the fully qualified domain name of the server where CA Identity Manager is installed, as in the following example:

  `http://myserver.mycompany.org/iam/im/neteautoRDB`

  Note: The alias is case-sensitive.

Click Next.

4. Select the NeteAutoRDB Directory CA Identity Manager directory to associate with the environment you are creating and click Next.

5. Configure support for public tasks, such as the self-registration and forgotten password tasks.

   Note: Users do not need to log in to access public tasks.

   a. Type the following alias for public tasks:

      neteautoRDBpublic

   b. Type SelfRegUser as the anonymous user account.

   c. Click Validate to view the user unique identifier (2, in this case).

6. Select the tasks and roles to create for the NeteAuto environment:

   - Select Import roles from the file.
   - Browse to the following location:

     im_admin_tools_dir\samples\NeteAutoRDB\NoOrganizations\RoleDefinitions.x ml

     In this path, *im_admin_tools_dir* defines the installed location of the CA Identity Manager administrative tools.

7. Specify a user to serve as the System Manager for this environment, and click Next:

   a. Type SuperAdmin in the System Manager field.

   b. Click Add.

   c. Click Next.

8. Review the settings for the environment.

   - Click Previous to modify.
   - Click Finish to create the CA Identity Manager environment with the current settings.

     The Environment Configuration Output screen shows the progress of the environment creation.

9. Click Finish to exit the CA Identity Manager environment wizard.
10. Start the CA Identity Manager environment.

Once you created the NeteAuto environment, you can:

- Create a skin for this CA Identity Manager environment as described in Setup the NeteAuto Skin (see page 37).
- Access the environment as described in Using the NeteAuto CA Identity Manager environment

How to Use the NeteAuto CA Identity Manager Environment

You can use the NeteAuto CA Identity Manager Environment to manage self-service tasks and users.

Self-Service Task Management

The self-service tasks include:

- Registering as a new user
- Logging in as a self-registered user
- Using the forgotten password feature

Register as a New User

Perform the following procedure to register as a new user.

Follow these steps:

1. Type the following URL in a browser:

   `http://`*`hostname`*`/iam/im/neteautopublic/index.jsp?task.tag=SelfRegistration`

   hostname

   Defines the fully qualified domain name of the system where CA Identity Manager is running.

   This URL directs you to the default ca console.

   At the Self-Registration: End-User License Agreement page, CA Identity Manager displays the CA website.

   Note: You can configure the default Self-Registration task to display custom End-User License Agreement. For instructions, see the *Administration Guide*.

2. Click Accept to proceed.
3. In the Profile tab, provide the following details:
   a. Type values for the required fields, indicated with an asterisk (*).
   b. Type password hints and answers.

      For a forgotten password case, CA Identity Manager provides the password hint and requests the answer. If the answer is correct, CA Identity Manager prompts the user to specify and confirm a new password.
4. Leave the Groups tab unchanged.
5. Click Submit.

Log In as a Self-Registered User

Perform the following procedure to log in as a self-registered user.

Follow these steps:

1. Type the following URL for the NeteAuto CA Identity Manager Environment in a browser:

   `http://`*`hostname`*`/iam/im/neteauto/imcss/index.jsp`

   hostname

   Defines the fully qualified domain name of the system where CA Identity Manager is running.
2. Log in using the username and password you specified when you registered.

Set Up the NeteAuto Skin

To set up the NeteAuto skin, you create a SiteMinder response in the SiteMinder Policy Server.

Follow these steps:

1. Log in to one of the following interfaces as an administrator with Domain privileges:
   - For CA SiteMinder Web Access Manager r12 or higher, log in to the Administrative UI.
   - For CA eTrust SiteMinder 6.0 SP5, log in to the Policy Server User Interface.

   Note: For information on using these interfaces, see the documentation for the version of SiteMinder that you are using.
2. Open the neteautoDomain.

3. Under neteautoDomain, select Realms.

   The following realms appear:

   neteauto_ims_realm

   Protects the CA Identity Manager environment.

   neteauto_pub_realm

   Enables support for public tasks, such as self-registration and forgotten password tasks.

4. Create a rule in each of the realms. Specify the following details:

   - Resource: *
   - Actions: GET, POST

   To simplify the administration, include the NeteAuto skin in the rule name.

5. Create a response for the domain with the following response attributes:

   - Attribute: WebAgent-HTTP-Header-Variable

     This attribute adds new HTTP header to the response.

   - Attribute Kind: Static
   - Variable Name: skin

     Variable Value: neteauto

6. Modify the policy that CA Identity Manager created in the neteautoDomain. Specify the following details:

   - Users
     - For LDAP: Select ou=People, ou=Employees, ou=NeteAuto in Available Members and add it to Current Members. Click OK.
     - For relational databases: Search for users where the id attribute equals *. Select all the users in Available Members and add them to Current Members. Click OK.

- Rules:
  - Add the rules that you created in Step 4.
  - For each rule, click Set Response. Associate each rule with the response that you created in Step 5.

Note: The neteauto skin is based on the imcss console. To view the skin, append /imcss/index.jsp to the URL for the NeteAuto CA Identity Manager Environment as follows:

http://*hostname*/iam/im/neteauto/imcss/index.jsp

Access the NeteAuto CA Identity Manager Environment (see page 40) provides complete instructions for accessing the Neteauto environment.

Use the Forgotten Password Feature

Perform the following procedure to use the forgotten password feature.

Follow these steps:

1. Type the following URL in a browser:

   http://*hostname*/iam/im/neteautopublic/index.jsp?task.tag=ForgottenPasswordReset

   hostname

   Defines the fully qualified domain name of the system where CA Identity Manager is running.

2. Type the unique identifier for the self-registered user that you created in Register as a New User (see page 36), and click Next.

3. Each time when you are prompted, answer the verification question. The answer is the one which you have provided during registration.

   Note: A correct response is required for each question. Canceling the task or closing the browser counts as a failed attempt.

4. Click Submit.

   CA Identity Manager prompts you to supply a new password.

User Management

User management includes the following operations:

- Accessing the NeteAuto CA Identity Manager environment
- Modifying a user
- Assigning the Group Manager role
- Creating a group
- Managing self-registered users

Access the NeteAuto CA Identity Manager Environment

Perform the following procedure to access the NeteAuto CA Identity Manager environment.

Follow these steps:

1. Type the following URL in a browser:

   `http://`*`hostname`*`/iam/im/neteauto/imcss/index.jsp`

   hostname

   Defines the fully qualified domain name, as in the following example:

   http://myserver.mycompany.com/iam/im/neteauto/imcss/index.jsp.

   Note: If you did not configure the Neteauto skin, you can use the following URL to access the Neteauto environment:

   http://*hostname*/iam/im/neteauto

2. In the login screen, type the following credentials:

   User Name

   SuperAdmin

   Password

   test

Modify a User

Perform the following procedure to modify a user.

Follow these steps:

1. Log in the NeteAuto environment as SuperAdmin using the password test.
2. Select Users, Manage Users, Modify User.

   The Select User screen appears.

3. Click Search.

   CA Identity Manager displays a list of users in the NeteAuto environment.

4. Select the NeteAuto administrator, as follows:

   - For LDAP directories, NeteAuto Administrator
   - For relational databases, NeteAuto Admin

   Click Select. CA Identity Manager displays the profile for the NeteAuto administrator.

5. In the Title field, type Manager. Click Submit.

   CA Identity Manager confirms the task submission.

6. Click OK to return to the main screen.

Assign the Group Manager Role

Assigning a group manager role is necessary. Perform the following procedure to assign a group manager.

Follow these steps:

1. As SuperAdmin, select the Roles and Tasks tab, then select Admin Roles, Modify Admin Roles.

2. Select the Group Manager role, and click Select.

   The profile for the Group Manager role appears.

3. Click the Members tab, and click Add under Member Policies.

   The Member Policy screen appears.

4. Under Member Rule, click the down arrow in the Users field.

   From the drop-down list, select where <user-filter>.

   The Users field changes to let you enter a filter for the rule.

5. Enter a membership rule as follows:

   a. In the first field, select Title from the drop-down list.

   b. In the second field, make sure the equal sign (=) is selected.

   c. In the third field, type Manager.

6. In the Scope Rules section, define rules for the users, groups, and organizations (when supported) as follows:

   a. In the Users field, click the down arrow to see a list of options. Select (all) from the list.

   b. Repeat Step 'a' in the Group and Organization fields (when supported).

   c. Leave the Access Tasks field blank.

7. Click OK.

   CA Identity Manager displays the member policy that you created.

8. Click Submit.

   CA Identity Manager confirms the task submission.

9. Click OK to return to the main screen.

10. Close CA Identity Manager.

Create a Group

Perform the following procedure to create a group.

Follow these steps:

1. Log in to CA Identity Manager as the NeteAuto administrator, as follows:

   - For LDAP directories, type the user name NeteAuto Administrator and the password test.

   - For relational databases, type the user name NeteAuto Admin and the password test.

   The list of tasks that the NeteAuto administrator can perform appears. Because the NeteAuto administrator can perform only a limited number of tasks, CA Identity Manager lists the tasks instead of categories.

2. Click Create Group.

3. Verify that Create a new group is selected, and click OK.

4. Implement one of the following steps that fits your case:

   - If the NeteAuto environment supports organizations:

     a. In the Org Name field, click the ellipsis symbol (...) to select the organization where CA Identity Manager creates the group.

     b. At the bottom of the Select Organization screen, expand NeteAuto.

     c. Select the Dealer organization.

   - If the NeteAuto environment does not support organizations, go to the next step.

5. Type the following information for the group:

   - Group Name: Dealer Administrators

   - Group Description: Administrators for NeteAuto dealerships.

6. Click the Membership tab and click the Add a user.

   The Select User screen appears.

7. Click Search.
8. Select the NeteAuto administrator, and click Select.
9. Click Submit to create the group.

Manage Self-Registered Users

Perform the following procedure when you want to manage self-registered users.

Follow these steps:

1. Log in to CA Identity Manager as a NeteAuto administrator, using the following credentials:

   - For LDAP directories:

     Username

     NeteAuto Administrator

     Password

     test

   - For relational databases:

     Username

     NeteAuto Admin

     Password

     test

   The list of tasks that the NeteAuto administrator can perform appears on the left side of the User Console. Because the NeteAuto administrator can perform only a limited number of tasks, CA Identity Manager lists the tasks instead of the categories.

2. Click Modify Group.
3. Click Search.

   CA Identity Manager displays a list of groups.

4. Select Dealer Administrators, and click Select.
5. Click the Membership tab, and click Add a user.

   The Select User screen appears.

6. Click Search.
7. In the User Search screen, select the user that you typed in Register as a New User (see page 36). Click Select.

8. Click Submit.

   CA Identity Manager confirms the task submission.

9. Click OK to return to the main screen.

To confirm that the user is a member of the created group, use the View Group task.

How to Configure Additional Features

Once you have installed the NeteAuto sample and practiced basic CA Identity Manager functionality, use the NeteAuto environment to practice and test additional CA Identity Manager features, including email notifications and workflow.

Note: For more information about these features, see the *Administration Guide*.

SiteMinder Login Name Restriction for Global User Name

If a user is required to log in to the SiteMinder Policy Server, the following characters or character strings cannot be part of a global user name:

&
*
:
()

Workaround

Avoid using these characters in the global user name.

Chapter 3: LDAP User Store Management

This section contains the following topics:

CA Identity Manager Directories

A *CA Identity Manager directory* describes how objects such as users, groups, and organizations are stored in the user directory and how it is represented in CA Identity Manager. A CA Identity Manager directory is associated with one or more CA Identity Manager environments.

How to Create a CA Identity Manager Directory

Creating a CA Identity Manager directory for an LDAP user store involves the following steps:

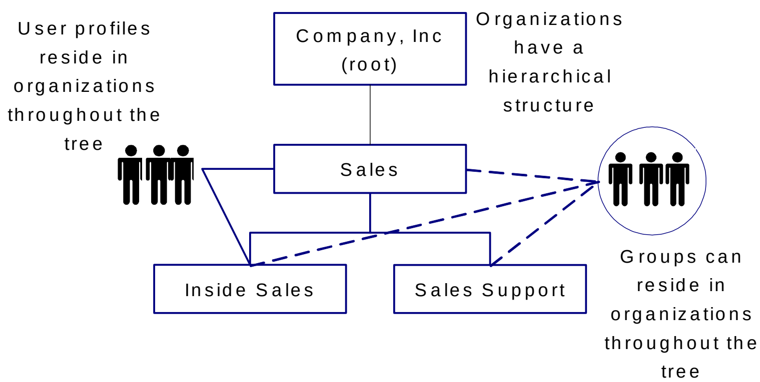
1. Determining the directory structure.
2. Describing the objects in the user store by modifying a directory configuration file (directory.xml) (see page 50).
3. Importing the directory configuration file and creating the directory (see page 145).

Note: When using SiteMinder, verify that you have applied the policy store schema prior creating a CA Identity Manager Directory. For more information on specific policy store schemas and how to apply them, see the *Installation Guide*.

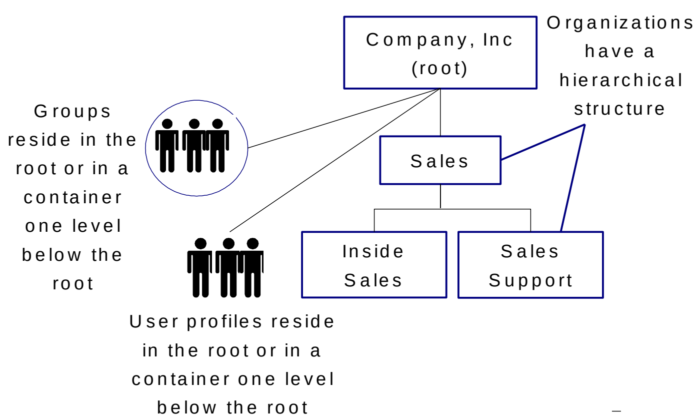
Directory Structure

CA Identity Manager supports the following directory structures:

- Hierarchical—Contains a parent organization (root) and suborganizations. The suborganizations can also have suborganizations, which creates a multilevel structure, as shown in the following illustration:

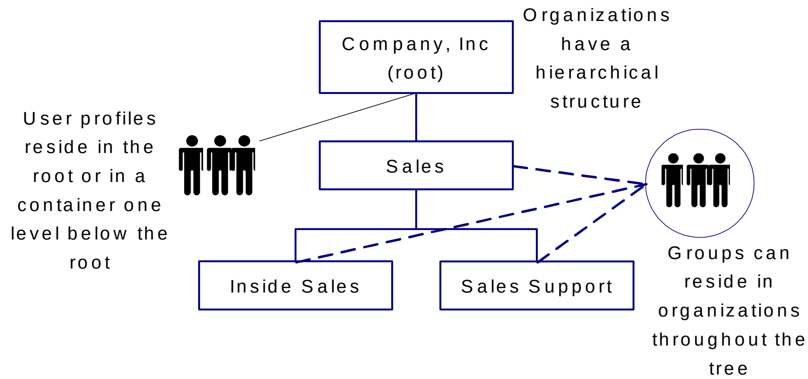


- Flat—User and groups are stored at the search root or in a container one level below the search root. Organizations have a hierarchical structure, as shown in the following illustration of a flat directory structure:



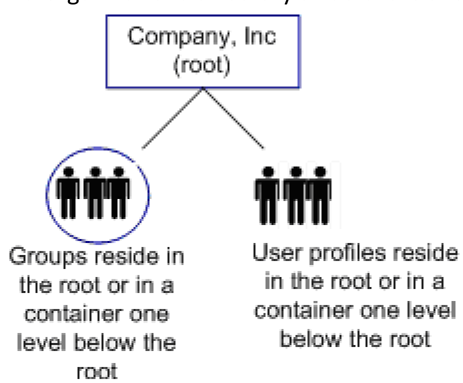
To facilitate user management and delegation in flat directory structures, users and groups belong to logical organizations. The logical organization is stored as an attribute in user and group profiles.

- Flat User—Organizations and groups are stored hierarchically, but users are stored at the search root or in a container one level below the search root. Illustration of a flat user directory structure is shown in the following diagram:



In flat user directory structures, users belong to logical organizations. The logical organization of a user is stored as an attribute in a user profile.

- No organizations—The directory does not include organizations. Users and groups are stored at the search root or in a container one level below the search root. A no-organizations directory structure is shown in the following illustration:



Note: A directory can contain more than one type of structure. For example, user profiles can be stored in a flat structure in one part of the directory and hierarchically in another. To support a hybrid directory structure, create multiple CA Identity Manager environments.

Directory Configuration File

To describe structure of a user directory to CA Identity Manager, create a directory configuration file.

The directory configuration file contains one or more of the following sections:

CA Identity Manager Directory Information

Contains information about the CA Identity Manager directory.

Note: Do not modify information in this section. CA Identity Manager prompts you to provide this information when you create a CA Identity Manager directory in the Management Console.

Attribute Validation

Defines the validation rules that apply to the CA Identity Manager directory.

Provider Information

Describes the user store that CA Identity Manager manages.

Directory Search Information

Enables you to specify how CA Identity Manager searches the user store.

User Object

Describes how users are stored in the user store and how it is represented in CA Identity Manager.

Group Object

Describes how groups are stored in the user store and how it is represented in CA Identity Manager.

Organization Object

Describes how organizations are stored and how it is represented in CA Identity Manager. Organization object provides details only when the user store includes organizations.

Self-Subscribing Object

Configures support for groups that self-service users can join.

Directory Groups Behavior

Specifies whether the CA Identity Manager directory supports dynamic and nested groups.

To create a directory configuration file, you modify a configuration template.

How to Select a Directory Configuration Template

CA Identity Manager supplies directory configuration templates that support different directory types and structures. To create a CA Identity Manager directory, modify the template that most closely matches your directory structure.

The templates described in the following table are installed with the Administrative Tools:

admin_tools\directoryTemplates*directory_type*\

The Administrative Tools are placed in the following default locations:

- **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

The types of directories and the corresponding configuration templates are shown in the following table:

Directory Type	Template
Active Directory (ADSI) LDAP directory with a hierarchical structure	ActiveDirectory\directory.xml
Microsoft ADAM Directory with a hierarchical structure	ADAM\directory.xml
IBM Directory Server directory with a hierarchical structure	IBMDirectoryServer\directory.xml
Novell eDirectory user directory with a hierarchical structure	eDirectory\directory.xml
Oracle Internet Directory with a hierarchical structure	OracleInternetDirectory\directory.xml
Sun Java System (SunOne or iPlanet) LDAP directory with a hierarchical structure	IPlanetHierarchical\directory.xml
Sun Java System (SunOne or iPlanet) LDAP directory with a flat structure	IPlanetFlat\directory.xml
Sun Java System (SunOne or iPlanet) LDAP directory that does not include organizations.	IPlanetNoOrganizations\directory.xml
CA Directory user store with a hierarchical structure	eTrustDirectory\directory.xml

Directory Type	Template
Provisioning Directory This template configures the Provisioning Directory for a CA Identity Manager environment. **Note:** You can use this configuration template as installed. You do not need to modify this template.	ProvisioningServer\directory.xml
Custom directory	Use the template that most closely resembles your directory.

Copy the configuration template to a new directory or save it with a different name to prevent overwriting it.

How to Describe a User Directory to CA Identity Manager

To manage a directory, CA Identity Manager must understand the structure and content of a directory. To describe the directory to CA Identity Manager, modify the directory configuration file (directory.xml) in the appropriate template directory.

The directory configuration file has the following important conventions:

- ##—Indicates required values.

  To provide all the required information, locate all double pound signs (##) and replace them with appropriate values. For example, ##DISABLED_STATE indicates that you must supply an attribute to store the account status of a user.

- @—Indicates values that CA Identity Manager populates. Do not modify these values in the directory configuration file. CA Identity Manager prompts you to supply the values when you import the directory configuration file.

Before you modify the directory configuration file, you need the following information:

- LDAP object classes for the user, group, and organization objects
- List of attributes in user, group, and organization profiles

How to Modify the Directory Configuration File

Perform the following steps to modify the directory configuration file.

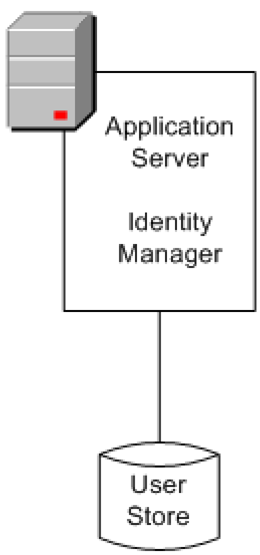
Note: Steps that are required are noted accordingly.

1. Limit the size of search results (see page 55).
2. Modify the default user, organization, or group managed objects.
3. Change the default attribute descriptions.
4. Modify well-known attributes (see page 74). (required)

   Well-known attributes identify special attributes, such as the password attribute, in CA Identity Manager.
5. Configure CA Identity Manager for your directory structure (see page 81) (required).
6. Enable users to subscribe to groups (see page 82).

Connection to the User Directory

CA Identity Manager connects to a user directory to store information such as a user, group, and organizational information as shown in the following illustration:



A new directory or database is not required. However, the existing directory or database must be on a system that has a fully qualified domain name (FQDN).

For a list of supported directory and database types, see the CA Identity Manager support matrix on the CA Support Site.

You configure a connection to the user store when you create a CA Identity Manager directory in the Management Console.

If you export the directory configuration after creating a CA Identity Manager directory, the user directory connection information is displayed in the Provider element of the directory configuration file.

Provider Element

Configuration information is stored in the Provider element and its subelements in the directory.xml file.

Note: If you are creating a CA Identity Manager directory, you do not need to provide directory connection information in the directory.xml file. You provide connection information in the CA Identity Manager Directory wizard in the Management Console. Modify the Provider element for updates only.

The Provider element includes the following subelements:

LDAP

Describes the user directory to which you are connecting.

Credentials

Provides the user name and password for accessing the LDAP user store.

Connection

Supplies the host name and port for the computer where the user store is located.

Provisioning Domain

Defines the Provisioning Domain that CA Identity Manager manages (for provisioning users only).

A completed Provider element resembles the following code:

```
<Provider type="LDAP" userdirectory="@SMDirName">
   <LDAP searchroot="@SMDirSearchRoot" secure="@SMDirSecure" />
   <Credentials user="@SMDirUser"
     cleartext="true">@SMDirPassword</Credentials>
   <Connection host="@SMDirHost" port="@SMDirPort" />
   <eTrustAdmin domain="@SMDirETrustAdminDomain" />
</Provider>
```

The Provider element includes the following parameters:

type

Specifies the type of the database. For all LDAP user stores, specify LDAP (default).

userdirectory

Specifies the name of the user directory connection.

Note: Do not specify a name for the user directory connection in the directory.xml file. CA Identity Manager prompts you to supply the name when you create the CA Identity Manager directory in the Management Console.

Note: The parameters are optional.

LDAP Subelement

The LDAP subelement includes the following parameters:

searchroot

Specifies the location in an LDAP directory that serves as the starting point for the directory—typically, an organization (o) or organizational unit (ou).

secure

Forces a Secure Sockets Layer (SSL) connection to the LDAP user directory, as follows:

- True—CA Identity Manager uses a secure connection.
- False—CA Identity Manager connects to the user directory without SSL (default).

Note: The parameters are optional.

Credentials Subelement

To connect to an LDAP directory, CA Identity Manager must provide valid credentials. The credentials are defined in the Credentials subelement, which resembles the following code:

```
<Credentials user="@SMDirUser" cleartext="true">
  "MyPassword"
</Credentials>
```

If you do not specify a password in the Credentials subelement prompts for the password, when you create the CA Identity Manager directory in the Management Console.

Note: We recommend specifying the password in the Management Console.

If you specify the password in the Management Console, CA Identity Manager encrypts the password for you. Otherwise, If you do not want the password to appear in clear text, encrypt the password using the password tool that is installed with CA Identity Manager.

Note: You can specify only one set of credentials. If you define multiple directories, as described in Connection Subelement, the credentials that you specify must apply to all the directories.

The Credentials subelement includes the following parameters:

user

Specifies the login ID for an account that can access the directory.

For provisioning users, the user account that you specify must have the Domain Administrator profile or an equivalent set of privileges in the Provisioning Server.

Note: Do not specify a value for the user parameter in the directory.xml file. CA Identity Manager prompts you to supply the login ID when you create the CA Identity Manager Directory in the Management Console.

cleartext

Determines whether the password is displayed in clear text in the directory.xml file, as follows:

- True—The password is displayed in clear text.
- False—The password is encrypted (default).

Note: The parameters are optional.

Connection Subelement

The Connection subelement describes the location of the user store that CA Identity Manager manages. This subelement includes the following parameters:

host

Specifies the host name or IP address of the system where the user directory is located.

Note: If the connecting system has an IPv6 address, enclose the IP address within the brackets ([ ]) as follows:

<Connection host="[2::9255:214:22ff:fe72:525a]" port="20389"

failover="[2::9255:214:22ff:fe72:525a]:20389"/>

port

Specifies the port number for the user directory.

failover

Specifies the host name and IP address of the system where redundant user stores exist, in case the primary system is unavailable. When the primary system becomes available again, the failover system continues to be used. To return to using the primary system, restart the secondary system. If multiple servers are listed, CA Identity Manager attempts to connect to the systems in the listed order.

Specify the host name and IP address in the failover attribute in a *space-separated* list, as follows:

failover="IPaddress:port IPaddress:port"

For example:

<Connection host="123.456.789.001" port="20389"

failover="123.456.789.002:20389 123.456.789.003:20389"/>

Note: Port 20389 is the default port for the Provisioning Server.

Note: The parameters are optional.

Provisioning Subelement

If CA Identity Manager environment includes provisioning, define the Provisioning Domain as follows:

```
<eTrustadmin domain="@SMDirProvisioningDomain" />
```

The Provisioning subelement includes the following parameter:

domain

Contains the name of the Provisioning Domain that CA Identity Manager manages.

When you create the CA Identity Manager directory in the Management Console, you are prompted for the domain name. So, verify that you specify a value for the domain parameter in the directory configuration file (directory.xml).

Directory Search Parameters

You can set the following search parameters in the DirectorySearch element:

maxrows

Specifies the maximum number of objects that CA Identity Manager can return when searching a user directory. When the number of objects exceeds the limit, an error is displayed.

By setting a value for the maxrows parameter, you can override the settings in the LDAP directory that limit search results. When conflicting settings apply, the LDAP server uses the lowest setting.

Note: The maxrows parameter does not limit the number of objects that are displayed on a CA Identity Manager task screen. To configure display settings, modify the list screen definition in the CA Identity Manager User Console. For instructions, see the *User Console Design Guide*.

maxpagesize

Specifies the number of objects that can be returned in a single search. If the number of objects exceeds the page size, CA Identity Manager performs multiple searches.

Note the following points when specifying maxpagesize:

- To use the maxpagesize option, the user store that CA Identity Manager manages must support paging. Some user store types require additional configuration to support paging. For more information, see How to Improve Performance for Large Searches (see page 91).
- If the user store does not support paging and also a value for maxrows is specified, CA Identity Manager uses only the maxrows value to control search size.

timeout

Determines the maximum number of seconds that CA Identity Manager searches a directory before terminating the search.

Note: The DirectorySearch element is optional. However, the directory supports paging (see page 91), we recommend specifying the DirectorySearch element.

More Information:

How to Improve Directory Search Performance (see page 90)
How to Improve Performance for Large Searches (see page 91)

User, Group, and Organization Managed Object Descriptions

In a CA Identity Manager, you manage the following types of objects that correspond to entries in a user directory:

Users

Represent users in an enterprise. A user belongs to a single organization.

Groups

Represent associations of users who have something in common.

Organizations

Represent business units. Organizations contain details such as users, groups, and other organizations.

An object description contains the following information:

- Information about the object (see page 110), such as the LDAP object class and the container in which objects are stored.
- The attributes that store information about an entry (see page 114). For example, the pager attribute stores a pager number.

Note: A CA Identity Manager environment supports only one type of user, group, and organization object. For example, all user objects have the same object class.

Managed Object Descriptions

A managed object is described by specifying object information in the User Object, Group Object, and Organization Object sections of the directory configuration file.

Note: When using the configuration template (directory.xml file), Organization Object section is unavailable for those user directories that do not support organizations.

Each of these sections contains ImsManagedObject elements, such as the following example:

```
<ImsManagedObject name="User" description="My Users"
objectclass="top,person,organizationalperson,inetorgperson" objecttype="USER">
```

Optionally, the ImsManagedObject element can include a Container element, such as the following example:

```
<Container objectclass="top,organizationalUnit" attribute="ou" value="people" />
```

Specify Object Information

Object information is specified by supplying values for various parameters.

Follow these steps:

1. Locate the ImsManagedObject element in the User Object, Organization Object, or Group Object section.

2. Supply values for the following parameters:

   name

   Specifies a unique name for the managed object.

   Note: This parameter is required.

   description

   Contains a description of the managed object.

   objectclass

   Specifies the name of the LDAP object class for the object type (user, group, or organization). The object class determines the list of available attributes for an object.

   If attributes from multiple object classes apply to an object type, list the object classes in a comma-delimited list. For example, if an object contains attributes from the person, organizationalperson and inetorgperson object classes, add these object classes as follows:

   objectclass="top,person,organizationalperson,inetorgperson"

   Each LDAP directory includes a set of predefined object classes. See directory server documentation for information about predefined object classes.

   Note: This parameter is required.

   objecttype

   Specifies the type of the managed object. The valid values are as follows:

   - User
   - Organization
   - Group

   Note: This parameter is required.

   maxrows

   Specifies the maximum number of objects that CA Identity Manager can return when searching a user directory. When the number of objects exceeds the limit, an error is displayed.

   By setting a value for the maxrows parameter, you can override the settings in the LDAP directory that limit search results. When conflicting settings apply, the LDAP server uses the lowest setting.

   Note: The maxrows parameter does not limit the number of objects that are displayed on a CA Identity Manager task screen. To configure display settings, modify the list screen definition in the CA Identity Manager User Console. For instructions, see the *User Console Design Guide*.

maxpagesize

Specifies the number of objects that can be returned in a single search. If the number of objects exceeds the page size, CA Identity Manager performs multiple searches.

Note the following points when specifying Search Page Size:

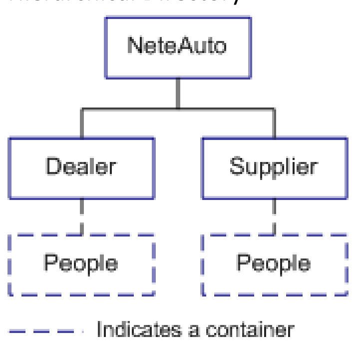
- To use the Search Page Size option, the user store that CA Identity Manager manages must support paging. Some user store types require additional configuration to support paging. For more information, see How to Improve Search Performance (see page 91).
- If the user store does not support paging and also a value for maxrows is specified, CA Identity Manager uses only the maxrows value to control search size.

3. Optionally, supply container information.

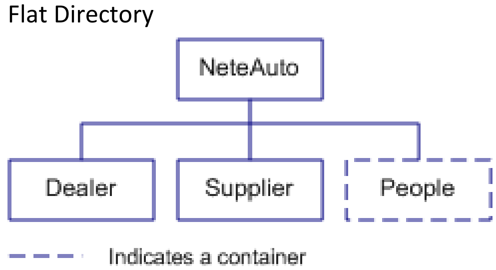
Containers

To simplify administration, you can group objects of a specific type in a container. When you specify a container in the directory configuration file, CA Identity Manager manages only entries in the container. For example, if you specify a user container named People, CA Identity Manager manages users in the People container, as shown in the following illustrations:

- Hierarchical Directory



- Flat Directory



In these examples, all users exist in the People containers.

When you specify a container, note the following points:

- If no container exists in an organization, CA Identity Manager creates the container as soon as the first entry is added. For a hierarchical directory, CA Identity Manager creates the container in the organization where the entry is added. For flat directories and directories not supporting organizations, CA Identity Manager creates the container under the search root, which you specify when you create the CA Identity Manager directory.
- CA Identity Manager ignores entries that are not in the specified container. For example, when you specify the People container, you cannot manage users existing outside of the People container.

  Note: To manage users who are not in the specified container, you can create another CA Identity Manager environment.

Containers and Well-Known Attributes

Well-known attributes are attributes that have special meaning in CA Identity Manager. When CA Identity Manager manages a user store including containers, the following well-known attributes identify information about the container:

%ORG_MEMBERSHIP%

Identifies the attribute that stores the full name (DN) of the container.

For example, the full name resembles as:

ou=People, ou=Employee, ou=NeteAuto, dc=security, dc=com

%ORG_MEMBERSHIP_NAME%

Identifies the attribute that stores the user-friendly name of the attribute.

For example, the user-friendly name of the container in the previous example is People.

These well-known attributes appear in the attribute descriptions in the User Object and Group Object sections of the directory.xml file, as follows:

```
 <ImsManagedObjectAttr physicalname="someattribute" description="Organization"
displayname="Organization" valuetype="String" required="true"
wellknown="%ORG_MEMBERSHIP%" maxlength="0" permission="WRITEONCE"
searchable="false" />
```

For hierarchical user store structures, the physicalname and wellknown parameters are mapped to the well-known attribute as follows:

```
<ImsManagedObjectAttr physicalname="%ORG_MEMBERSHIP%" description="Organization"
displayname="Organization" valuetype="String" required="true"
wellknown="%ORG_MEMBERSHIP%" maxlength="0" permission="WRITEONCE"
searchable="false" />
```

The example indicates that CA Identity Manager automatically derives the container DN and user-friendly name from other information in the directory.xml file.

For flat user store structures, supply the physical attribute names.

Note: See How to Describe a Flat User Directory Structure (see page 81) for instructions.

Specify a User or Group Container

Perform the following procedure to specify a user or group container.

Follow these steps:

1. Locate the Container element in the User Object or Group Object section.
2. Supply values for the following parameters:

   objectclass

   Determines the LDAP object class of the container where objects of a specific type are created. For example, the default value for the user container is "top,organizationalUnit," which indicates that users are created in LDAP organizational units (ou).

   When you are managing dynamic or nested groups, be sure to specify an objectclass supporting these group types (see page 83).

   Note: This parameter is required.

   attribute

   Specifies the attribute that stores the container name, for example, ou.

   The attribute is paired with the value to form the relative DN of the container, as in the following example:

   ou=People

   Note: This parameter is required.

value

Specifies the name of the container.

Note: This parameter is required.

Note: You cannot specify containers for organizations.

Attribute Descriptions

An attribute stores information about an entry, such as a telephone number or address. An entry attribute determines its profile.

In the directory configuration file, attributes are described in ImsManagedObjectAttr elements. In the User Object, Group Object and Organization Object sections of the directory configuration file, you can do the following actions:

- Modify default attribute descriptions to describe the attributes in your user store.
- Create new attribute descriptions by copying an existing description and modifying values as needed.

For each attribute in user, group, and organization profiles, there is one ImsManagedObjectAttr element. For example, an ImsManagedObjectAttr element is described as a user ID.

An ImsManagedObjectAttr element resembles the following code:

```
<ImsManagedObjectAttr physicalname="uid" displayname="User ID" description="User
ID" valuetype="String" required="true" multivalued="false" wellknown="%USER_ID%"
maxlength="0" />
```

The ImsManagedObjectAttr has the following parameters:

physicalname

This parameter must contain one of the following items:

- The name of the LDAP attribute where the profile value is stored. For example, user ID is stored in the uid attribute in the user directory.

  Note: To improve performance, index LDAP attributes that are used in search queries in the User Console.

- A well-known attribute (see page 74). When you supply a well-known attribute, CA Identity Manager computes the value automatically. For example, on specifying a well-known attribute %ORG_MEMBERSHIP%, CA Identity Manager determines the organization to which the entry belongs, based upon the DN of an entry.

description

Contains the description of the attribute

displayname

Specifies a unique name for the attribute.

In the User Console, the display name appears in the list of attributes that are available to add to a task screen. This parameter is required.

Note: Do not modify the displayname of an attribute in the directory configuration file (directory.xml). To change the name of the attribute on a task screen, you can specify a label for the attribute in the task screen definition. For more information, see the A*dministration Guide*.

valuetype

Specifies data type of the attribute. The valid values are as follows:

String

The value can be any string.

This is the default value.

Integer

The value must be an integer.

Note: Integer does not support decimal numbers.

Number

The value must be an integer. The number option supports decimal numbers.

Date

The value must parse to a valid date using the pattern:

MM/dd/yyyy

ISODate

The value must parse to a valid date using the pattern yyyy-MM-dd.

UnicenterDate

The value must parse to a valid date using the pattern YYYYYYYDDD where:

YYYYYYY is a seven number representation for a year beginning with three zeros. For example: 0002008

DDD is the three number representation for the day beginning with zeros, as needed. Valid values include in range from 001 to 366.

Structured

This type of attribute consists of structured data that enables a single attribute value to store multiple related values. For example, a structured attribute contains values such as First Name, Last Name, and Email Address values.

Certain endpoint types use these attributes but are managed through CA Identity Manager.

Note: CA Identity Manager can display structured attributes in a table in the User Console. When users edit values in the table, the values are stored in the user store, propagating back to the endpoint. For more information about displaying multivalued attributes, see the *Administration Guide*.

required

Indicates whether the attribute is required, as follows:

- True—The attribute is required.
- False—The attribute is optional (default).

Note: If an attribute is required for an LDAP directory server, set the required parameter to true.

multivalued

Indicates whether the attribute can have multiple values. For example, the group membership attribute is multivalued to store the user DN of each group member. The valid values are as follows:

- True—The attribute can have multiple values.
- False—The attribute can have only a single value (default).

Important! The Group Membership and Admin Roles attributes in the User object definition must be multivalued.

wellknown

Defines the name of the well-known attribute.

Well-known attributes have a specific meaning in CA Identity Manager (see page 74). They are identified in the syntax:

%ATTRIBUTENAME%

maxlength

Defines the maximum length that a value of an attribute can have. Set the maxlength parameter to 0 to specify an unlimited length.

Note: This parameter is required.

permission

Indicates whether the value of an attribute can be modified in a task screen. The valid values are as follows:

READONLY

The value is displayed but cannot be modified.

WRITEONCE

The value cannot be modified once the object is created. For example, a user ID cannot be changed after the user is created.

READWRITE

The value can be modified (default).

hidden

Indicates whether an attribute appears in CA Identity Manager task forms. The valid values are as follows:

- True—The attribute is not displayed to users.
- False—The attribute is displayed to users (default).

Logical attributes use hidden attributes.

Note: For more information, see the *Programming Guide for Java*.

system

Specifies only CA Identity Manager used attributes. Users in the User Console not to modify the attributes. The valid values are as follows:

- True—Users cannot modify the attribute. The attribute is hidden in the CA Identity Manager user interface.
- False—Users can modify this attribute. The attribute is available to add to task screens in the CA Identity Manager user interface. (default)

validationruleset

Associates a validation rule-set with the attribute.

Verify that the validation rule set that you specify is defined in a ValidationRuleSet element in the directory configuration file.

objectclass

Indicates the LDAP auxiliary class for a user, group, or organization attribute when the attribute is not part of the primary objectclass specified in the ImsManagedObject element.

For example, assume that the primary object class for users is top, person, and organizationalperson, which defines the following user attributes:

- common name (cn)
- surname (sn)
- user id (uid)
- password (userPassword)

To include the attribute employeeID, which is defined in the Employee auxiliary class, you would add the following attribute description:

```
<ImsManagedObjectAttr physicalname="employeeID" displayname="Employee ID"
description="Employee ID" valuetype="String" required="true" multivalued="false"
maxlength="0" objectclass="Employee"/>
```

Specify Attribute Descriptions

Describing attributes involves the following steps:

1. Read the relevant sections among the following topics:
   - CA Directory Considerations (see page 72)
   - Microsoft Active Directory Considerations (see page 73)
   - IBM Directory Server Considerations (see page 73)
   - Oracle Internet Directory Considerations (see page 74)
2. In the User Object, Group Object and Organization Object sections of the directory configuration file, do the following actions:
   - Modify default attribute descriptions to describe your directory attributes.
   - Create new attribute descriptions by copying an existing description and modifying values as needed.

   Note: Assume that a new attribute description is created and a physical attribute is specified. Be sure that the physical attribute must exist in the object class (or classes) that you specified for the object type.
3. (Optional) Change the display settings (see page 69) for the attribute to prevent displaying sensitive information, such as passwords or salaries, in the User Console.
4. (Optional) Configure a default sort order.
5. If you are managing a directory with a Flat or Flat User structure or a directory excluding organizations, go to Describe the User Directory Structure (see page 81).

Managing Sensitive Attributes

CA Identity Manager provides the following methods for managing sensitive attributes:

- Data classifications for attributes

  Data classifications allow you to specify display and encryption properties for attributes in the directory configuration file (directory.xml).

  You can define data classifications that manage sensitive attributes as follows:

  - In CA Identity Manager task screens, display the value of an attribute as a series of asterisks.

    For example, you can display passwords as asterisks instead of displaying them in clear text.

  - In View Submitted Task screens, hide the attribute value.

    This option enables you to hide attributes from administrators. For instance, hiding salary details such as the salary from administrators who view task status in CA Identity Manager but do not need to view salary details.

  - Ignore certain attributes when creating a copy of an existing object.
  - Encrypt an attribute

- Field styles in task profile screens

  If you do not want to modify an attribute in the directory.xml file, set the display property for the attribute in screen definitions where the sensitive attribute appears.

  The field style enables you to display attributes, such as passwords, as a series of asterisks instead of clear text.

  Note: For more information about the field style for sensitive attributes, search for field styles in the User Console help.

Data Classification Attributes

The Data Classification element provides a way to associate additional properties with an attribute description. The values in this element determine how CA Identity Manager handles the attribute. This element supports the following parameters:

- sensitive

  Causes CA Identity Manager to display the attribute as a series of asterisks (*) in View Submitted Tasks screens. This parameter prevents old and new values for the attribute from appearing in clear text in View Submitted tasks screens.

  Additionally, if you create a copy of an existing user in the User Console, this parameter prevents the attribute from being copied to the new user.

- vst_hide

  Hides the attribute in the Event Details screen for the View Submitted Tasks tab. Unlike sensitive attributes, which are displayed as asterisks, vst_hidden attributes are not displayed.

  You can use this parameter to prevent changes to an attribute, such as the salary, from displaying in View Submitted Tasks.

- ignore_on_copy

  Causes CA Identity Manager to ignore an attribute when an administrator creates a copy of an object in the User Console. For example, assume that you have specified ignore_on_copy for the password attribute on a user object. When copying a user profile, CA Identity Manager does not apply the password of the current user to the new user profile.

- AttributeLevelEncrypt

  Encrypts attribute values when they are stored in the user store. If CA Identity Manager is FIPS 140-2 enabled, CA Identity Manager uses RC2 encryption or FIPS 140-2 encryption.

  For more information about FIPS 140-2 support in CA Identity Manager, see the *Configuration Guide*.

  The attributes appear in clear text during runtime.

  Note: To prevent attributes from appearing in clear text in screens, you can also add a sensitive data classification element to encrypted attributes. For more information, see How to Add Attribute-Level Encryption (see page 70).

- PreviouslyEncrypted

  Causes CA Identity Manager to detect and decrypt any encrypted values in the attribute when it accesses the object in the user store.

  You use this data classification to decrypt any previously encrypted values.

  The clear text value is saved to the store when you save the object.

Configure Data Classification Attributes

Follow these steps:

1. Locate the attribute in the directory configuration file.
2. After the attribute description, add the following attribute:

   <DataClassification name="*parameter*">

   parameter

   Represents one of the following parameters:

   sensitive

   vst_hide

   ignore_on_copy

   AttributeLevelEncrypt

   PreviouslyEncrypted

   For example, an attribute description that includes the vst_hide data classification attribute resembles the following code:

```
<ImsManagedObjectAttr physicalname="salary" displayname="Salary"
description="salary" valuetype="String" required="false" multivalued="false"
maxlength="0">
    <DataClassification name="vst_hide"/>
```

Attribute-Level Encryption

You can encrypt an attribute in the user store by specifying an AttributeLevelEncypt data classification for that attribute in the directory configuration file (directory.xml). When attribute-level encryption is enabled, CA Identity Manager encrypts the value of that attribute before storing it in the user store. The attribute is displayed as clear text in the User Console.

Note: To prevent attributes from appearing in clear text in screens, you can also add a sensitive data classification element to encrypted attributes. For more information, see How to Add Attribute-Level Encryption (see page 70).

If FIPS 140-2 support is enabled, the attribute is encrypted using RC2 encryption or FIPS 140-2 encryption.

Before you implement the attribute-level encryption, note the following points:

- CA Identity Manager cannot find encrypted attributes in a search.

  Assume that an encrypted attribute is added to a member, admin, owner policy, or an identity policy. CA Identity Manager cannot resolve the policy correctly because it cannot search the attribute.

  Consider setting the attribute to searchable="false" in the directory.xml file—For example:

  <ImsManagedObjectAttr physicalname="title" description="Title" displayname="Title" valuetype="String" maxlength="0" searchable="false">

  <DataClassification name="AttributeLevelEncrypt"/>

  </ImsManagedObjectAttr>

- If CA Identity Manager uses a shared user store and Provisioning Directory, do not encrypt Provisioning Server attributes.

- Do not enable AttributeLevelEncrypt for user passwords in environments that meet the following criteria:

  - include CA SiteMinder integration, and
  - Store users in a relational database

  When CA Identity Manager integrates with CA SiteMinder, encrypted passwords cause issues when new users try to log in, and enter passwords in clear text.

- If you enable attribute-level encryption for a user store that is used by applications other than CA Identity Manager, the other applications cannot use the encrypted attribute.

How To Add Attribute-Level Encryption

Assume that you have added an attribute-level encryption to a CA Identity Manager directory. CA Identity Manager automatically encrypts existing clear text attribute values when you save the object which is associated with the attribute. For example, encrypting the password attribute encrypts the password when it saves the profile of the user.

Note: To encrypt the attribute value, the task that you use to save the object must include the attribute. To encrypt the password attribute in the previous example, make sure that the password field is added to the task you use to save the object, such as the Modify User task.

All new objects are created with encrypted values in the user store.

Follow these steps:

1. Complete one of the following tasks:
   - Create a CA Identity Manager directory
   - Update an existing directory by exporting the directory settings.
2. Add the following data classification attributes to the attribute that you want to encrypt in the directory.xml file:

   AttributeLevelEncrypt

   Persists the attribute value in an encrypted form in the user store.

   sensitive (optional)

   Hides the attribute value in CA Identity Manager screens. For example, a password is displayed as asterisks (*).

   For example:
   ```
   <ImsManagedObjectAttr physicalname="salary" displayname="Salary"
   description="salary" valuetype="String" required="false"
   multivalued="false" maxlength="0" searchable="false">
   <DataClassification name="AttributeLevelEncrypt"/>
   <DataClassification name="sensitive"/>
   ```
3. If you have created a CA Identity Manager Directory, associate the directory with an environment.
4. To force CA Identity Manager to encrypt all values immediately, modify all objects using the Bulk Loader.

   Note: For more information about the Bulk Loader, see the *Administration Guide*.

How to Remove Attribute-Level Encryption

If you have an encrypted attribute in the CA Identity Manager Directory and it is stored with the value of that attribute as a clear text, then you can remove the AttributeLevelEncrypt data classification.

Once the data classification has been removed, CA Identity Manager stops encrypting the new attribute values. Existing values are decrypted when you save the object which is associated with the attribute.

Note: To decrypt the attribute value, the task that you use to save the object must include the attribute. For example, to decrypt a password for an existing user, you save the user object with a task that includes the password field, such as the Modify User task.

To force CA Identity Manager to detect and decrypt any encrypted values that remain in the user store for the attribute, you can specify another data classification, PreviouslyEncrypted. The clear text value is saved to the user store when you save the object.

Note: Adding the PreviouslyEncrypted data classification adds extra processing on every object load. To prevent performance issues, consider adding the PreviouslyEncrypted data classification, loading and saving each object that is associated with that attribute, and then removing the data classification. This method automatically converts all stored encrypted values to stored clear text.

Follow these steps:

1. Export the directory settings for the appropriate CA Identity Manager Directory.
2. In the directory.xml file, remove the data classification, AttributeLevelEncrypt, from attributes that you want to decrypt.
3. If you want to force CA Identity Manager to remove previously encrypted values, add the PreviouslyEncrypted data classification attribute.

   For example:

   ```
   <ImsManagedObjectAttr physicalname="salary" displayname="Salary"
   description="salary" valuetype="String" required="false" multivalued="false"
   maxlength="0" searchable="false">
   <DataClassification name="PreviouslyEncrypted"/>
   ```
4. To force CA Identity Manager to decrypt all values immediately, modify all objects using the Bulk Loader.

   Note: For more information about the Bulk Loader, see the *Administration Guide*.

CA Directory Considerations

When you describe attributes for a CA Directory user store, note the following points:

- Attribute names are case-sensitive.
- Using the seeAlso attribute as the attribute that indicates a self-subscribing group may cause errors when administrators create groups.

Using the photo attribute as the attribute that indicates status of a user account (enabled or disabled) may cause errors when an administrator creates a user.

Note: For additional information about CA Directory requirements, see the CA Directory documentation.

Microsoft Active Directory Considerations

When you describe attributes for Active Directory, note the following points:

- The case of the attributes specified in attribute descriptions must match the case of the attributes in Active Directory. For example, when you select the unicodePwd attribute as the attribute to stores user passwords, specify the unicodePwd (with a capital P) in the directory configuration file.
- For user and group objects, make sure that you include the sAMAccountName attribute.

IBM Directory Server Considerations

When you describe attributes for an IBM Directory Server user directory, see the following sections:

- Groups in Directory Server Directories (see page 73)
- The Objectclass "Top" in the Organization Object Description (see page 74)

Groups in Directory Server Directories

IBM Directory Server requires groups to contain at least one member. To address this requirement, CA Identity Manager adds a *dummy user* as a member of a new group when the group is created.

Configure a Dummy User

Follow these steps:

1. In the Group Object section of the directory configuration file, locate the following elements:

   ```
   <PropertyDict name="DUMMY_USER">
       <Property name="DUMMY_USER_DN">##DUMMY_USER_DN</Property>
   </PropertyDict>
   ```

   Note: If these elements do not exist in the directory configuration file, add them exactly as they appear here.

2. Replace ##DUMMY_USER_DN with a user DN. CA Identity Manager adds this DN as a member of all new groups.

   Note: If you specify the DN of an existing user, then, that user appear as a member of all the groups of CA Identity Manager. To prevent the *dummy user* from appearing as a group member, specify a DN that does not exist in the directory.

3. Save the directory configuration file.

The Objectclass Top in the Organization Object Description

Important! In the description of the organization object in the directory configuration file, do not include the objectclass top.

For example, when the objectclass of the organization object is top, organizationalUnit, specify the objectclass as follows:

```
<ImsManagedObject name="Organization" description="My Organizations"
objectclass="organizationalUnit" objecttype="ORG">
```

Including top can cause unpredictable search results.

Oracle Internet Directory Considerations

When you describe attributes for an Oracle Internet Directory (OID) user store, specify LDAP attributes using lowercase letters only.

Well-Known Attributes for an LDAP User Store

Well-known attributes have special meaning in CA Identity Manager. They are identified as shown in the following syntax:

%ATTRIBUTENAME%

In this syntax, *ATTRIBUTENAME* must be uppercase.

A well-known attribute is mapped to one physical attribute, using an attribute description (see page 114).

In the following attribute description, the attribute userpassword is mapped to the well-known attribute %PASSWORD% so that CA Identity Manager treats the value in userpassword as a password as follows:

```
<ImsManagedObjectAttr
        physicalname="userpassword"
        displayname="Password"
        description="Password"
        valuetype="String"
        required="false"
        multivalued="false"
        wellknown="%PASSWORD%"
        maxlength="0" />
```

Some well-known attributes are required; others are optional.

User Well-Known Attributes

A list of user well-known attributes and the items to which they map follows:

%ADMIN_ROLE_CONSTRAINT%

Maps to the list of admin roles of an administrator.

The physical attribute that mapped to %ADMIN_ROLE_CONSTRAINT% must be multivalued to accommodate multiple roles.

We recommend indexing the LDAP attribute that is mapped to %ADMIN_ROLE_CONSTRAINT%.

%CERTIFICATION_STATUS%

Maps to the certification status of a user.

This attribute is required to use the user certification feature.

Note: For more information about user certification, see the *Administration Guide*.

%DELEGATORS%

Maps to a list of users who have delegated work items to the current user.

This attribute is required to use delegation. The physical attribute that mapped to %DELEGATORS% must be multivalued and capable of holding strings.

Important! Editing this field directly using CA Identity Manager tasks or an external tool can cause significant security implications.

%EMAIL%

Maps to an email address of a user.

Required to use the email notification feature.

%ENABLED_STATE%

(Required)

Maps to the status of a user.

Note: This attribute must match the Disabled Flag user directory attribute in the SiteMinder user directory connection.

%FIRST_NAME%

Maps to the first name of a user.

%FULL_NAME%

Maps to the first and last names of a user.

%IDENTITY_POLICY%

Specifies the list of identity policies that have been applied to a user account and a list of unique Policy Xpress policy IDs that have performed add or remove actions on the user object.

CA Identity Manager uses this attribute to determine whether applying an identity policy to a user is required or not. Assume that the policy has the Apply Once setting enabled and the policy is listed in the %IDENTITY_POLICY% attribute. CA Identity Manager does not apply the changes in the policy to the user.

Note: For more information about identity policies, see the *Administration Guide*.

%LAST_CERTIFIED_DATE%

Maps to the date when the roles are certified to a user.

Required to use the user certification feature.

Note: For more information about user certification, see the *Administration Guide*.

%LAST_NAME%

Maps to the last name of a user.

%MEMBER_OF%

Maps to the list of groups of which the user is a member.

The physical attribute that mapped to %MEMBER_OF% must be multivalued to accommodate multiple groups.

Using this attribute improves response time when searching groups of a user.

You can use this attribute with Active Directory or any directory schema that maintains group membership of a user on the user object.

%ORG_MEMBERSHIP%

(Required)

Maps to the DN of the organization to which the user belongs.

CA Identity Manager uses this well-known attribute to determine structure of a directory (see page 81).

This attribute is not required when the user directory does not include organizations.

%ORG_MEMBERSHIP_NAME%

(Required)

Maps to the user-friendly name of the organization in which the profile of the user exists.

This attribute is not required when the user directory does not include organizations.

%PASSWORD%

Maps to the password of a user.

This attribute must match the Password Attribute in the SiteMinder user directory connection.

Note: The value of the %PASSWORD% attribute is always displayed as a series of asterisk (*) characters in CA Identity Manager screens, even when the attribute or field is not set to hide passwords.

%PASSWORD_DATA%

(Required for password policy support)

Specifies the attribute that tracks password policy information.

Note: The value of the %PASSWORD_DATA% attribute is always displayed as a series of asterisk (*) characters in CA Identity Manager screens, even when the attribute or field is not set to hide passwords.

%PASSWORD_HINT%

(Required)

Maps to a user-specified question and answer pair. The question and answer pair is used when users forget their passwords.

To support multiple question and answer pairs, make sure that the %PASSWORD_HINT% attribute is multivalued.

If you are using Password Services feature of SiteMinder to manage passwords, the Password Hint attribute must match the Challenge/Response attribute in the SiteMinder user directory.

Note: The value of the %PASSWORD% attribute is always displayed as a series of asterisk (*) characters in CA Identity Manager screens, even when the attribute or field is not set to hide passwords.

%USER_ID%

(Required)

Maps to the ID of a user.

Group Well-Known Attributes

The following items are the list of group well-known attributes:

%GROUP_ADMIN_GROUP%

Indicates which attribute stores a list of groups that are administrators of the group. For example, when group 1 is an administrator of group A, group 1 is stored in the %GROUP_ADMIN_GROUP% attribute.

Note: If you do not specify a %GROUP_ADMIN_GROUP% attribute, CA Identity Manager stores administrator groups in the %GROUP_ADMIN% attribute.

Note: To add a group as an administrator of another group, see the *Administration Guide*.

%GROUP_ADMIN%

Indicates which attribute contains the DNs of administrators of a group.

The physical attribute that mapped to %GROUP_ADMIN% must be multivalued.

%GROUP_DESC%

Indicates which attribute contains description of a group.

%GROUP_MEMBERSHIP%

(Required)

Indicates which attribute contains a list of the member of a group.

The physical attribute that mapped to %GROUP_MEMBERSHIP% must be multivalued.

The %GROUP_MEMBERSHIP% well-known attribute is not required for Provisioning user directories.

%GROUP_NAME%

(Required)

Indicates which attribute stores a group name.

%ORG_MEMBERSHIP%

(Required)

Indicates which attribute contains the DN of the organization to which the group belongs.

CA Identity Manager uses this well-known attribute to determine structure of the directory (see page 81).

This attribute is not required when the user directory does not include organizations.

%ORG_MEMBERSHIP_NAME%

Indicates which attribute contains the user-friendly name of the organization in which the group exists.

This attribute is not valid for user directories that do not include organizations.

%SELF_SUBSCRIBING%

Indicates which attribute determines whether users can subscribe to a group (see page 80).

%NESTED_GROUP_MEMBERSHIP%

Indicates which attribute stores a list of groups that are members of the group. For example, when group 1 is a member of group A, group 1 is stored in the %NESTED_GROUP_MEMBERSHIP% attribute.

If you do not specify a %NESTED_GROUP_MEMBERSHIP% attribute, CA Identity Manager stores nested groups in the %GROUP_MEMBERSHIP% attribute.

To include groups as members of other groups, configure support for nested groups as described in Configuring Dynamic and Nested Groups for instructions.

%DYNAMIC_GROUP_MEMBERSHIP%

Indicates which attribute stores the LDAP query that generates a dynamic group (see page 138).

Note: To extend the available attributes for the Group object to include %NESTED_GROUP_MEMBERSHIP% and %DYNAMIC_GROUP_MEMBERSHIP% attributes, you can use auxiliary object classes.

Organization Well-Known Attributes

The following well-known attributes apply only to environments that support organizations:

%ORG_DESCR%

Indicates which attribute contains description of an organization.

%ORG_MEMBERSHIP%

(Required)

Indicates which attribute contains the DN of the parent organization of an organization.

%ORG_MEMBERSHIP_NAME%

Indicates which attribute contains the user-friendly name of the parent organization of an organization.

%ORG_NAME%

(Required)

Indicates which attribute contains the name of the organization.

%ADMIN_ROLE_CONSTRAINT% Attribute

When you create an admin role, you specify one or more rules for role membership. Users who satisfy the membership rules are granted the role. For example, when the membership rule for the User Manager role is title=User Manager, users who have the title User Manager possess the User Manager role.

Note: For more information about rules, see the *Administration Guide*.

%ADMIN_ROLE_CONSTRAINT% enables you to designate a profile attribute to store the admin roles of an administrator.

How to Use the %ADMIN_ROLE_CONSTRAINT% Attribute

To use %ADMIN_ROLE_CONSTRAINT% as the constraint for all admin roles, do the following tasks:

- Pair the %ADMIN_ROLE_CONSTRAINT% well-known attribute with a multivalued profile attribute to accommodate multiple roles.
- When you configure an admin role in the User Console, make sure about the following constraint:

  `Admin Roles equals` *`role name`*

  role name

  Defines the name of the role for which you are supplying the constraint, as in the following example:

  ```
  Admin Roles equals User Manager
  ```

Note: Admin Roles is the default displayname for the %ADMIN_ROLE_CONSTRAINT% attribute.

Configure Well-Known Attributes

Perform the following procedure to configure well-known attributes.

Follow these steps:

1. In the directory configuration file, search for the following sign:

   ##

2. Replace the value that begins with ## with the appropriate LDAP attribute.
3. Repeat Steps 1 and 2 until you have replaced all required values.
4. Map optional well-known attributes to physical attributes, as necessary.
5. Save the directory configuration file.

Describe the User Directory Structure

CA Identity Manager uses the %ORG_MEMBERSHIP% well-known attribute to determine the structure of a user directory.

The procedure for describing the user directory structure depends on the type of directory structure.

How to Describe a Hierarchical Directory Structure

The directory configuration file is already configured for a hierarchical directory structure. As a result, you do not have to modify the %ORG_MEMBERSHIP% attribute description.

How to Describe a Flat User Directory Structure

Follow these steps:

1. Locate the %ORG_MEMBERSHIP% attribute description in the User Object section of the directory.xml file.
2. In the physicalname parameter, replace %ORG_MEMBERSHIP% with the name of the attribute that stores the organization to which the user belongs.

How to Describe a Flat Directory Structure

Follow these steps:

1. Locate the %ORG_MEMBERSHIP% attribute description in the User Object section of the directory.xml file.
2. In the physicalname parameter, replace %ORG_MEMBERSHIP% with the name of the attribute that stores the organization to which the user belongs.
3. Repeat Step 1 in the Group Object section.
4. In the physicalname parameter, replace %ORG_MEMBERSHIP% with the name of the attribute that stores the organization to which the group belongs.

How to Describe a User Directory that Does Not Support Organizations

Verify that no object descriptions or well-known attributes are defined for organizations in directory.xml.

How to Configure Groups

For configuration, groups can be divided as follows:

- Self-subscribing groups
- Dynamic and Nested groups

Configure Self-Subscribing Groups

You can enable self-service users to join groups by configuring support for self-subscribing groups in the directory configuration file.

When a user self-registers, CA Identity Manager looks for groups in specified organizations and then, displays the self-subscribing groups to the user.

Follow these steps:

1. In the Self-subscribing Groups section, add a SelfSubscribingGroups element as follows:

   <SelfSubscribingGroups type=*search_type* org=*org_dn*>

2. Add values for the following parameters:

   type

   Indicates where CA Identity Manager searches for self-subscribing groups as follows:

   - NONE—CA Identity Manager does not search for groups. Specify NONE to prevent users from self-subscribing to groups.
   - ALL—CA Identity Manager begins searching for groups at the root. Specify ALL when users can subscribe to groups throughout a hierarchical directory.
   - INDICATEDORG—CA Identity Manager searches for self-subscribing groups in the organization of a user and its suborganizations. For example, when the profile of a user is in the Marketing organization, CA Identity Manager searches for self-subscribing groups in the Marketing organization, and in all suborganizations.
   - SPECIFICORG—CA Identity Manager searches in a specific organization. Supply the distinguished name (DN) of the specific organization in the org parameter.

org

Specifies the unique identifier of the organization where CA Identity Manager searches for self-subscribing groups.

Note: Make sure that you specify the org parameter when type=SPECIFICORG.

Once support for self-subscribing groups is configured in the CA Identity Manager directory, CA Identity Manager administrators can specify which groups are self-subscribing in the User Console.

Note: For more information about managing groups, see the *Administration Guide*.

Configure Dynamic and Nested Groups

If you are managing an LDAP user store, you can configure support for the following types of groups in the directory configuration file:

Dynamic Groups

Enables you to define group membership by specifying an LDAP filter query in the User Console dynamically. With dynamic groups, administrators do not have to search for and add group members individually.

Nested Groups

Enables you to add groups as members of other groups.

You can enable dynamic and nested groups using the directory configuration file.

Follow these steps:

1. Map the following well-known attributes (see page 78) to a physical attribute for the Group managed object as needed:
   - %DYNAMIC_GROUP_MEMBERSHIP%
   - %NESTED_GROUP_MEMBERSHIP%

   Note: The physical attribute that you select must support multiple values.

2. In the Directory Groups Behavior section, add the following GroupTypes element:

   `<GroupTypes type=group>`

   Note: GroupTypes is case-sensitive.

3. Type a value for the following parameter:

   group

   Enables support for dynamic and nested groups. The valid values are as follows:

   - NONE—CA Identity Manager does not support dynamic and nested groups.
   - ALL—CA Identity Manager supports dynamic and nested groups.
   - DYNAMIC—CA Identity Manager supports dynamic groups only.
   - NESTED—CA Identity Manager supports nested groups only.

Once support for dynamic and nested groups is configured in the CA Identity Manager directory, CA Identity Manager administrators can specify which groups are dynamic and nested in the User Console.

Note: Consider that you have set the group type to NESTED or ALL *without* setting the %NESTED_GROUP_MEMBERSHIP% well-known parameter. In such case, CA Identity Manager stores both the nested groups and users in the %GROUP_MEMBERSHIP% well-known parameter. Processing group membership may be slightly slower.

Add Support for Groups as Administrators of Groups

If you are managing an LDAP user store, you can enable groups to serve as administrators of other groups. When you assign a group as an administrator, only administrators of that group are administrators of the specified group. Members of the administrator group you specify have no privileges to manage the group.

Follow these steps:

1. Map the %GROUP_ADMIN_GROUP% well-known attribute to a physical attribute that stores the list of groups that serve as administrators.

   Note: The physical attribute that you select must support multiple values.

   Group Well-Known Attributes (see page 78) provides more information about the %GROUP_ADMIN_GROUP% attribute.

   Note:If you set the admin group type to ALL without setting the %GROUP_ADMIN_GROUP% well known, CA Identity Manager stores administrator groups in the %GROUP_ADMIN% attribute.

2. In the Directory AdminGroups Behavior section, configure the AdminGroupTypes element as follows:

   <AdminGroupTypes type="ALL">

   The default AdminGroupTypes is NONE.

   Note: "AdminGroupTypes" is case-sensitive.

Once support for groups as administrators is configured in the CA Identity Manager directory, CA Identity Manager administrators can specify groups as administrators of other groups in the User Console.

Validation Rules

A validation rule enforces requirements on data that a user types in a task screen field. The requirements can enforce a data type or format. So, make sure that the data is valid in the context of other data on the task screen.

Validation rules are associated with profile attributes. CA Identity Manager makes sure that the data entered for a profile attribute satisfies any associated validation rules before processing a task.

You can define validations rules and can associate them with profile attributes in the directory configuration file.

Additional CA Identity Manager Directory Properties

You can configure the following additional properties:

- Sort order for search results.
- Search across object classes to verify that a new user does not exist already.
- Wait time to avoid CA Identity Manager timed out before completion of the replication of data, from the master LDAP directory to the slave LDAP directory.

Configure Sort Order

You can specify a sort attribute for each managed object, such as users, groups, or organizations. CA Identity Manager uses this attribute to sort search results in custom business logic, which you create with the CA Identity Manager APIs.

Note: The sort attribute does not affect the way search results are displayed in the User Console.

For example, when you specify the cn attribute for the user object, CA Identity Manager sorts the results of a search for users alphabetically by the cn attribute.

Follow these steps:

1. After the last IMSManagedObjectAttr element in the section for the managed object to which the sort order applies, add the following statements:

   ```
   <PropertyDict name="SORT_ORDER">
       <Property name="ATTR">your_sort_attribute
       </Property>
   </PropertyDict>
   ```

2. Replace *your_sort_attribute* with the attribute on which CA Identity Manager sorts the search results.

Note: Specify only one physical attribute. Do not specify a well-known attribute.

For example, assume that you have to sort user search results based on the value of the cn attribute. Add the following elements after the last IMSManagedObjectAttr element in the User Object section of the directory configuration file:

```
<!-- ******************* User Object ******************* -->
   <ImsManagedObject name="User" description="My Users"
      objectclass="top,person,organizationalperson,user"
      objecttype="USER">
   .
   .
   .
        <ImsManagedObjectAttr physicalname="departmentnumber"
                displayname="Department" description="Department"
                valuetype="String" required="true"
                multivalued="false" maxlength="0" />
        <PropertyDict name="SORT_ORDER">
                <Property name="ATTR">cn</Property>
        </PropertyDict>
  </ImsManagedObject>
```

Search across Objectclasses

CA Identity Manager searches the user store to check if the user exists or not when you create a user. This search is limited to users who have the objectclasses specified in the user object definition in the directory configuration file (directory.xml). If no existing user is found in those objectclasses, CA Identity Manager tries to create the user.

If a user exists with the same unique identifier (user ID) but a different objectclass, the LDAP server fails to create the user. The error is reported in the LDAP server, but CA Identity Manager does not recognize the error. CA Identity Manager appears to create the user successfully.

To prevent this issue, you can configure a SEARCH_ACROSS_CLASSES property that causes CA Identity Manager to search users across all objectclass definitions when checking for existing users.

Note: This property affects only searches for duplicate users when performing tasks such as creating a user. For all other searches, objectclass constraints apply.

Follow these steps:

1. In the directory configuration file (directory.xml), locate the ImsManagedObject element that describes the user object.

2. Add the following PropertyDict element:

```
<PropertyDict name="SEARCH_ACROSS_CLASSES" description="allowing checking an
attribute across classes ">
<Property name="ENABLE">true</Property>
</PropertyDict>
```

Note: The PropertyDict element must be the last element in the ImsManagedObject element, as in the following example:

```
<ImsManagedObject name="User" description="My Users"
objectclass="top,person,organizationalperson,inetorgperson,customClass"
objecttype="USER">
<ImsManagedObjectAttr physicalname="departmentnumber" displayname="Department"
description="Department" valuetype="String" required="true"
multivalued="false" maxlength="0" />
.
.
.
<PropertyDict name="SEARCH_ACROSS_CLASSES" description="allow checking an
attribute across classes ">
<Property name="ENABLE">true</Property>
</PropertyDict>
```

Specify Replication Wait Time

In a deployment that includes replication between master and slave LDAP directories, you can configure the SiteMinder Policy Server to communicate with a slave directory. In this configuration, the Policy Server automatically detects referrals that point to the master directory during operations that write data to the LDAP directory. The data is stored in the master LDAP directory and replicated to the slave LDAP directory according to the replication scheme of your network resources.

In this configuration, when you create an object in CA Identity Manager, the object is created in the master directory and it is also replicated to the slave directory. A delay may occur during the replication process that causes the create action to fail in CA Identity Manager.

To prevent this issue from occurring, you can specify the amount of time (in seconds) that CA Identity Manager waits before "timing out" in the REPLICATION_WAIT_TIME property.

Follow these steps:

1. In the directory configuration file (directory.xml), locate the ImsManagedObject element that describes the user object.
2. Add the following PropertyDict element:

```
<PropertyDict name="REPLICATION_WAIT_TIME" description="time delay in seconds
for LDAP provider to allow replication to propagate from master to slave">
<Property name=REPLICATION_WAIT_TIME"><time in seconds></Property>
</PropertyDict>
```

Note: The PropertyDict element must be the last element in the ImsManagedObject element, as in the following example:

```
<ImsManagedObject name="User" description="My Users"
objectclass="top,person,organizationalperson,inetorgperson,customClass"
objecttype="USER">
<ImsManagedObjectAttr physicalname="departmentnumber" displayname="Department"
description="Department" valuetype="String" required="true"
multivalued="false" maxlength="0" />
.
.
.
<PropertyDict name="REPLICATION_WAIT_TIME" description="time delay in seconds
for LDAP provider to allow replication to propagate from master to slave">
<Property name=REPLICATION_WAIT_TIME">800</Property>
</PropertyDict>
```

When the replication wait time is not defined, the default value 0 is used.

Specify LDAP Connection Settings

To improve performance, you can specify the following parameters in the directory configuration file (directory.xml):

Connection Timeout

Specifies the maximum number of milliseconds that CA Identity Manager searches a directory before terminating the search.

This property is specified in the directory configuration file as follows:

com.sun.jndi.ldap.connect.timeout

Connection Pool Max Size

Specifies the maximum number of connections that CA Identity Manager can make to the LDAP directory.

This property is specified in the directory configuration file as follows:

com.sun.jndi.ldap.connect.pool.maxsize

Connection Pool Default Size

Specifies the default number of connections between CA Identity Manager and the LDAP directory.

This property is specified in the directory configuration file as follows:

com.sun.jndi.ldap.connect.pool.prefsize

Follow these steps:

1. In the directory configuration file (directory.xml), locate the ImsManagedObject element that describes the user object.
2. Add the following PropertyDict element:

```
<PropertyDict name="LDAP_CONNECTION_SETTINGS" description="LDAP Connection
Settings">
    <Property name="com.sun.jndi.ldap.connect.timeout">5000</Property>
    <Property name="com.sun.jndi.ldap.connect.pool.maxsize">200</Property>
    <Property name="com.sun.jndi.ldap.connect.pool.prefsize">10</Property>
</PropertyDict>
```

3. Save the directory.xml file.

   CA Identity Manager configures these settings when you create the CA Identity Manager directory with this file.

How to Improve Directory Search Performance

To improve the performance of directory searches for users, organizations, and groups, do the following points:

- Index the attributes that administrators can specify in search queries.

  Note: For Oracle Internet Directory, a search may fail when an attribute in a search query is not indexed.

- Configure page size and maximum row settings (see page 91) to determine how CA Identity Manager handles large searches.

- Tune the user directory. See the documentation for the user directory that you are using.

How to Improve Performance for Large Searches

When CA Identity Manager manages a huge user store, searches that return many results can cause the system to run out of memory. To help prevent memory issues, you can define limits for large searches.

The following two settings determine how CA Identity Manager handles large searches:

- Maximum number of rows

  Specifies the maximum number of results that CA Identity Manager can return when searching a user directory. When the number of results exceeds the limit, an error is displayed.

- Page size

  Specifies the number of objects that can be returned in a single search. If the number of objects exceeds the page size, CA Identity Manager performs multiple searches.

  Note the following points when specifying Page Size:

  - To use the Search Page Size option, the user store that CA Identity Manager manages must support paging. Some user store types require additional configuration to support paging. For more information, see the following topics:

    Configure Sun Java System Directory Server Paging Support (see page 92)

    Configure Active Directory Paging Support

  - If the user store does not support paging and a value for maxrows is specified, CA Identity Manager uses only the maxrows value to control search size.

You can configure maximum row limits and page size in the following places:

- User Store

  In most user stores and databases, you can configure search limits.

  Note: For more information, see the documentation for the user store or database that you are using.

- CA Identity Manager Directory

  You can configure the DirectorySearch element (see page 55) in the directory configuration file (directory.xml) that you use to create the CA Identity Manager Directory.

  By default, the value for maximum rows and page size is unlimited for existing directories. For new directories, the value for maximum rows is unlimited and the value for page size is 2000.

- Managed object definition

  To set maximum row limits and page sizes applying to one type of object instead of an entire Directory, configure the *managed object definition* (see page 57) in the directory.xml file that you use to create the CA Identity Manager Directory.

  Setting limits for a managed object type allows you to do adjustments that are based on business requirements. For example, most companies have more users than groups. Those companies can set limits for user object searches only.

- Task Search screens

  You can control the number of search results that users see in search and list screens in the User Console. If the number of results exceeds the number of results per page that are defined for the task, users see links to additional pages of results.

  This setting does not affect the number of results that are returned by a search.

  Note: For information on setting page size in search and list screens, see the *Administration Guide*.

If maximum row limits and page sizes are defined in multiple places, the most specific setting applies. For example, managed object settings take precedence over directory level settings.

Configure Sun Java System Directory Server Paging Support

Sun Java System Directory Servers support Virtual List View (VLV), a method for delivering search results in a certain order or in certain subsets. This method differs from Simple Paged Results, which CA Identity Manager expects.

To use VLV, you set permissions and create indexes. CA Identity Manager includes the following files that you must configure paging support:

- vlvcntrl.ldif
- vlvindex.ldif
- runvlvindex.cmd, runvlvindex.sh

These files are included as part of the NeteAuto sample, in samples\NeteAuto in the Administrative Tools.

The Administrative Tools are installed in the following default locations:

Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager

UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/

Follow these steps:

1. Add the following parameter to the DirectorySearch element (see page 55) in the directory.xml file for the CA Identity Manager directory as follows:

   ```
   minsortrules="1"
   ```

   Note: If you are modifying an existing CA Identity Manager directory, see How to Update a CA Identity Manager Directory (see page 172).

2. Set permissions for the vlvcntrl.ldif file as follows:
   `ldapmodify -D "cn=Directory Manager" -w` *`password`* `-p` *`port`* `-f vlvcntrl.ldif`

3. Import VLV Search and Index definitions as follows:
   `ldapmodify -D "cn=Directory Manager" -w` *`password`* `-p` *`port`* `-f vlvindex.ldif`

4. Stop the directory as follows:
   `stop-slapd`

5. Build the indexes using runvlvindex.

6. Start the directory as follows:
   `start-slapd`

Configure Active Directory Paging Support

To configure paging support in Active Directory, complete the following high-level steps:

- Configure support for Virtual List View (see page 93).
- Configure the MaxPageSize for Active Directory (see page 94). (**For Directories created before CA Identity Manager r12.5 SP7 only**)

Configure Support for Virtual List View (VLV)

Active Directory supports Virtual List View (VLV), a method for delivering search results in a certain order or in certain subsets. This method differs from Simple Paged Results, which CA Identity Manager expects.

To use VLV, you set permissions and create indexes. CA Identity Manager includes the following files that you must configure paging support:

- vlvcntrl.ldif
- vlvindex.ldif
- runvlvindex.cmd, runvlvindex.sh

These files are included as part of the NeteAuto sample, in samples\NeteAuto in the Administrative Tools.

The Administrative Tools are installed in the following default locations:

Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager

UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/

Follow these steps:

1. Add the following parameter to the DirectorySearch element (see page 55) in the directory.xml file for the CA Identity Manager directory as follows:

   ```
   minsortrules="1"
   ```

   Note: If you are modifying an existing CA Identity Manager directory, see How to Update a CA Identity Manager Directory (see page 172).

2. Set permissions for the vlvcntrl.ldif file as follows:
   ```
   ldapmodify -D "cn=Directory Manager" -w password -p port -f
   vlvcntrl.ldif
   ```

3. Import VLV Search and Index definitions as follows:
   ```
   ldapmodify -D "cn=Directory Manager" -w password -p port -f
   vlvindex.ldif
   ```

4. Stop the directory as follows:
   ```
   stop-slapd
   ```

5. Build the indexes using runvlvindex.

6. Start the directory as follows:
   ```
   start-slapd
   ```

Configure the Active Directory MaxPageSize

Active Directory uses 1000 as the default MaxPageSize. Assume that the maxpagesize attribute value in directory.xml is greater than or equal to 1000. In such case, CA Identity Manager fails to display a warning when the number of search results exceeds the maxrows value in directory.xml. In this case, administrators who perform the search are not aware that some search results are omitted.

To prevent this issue, verify that the maxpagesize attribute value for the Directory and each managed object is less than the Active Directory MaxPageSize.

Assume that you are creating a CA Identity Manager Directory using the template directory.xml file that is installed with CA Identity Manager 12.5 SP7 or higher. In this case, you do not need to perform any additional steps for paging support. The maxpagesize attribute in directory.xml is set by default.

If you are adding paging support to an existing CA Identity Manager Directory, the maxpagesize attribute in directory.xml must be less than 1000.

Also, if the Active Directory MaxPageSize is 1000, be sure to set the maxpagesize attribute appropriately for the CA Identity Manager Directory and all managed objects.

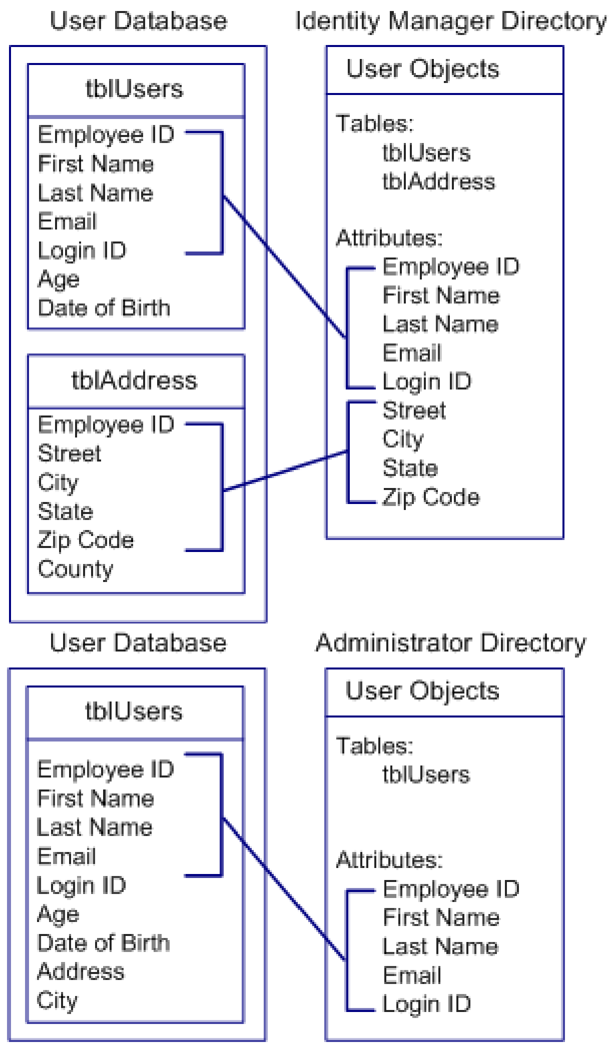
Chapter 4: Relational Database Management

This section contains the following topics:

CA Identity Manager Directories

A *CA Identity Manager directory* describes how objects, such as users, groups, and (optionally) organizations are stored in the user store and how they are represented in CA Identity Manager. A CA Identity Manager directory is associated with one or more CA Identity Manager environments.

The following illustration shows how a CA Identity Manager directory relates to a user store:



Note: Some user attributes in the database are not part of the CA Identity Manager directory. Therefore, CA Identity Manager does not manage them.

Important Notes When Configuring CA Identity Manager for Relational Databases

Before you configure CA Identity Manager to manage a relational database, ensure that the database meets the following requirements:

- The database must be accessible through a JDBC driver or an Open Database Connectivity (ODBC) driver (when CA Identity Manager integrates with SiteMinder). The driver must support outer joins. If more than two tables are used to represent a managed object, the driver must also support nested outer joins.

  Note: If the driver does not support outer joins, CA Identity Manager uses inner joins when querying the database. This can cause unexpected query results.

- Uniquely identify each object that CA Identity Manager manages, such as a user, group, or organization (when supported). For example, the unique identifier for users may be a login ID.

  Note: Make sure that the unique identifier is stored in a single column.

- CA Identity Manager requires some multivalued attributes, which can be stored as a delimited list in a single cell or in multiple rows in a separate table. For example, the following tblGroupMembers table stores the members of a group:

ID	Members
Research	dmason
Research	rsavory
Marketing	dmason
Marketing	awelch

  The ID column contains the unique identifier for a group and the Members column contains the unique identifier for a member of the group. For example, dmason and rsavory are members of the Research group. When a new member is added to that group, another row is added to tblGroupMembers.

- When your environment includes organizations, do the following task:

  - Edit and run a SQL script, included with CA Identity Manager, against the database to configure organization support (see page 140).

  - CA Identity Manager requires a top-level organization, named the root. All other organizations relate to the root organization.

    For more information about organization requirements, see Organizations Management (see page 139).

Create an Oracle Data Source for WebSphere

Follow these steps:

1. In the WebSphere Administrative Console, navigate to the JDBC provider that you created when you configured the JDBC driver.
2. Create a data source with the following properties and click Apply:

   Name: User Store Data Source

   JNDI Name: userstore

   URL: jdbc:oracle:thin:@*db_systemname*:1521:*oracle_sid*
3. Configure a new J2C Authentication Data Entry for the User Store Data Source:
   a. Enter the following properties:

      Alias: User Store

      User ID: *username*

      password: *password*

      where *username* and *password* are the username and password for the account you specified when you created the database.
   b. Click OK, then use the navigation links at the top of the screen to return to the data source you are creating.
4. Select the User Store J2C Authentication Data Entry that you have created from the list box in the following fields:
   - Component-managed Authentication Alias
   - Container-managed Authentication Alias
5. Click OK, then save the configuration.

   Note: To verify that the data source is configured correctly, click Test Connection in the configuration screen for the data source. If the test connection fails, restart WebSphere and test the connection again.

How to Create a CA Identity Manager Directory

Follow these steps:

1. If you are using SiteMinder, apply the policy store schema prior creating a CA Identity Manager Directory.

   Note: For more information on specific policy store schemas and how to apply them, see the *Installation Guide*.

2. If you are using SiteMinder, create an ODBC data source for use with SiteMinder (see page 107).

3. Create a data source for the user database that CA Identity Manager manages.

4. Describe the database to CA Identity Manager by modifying a directory configuration file (directory.xml). For more information, see How to Describe a Database in a Directory Configuration File.

5. In the Management Console, import the directory configuration file and create the directory.

How To Create a JDBC Data Source

CA Identity Manager requires a JDBC datasource in the application server where CA Identity Manager is installed to connect to the user store. The instructions for creating a datasource are different for each application server.

Create a JDBC Data Source for JBoss Application Servers

Follow these steps:

1. Create a copy of the following file:

   jboss_home\server\default\deploy\objectstore-ds.xml

   jboss home

   The installed location of the Jboss application server where CA Identity Manager is installed.

   The new file must exist in the same location.

2. Rename the file to userstore-ds.xml.

3. Edit userstore-ds.xml as follows:

   a. Locate the <jndi-name> element.

   b. Change the value of the <jndi-name> element from jdbc/objectstore to userstore as follows:

      <jndi-name>userstore</jndi-name>

c. In the <connection-url> element, change the DatabaseName parameter to the name of the database that serves as the user store as follows:

<connection-url>

jdbc:sqlserver://*ipaddress:port*;selectMethod=cursor;DatabaseName=*userstore_name*

</connection-url>

ipaddress

Specifies the IP address of the machine where the user store is installed.

port

Specifies the port number for the database

userstore_name

Specifies the name of the database that serves as the user store.

4. Perform the following steps if you plan to create a JBoss security realm, which is required for support FIPS:

   a. Rename the security-domain to <security-domain>imuserstoredb</security-domain>.

   b. Save the file.

   c. Omit the remaining steps. Instead, complete the steps in Create a JBoss Security Realm for the JDBC Data Source (see page 103).

5. Make the following additional changes to userstore-ds.xml:

   a. Change the value of the <user-name> element to the username for an account that has read and write access to the user store.

   b. Change the value of the <password> element to the password for the account specified in the <user-name> element.

   Note: The user-name and password appear in clear text in this file. Therefore, you may decide to create a JBoss Security realm instead of editing userstore-ds.xml.

6. Save the file.

Use a JBoss Security Realm for the JDBC Data Source

Assure that you are creating a JDBC data source in a JBoss application server. You can configure the data source to use a user name and password or can configure it to use a security realm.

Important! Make sure that the JBoss Security Realm option is used if FIPS is being used.

Follow these steps:

1. Complete the steps in Create a JDBC Data Source for JBoss Application Servers (see page 101).

   Do not specify a user name and password in the userstore-ds.xml as described in step 4.

2. Open login-cfg.xml in *jboss_home*\server\default\conf.

3. Locate the following entry in the file:

```
<application-policy name="imobjectstoredb">
  <authentication>
    <login-module
code="com.netegrity.jboss.datasource.PasswordEncryptedLogin" flag="required">
      <module-option name="userName">fwadmin</module-option>
      <module-option
name="password">{PBES}:gSex2/BhDGzEKWvFmzca4w==</module-option>
      <module-option
name="managedConnectionFactoryName">jboss.jca:name=jdbc/objectstore,service=N
oTxCM</module-option>
    </login-module>
  </authentication>
</application-policy>
```

4. Copy the complete entry and paste it within the <policy> and </policy> tags in the login-cfg.xml file.

5. In the entry you pasted in the file, make the following changes:

   a. Change the name attribute value from imobjectstoredb to imuserstoredb as follows:

   ```
   <application-policy name="imuserstoredb">
   ```

   b. Specify the name of the user used to authenticate against the user store as follows:

   ```
   <module-option name="userName">user_store_user</module-option>
   ```

   c. Specify the password for the user in the previous step as follows:

   ```
   <module-option name="password">user_store_user_password</module-option>
   ```

   Note: To encrypt the user store password, use the password tool (pwdtools) that is installed with CA Identity Manager.

d. In the <module-option name="managedConnectionFactoryName"> element, provide the correct jdbc.jca:name as follows:

```
<module-option name="managedConnectionFactoryName">
   jboss.jca:name=userstore,service=NoTxCM
</module-option>
```

6. Save the file.
7. Restart the application server.

Create a JDBC Data Source for WebLogic

You create a data source in the WebLogic Administration Console.

Note: See the Oracle WebLogic 11 Documentation for complete information about Weblogic Connection Pools.

Follow these steps:

1. Create a JDBC Data Source with the following parameters in the WebLogic Administration Console:

   Name: User Store Data Source

   JNDI Name: userstore

2. Create the connection pool for the data source with the following information:

   - For SQL Server 2005 databases, use the following values:

     URL: jdbc:sqlserver://*db_systemName*:1433

     Driver Class Name: com.microsoft.sqlserver.jdbc.SQLServerDriver

     Properties: user=*username*

     databaseName=*user store name*

     selectMethod=cursor

     Password: *password*

   - For Oracle databases, use the following values:

     URL: jdbc:oracle:thin:@*tp_db_systemname*:1521:*oracle_SID*

     Driver Class Name: oracle.jdbc.driver.OracleDriver

     Properties: user=*username*

     Password: *password*

3. After configuration, set the target for the pool to the server instance *wl_server_name*.

   After you deploy the pool, check the console to see if any errors occurred.

   Note: You may see an error that says the data source cannot be created with a non-existent pool. To resolve this error, restart WebLogic.

WebSphere Data Sources

The following sections describe how to create a SQL or Oracle data source for WebSphere application servers.

Create a SQL Server Data Source for WebSphere

Follow these steps:

1. In the WebSphere Administrative Console, navigate to the JDBC provider that you created when you configured the JDBC driver.
2. Select Data Sources in the Additional Properties section.
3. Create a data source with the following properties and click Apply:

   Name: User Store Data Source

   JNDI Name: userstore

   databaseName: *userstore_name*

   serverName: *db_systemname*

4. Configure the selectMethod property as follows:
   a. Select Custom Properties in the Additional Properties section.
   b. Click the selectMethod custom property.
   c. Enter the following text in the Value field:

      cursor

   d. Click OK, then use the navigation links at the top of the screen to return to the data source you are creating.

5. Configure a new J2C Authentication Data Entry for the User Store Data Source:

   a. Select J2EE Connector Architecture (J2C) authentication data entries from the Related Items section.

   b. Click New.

   c. Enter the following properties:

      Alias: User Store

      User ID: *username*

      password: *password*

      where *username* and *password* are the username and password for the account you specified when you created the database.

   d. Click OK, then use the navigation links at the top of the screen to return to the data source you are creating.

6. Select the User Store J2C Authentication Data Entry that you created from the list box in the Component-managed Authentication Alias field.

7. Click OK, then save the configuration.

   Note: To verify that the data source is configured correctly, click Test Connection in the configuration screen for the data source. If the test connection fails, restart WebSphere and test the connection again.

Create an Oracle Data Source for WebSphere

Follow these steps:

1. In the WebSphere Administrative Console, navigate to the JDBC provider that you created when you configured the JDBC driver.

2. Create a data source with the following properties and click Apply:

   Name: User Store Data Source

   JNDI Name: userstore

   URL: jdbc:oracle:thin:@*db_systemname*:1521:*oracle_sid*

3. Configure a new J2C Authentication Data Entry for the User Store Data Source:

   a. Enter the following properties:

      Alias: User Store

      User ID: *username*

      password: *password*

      where *username* and *password* are the username and password for the account you specified when you created the database.

   b. Click OK, then use the navigation links at the top of the screen to return to the data source you are creating.

4. Select the User Store J2C Authentication Data Entry that you have created from the list box in the following fields:

   - Component-managed Authentication Alias
   - Container-managed Authentication Alias

5. Click OK, then save the configuration.

   Note: To verify that the data source is configured correctly, click Test Connection in the configuration screen for the data source. If the test connection fails, restart WebSphere and test the connection again.

How to Create an ODBC Data Source For Use with SiteMinder

If CA Identity Manager integrates with SiteMinder, define an ODBC data source on the SiteMinder machine that points to the database. Note the name of the data source for later use. Proceed as follows:

- **Windows:** Configure the ODBC data source as a System DN. See your Windows operating system documentation for instructions.
- **UNIX:** Add an entry specifying the parameters for the ODBC data source in the system_odbc.ini file located in *policy_server_installation*/db.

How to Describe a Database in a Directory Configuration File

To manage a database, CA Identity Manager must understand the database structure and content. To describe the database to CA Identity Manager, create a directory configuration file (directory.xml).

The directory configuration file contains one or more of the following sections:

CA Identity Manager Directory Information

Contains information about the CA Identity Manager directory that CA Identity Manager uses.

Attribute Validation

Defines the validation rules that apply to the CA Identity Manager directory.

Provider Information

Describes the user store that CA Identity Manager manages.

Directory Search Information

Enables you to specify how CA Identity Manager searches the user store.

User Object (see page 109)

Describes how users are stored in the user store and how they are represented in CA Identity Manager.

Group Object (see page 109)

Describes how groups are stored in the user store and how they are represented in CA Identity Manager.

Organization Object (see page 109)

Describes how organizations are stored and how they are represented in CA Identity Manager.

Self-Subscribing Groups

Configures support for groups that self-service users can join.

The directory where you installed the administrative tools for CA Identity Manager includes the following directory configuration file template for relational databases:

admin_tools\directoryTemplates\RelationalDatabase\directory.xml

admin_tools

Defines the installed location of CA Identity Manager administrative tools, as in the following examples:

- **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

Note: The directory configuration file template in directoryTemplates\RelationalDatabase is configured for environments that support organizations. To see a directory configuration file for an environment that does not include organizations, you can look at the directory.xml file for the NeteAuto sample located in *admin_tools*\samples\NeteAutoRDB\NoOrganization

Copy the configuration template to a new directory or save it with a different name to prevent overwriting it. You can then modify the template to reflect your database structure.

The directory configuration file has two important conventions:

- ##—Indicates required values.

  To provide all of the required information, locate all double pound signs (##) and replace them with appropriate values. For example, ##PASSWORD_HINT indicates that you must supply an attribute to store a question that a user answers to receive a temporary password in the case of a forgotten password.

- @—Indicates values that CA Identity Manager populates. Do not modify these values in the directory configuration file. CA Identity Manager prompts you to supply the values when you import the directory configuration file.

Before you modify the directory configuration file, you need the following information:

- Table names for the user, group, and organization objects (when your structure includes organizations).
- A list of attributes in user, group, and organization profiles (when your structure includes organizations).

Modify the Directory Configuration File

Perform the following procedure to modify the directory configuration file.

Follow these steps:

1. Configure a connection to the database.
2. Specify the amount of time that CA Identity Manager searches a directory before terminating the search.
3. Define the user and group managed objects that CA Identity Manager manages (see page 109).
4. Modify well-known attributes.

   Well-known attributes identify special attributes, such as the password attribute, in CA Identity Manager.
5. Configure support for self-subscribing groups.
6. If your environment includes organizations, configure organization support.

More information:

Managed Object Descriptions (see page 109)
Organization Management (see page 139)
How to Configure Self-Subscribing Groups (see page 138)
Well-Known Attributes for a Relational Database (see page 132)

Managed Object Descriptions

In CA Identity Manager, you manage the following types of objects, corresponding to entries in a user store:

- Users—Represent users in an enterprise.
- Groups—Represent associations of users who have something in common.

- (Optional) Organizations—Represent business units. Organizations may contain users, groups, and other organizations.

  Note: Organization Management (see page 139) provides information about configuring organizations.

An object description contains the following information:

- Information about the object (see page 110), such as the tables in which the object is stored.
- The attributes that store information about an entry (see page 114). For example, the pager attribute stores a pager number.

Important! A CA Identity Manager environment supports only one type of user, group, and organization object.

How to Describe a Managed Object

A managed object is described by specifying object information in the User Object, Group Object, and Organization Object sections (when the database includes organizations) of the directory configuration file.

Each of these sections contains an ImsManagedObject element, such as the following code:

```
<ImsManagedObject name="User" description="My Users">
```

The ImsManagedObject element may include the following elements:

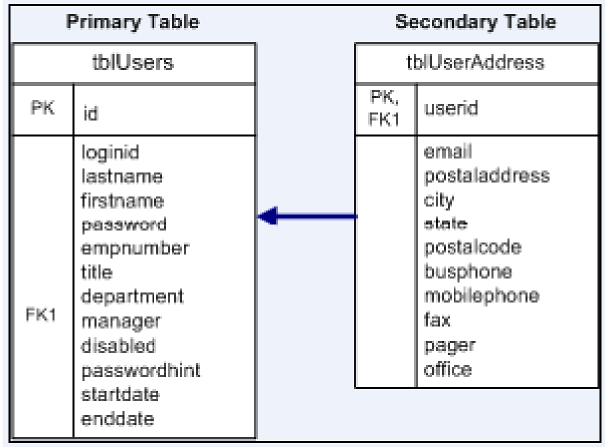
- Table (required)
- UniqueIdentifier (required)
- ImsManagedObjectAttr (required)
- RootOrg (for organization objects only)

Database Tables

Use the Table element in the directory configuration file to define the tables that store information about a managed object.

Each managed object must have one primary table that contains the unique identifier for the object. Additional information can be stored in secondary tables.

The following illustration shows a database that stores user information in a primary and secondary table:



If the information of an object is stored in multiple tables, create a Table element for each table. Use the Reference element in the Table element for a secondary table to define its relationship to the primary table.

For example, if basic information about a user is stored in tblUsers and address information is stored in tblUserAddress, the table definitions for the User managed object would resemble the following entries:

```
<Table name="tblUsers" primary="true" />
<Table name="tblUserAddress">
  <Reference childcol="userid" primarycol="id" />
</Table name>
```

Table Elements

The parameters for a table element are as follows:

name

(Required)

Specifies the name of the table that stores some or all of the attributes in a managed profile of an object.

primary

Indicates whether the table is the primary table for the managed object. The primary table contains the unique identifier for the object, as follows:

- True—The table is the primary table.
- False—The table is a secondary table (default).

If you do not specify the primary parameter, CA Identity Manager assumes the table as a secondary table.

Note: Only one table can be the primary table.

filter

Identifies a subset of the table entries that apply to the managed object.

The optional filter parameter may resemble the following example:

```
filter="ORG=2"
```

Note: The filter applies only to queries that CA Identity Manager generates. If you overwrite a generated query with a custom query, specify the filter in the custom query.

fullouterjoin

Indicates whether the outer join is a full outer join.

- True— The outer join is a full outer join. In this case, the condition that is required to return a valid row is found in both tables in the join for a returned row.
- False—The outer join is a left outer join relative to the primary table. In this case, only the rows in one table in the query must satisfy the condition (default).

Note: The parameters are optional unless otherwise specified.

The Table parameter can contain one or more Reference elements to link a primary table to secondary tables.

Reference Element

The parameters in the Reference element are as follows:

childcol

Indicates the column in the secondary table (specified in the corresponding Table element) that maps to the column in the primary table.

primarycol

Indicates the column in the primary table that maps to the column in the secondary table.

Note: The parameters are optional unless otherwise specified.

Specify Object Information

Object information is specified by supplying values for various parameters.

Follow these steps:

1. Locate the ImsManagedObject element in the User Object, Group Object, or Organization Object section.
2. Supply values for the following parameters:

   name

   (Required)

   Provides a unique name for the managed object.

   description

   Provides the description of the managed object.

   objecttype

   (Required)

   Specifies the type of managed object. The valid values are as follows:

   - USER
   - GROUP
   - ORGANIZATION

   The ImsManagedObject element must resemble the following code:

   ```
   <ImsManagedObject name="User" description="My  Users" objecttype="USER">
   ```

3. Supply Table information, as described in Database Tables (see page 110).
4. Specify the column that contains the unique identifier for the object (see page 113).
5. Describe the attributes that constitute the profile of the object (see page 114).
6. If you are configuring an organization object, go to Organizations Management (see page 139).

How to Specify the Unique Identifier for a Managed Object

Each object that CA Identity Manager manages must have a unique identifier. Make sure that the unique identifier is stored in a single column in the primary table of the managed object. Primary tables are described in Database Tables (see page 110).

Use the UniqueIdentifier and UniqueIdentifierAttr elements to define the unique identifier as follows:

```
<UniqueIdentifier>
  <UniqueIdentifierAttr name="tablename.columnname" />
</UniqueIdentifier>
```

The UniqueIdentifierAttr element requires the name parameter. The value of the name parameter is the attribute in which the unique identifier is stored. The value can be a physical attribute or a well-known attribute (see page 74).

When you specify a physical attribute, note the following points:

- Make sure that the specified attribute exists in the database and it is defined in the directory configuration file, as described in How to Modify Attribute Descriptions (see page 114). In the attribute description, be sure to specify read-only or write-once permission to prevent the unique identifier from changing during a session.
- Use the following syntax to specify a physical attribute:

  `tablename.columnname`

  tablename

  Defines the name of the table where the attribute is located. The table you specify must be the primary table.

  columnname

  Defines the name of the column that stores the attribute.
- If the database generates the unique identifier, specify a custom operation for the attribute (see page 124). For example, you may have to specify an operation that fetches the last generated identifier from the database.

How to Modify Attribute Descriptions

An attribute stores information about a user, group, or organization entity, such as a telephone number or address. The attributes of an entity determine its profile.

In the directory configuration file, attributes are described in ImsManagedObjectAttr elements. In the User Object, Group Object and Organization Object sections of the directory configuration file:

- Modify default attribute descriptions to describe your database attributes.
- Create new attribute descriptions by copying an existing description and modifying values as needed.

For each attribute in user, group, and organization profiles, there is only one ImsManagedObjectAttr element. For example, an ImsManagedObjectAttr element may describe a user ID.

An ImsManagedObjectAttr element resembles the following code:

```
<ImsManagedObjectAttr
 physicalname="tblUsers.id"
 displayname="User Internal ID"
 description="User Internal ID"
 valuetype="Number"
 required="false"
 multivalued="false"
 maxlength="0"
 hidden="false"
 permission="READONLY">
```

Note: When you are using an Oracle database, note the following points while configuring managed object attributes:

- Oracle databases are case-sensitive by default. The case of the attributes and table names in the directory configuration file must match the case of the attributes in Oracle.

  Be sure to specify a maximum length for String datatypes to prevent truncation. To limit the length of strings, you can create a validation rule to display an error when a user types a string that exceeds the maximum length.

The ImsManagedObjectAttr parameters are as follows.

Note: The parameters are optional unless otherwise specified.

physicalname

(Required)

Specifies the physical name of the attribute, and it must contain one of the following details:

- The name and location where the value is stored.

  Format: *tablename.columnname*

  For example, when an attribute is stored in the id column in the tblUsers table, the physical name for that attribute is as follows:

  tblUsers.id

  You are required to define each table that contains an attribute in a Table element (see page 110).

- A well-known attribute.

  A well-known attribute can represent a computed value. For example, you can use a well-known attribute to refer to an attribute computed using a custom operation (see page 124).

displayname

(Required)

Specifies a unique name for the attribute.

In the User Console, the display name appears in the list of attributes that are available to add to a task screen.

Note: Do not modify the displayname of an attribute in the directory configuration file (directory.xml). To change the name of the attribute on a task screen, you can specify a label for the attribute in the task screen definition. For more information, see the *Administration Guide*.

description

Provides the description of the attribute.

valuetype

Specifies the data type of the attribute. The valid values are as follows:

String

The value can be any string.

This is the default value.

Integer

The value must be an integer.

Note: Integer does not support decimal numbers.

Number

The value must be an integer. The number option supports decimal numbers.

Date

The value must parse to a valid date using the pattern:

MM/dd/yyyy

ISODate

The value must parse to a valid date using the pattern yyyy-MM-dd.

UnicenterDate

The value must parse to a valid date using the pattern YYYYYYYDDD where:

YYYYYYY is a seven number representation of year beginning with three zeros. For example: 0002008

DDD is the three number representation for the day beginning with zeroes, as needed. Valid values include from 001 to 366.

If the valuetype of an attribute is incorrect, CA Identity Manager queries may fail.

To make sure that an attribute is stored correctly in the database, you can associate it with a validation rule.

required

Indicates whether a value is required to specify for the attribute, as follows:

- True—Required
- False—Optional (default)

multi-valued

Indicates whether the attribute can have multiple values, as follows:

- True— An attribute can have multiple values.
- False— An attribute can have only a single value (default).

For example, the group membership attribute in a user profile is multi-valued to store the groups to which a user belongs.

To store multi-valued attributes in a delimited list instead of in a multi-row table, you are required to define the delimiter character in the delimiter parameter.

Make sure that the number of possible values and the length of each value that the column enables are sufficient.

Important! Make sure that the Group Membership attribute in the User object definition is multi-valued.

wellknown

Provides the name of the well-known attribute.

Well-known attributes have a specific meaning in CA Identity Manager.

Format: %*ATTRIBUTENAME*%

Note: When a custom operation is associated with an attribute, you are required to specify a well-known attribute (see page 74).

maxlength

Determines the maximum size of the column.

permission

Indicates whether the value of an attribute can be modified in a task screen, as follows:

READONLY

The value is displayed but cannot be modified.

WRITEONCE

The value cannot be modified once the object is created. For example, a user ID cannot be changed after the user is created.

READWRITE

The value can be modified (default).

hidden

Indicates whether an attribute appears in the CA Identity Manager task screens, as follows:

- True—The attribute is not displayed to users.
- False—The attribute is displayed to users (default).

Logical attributes use hidden attributes.

Note: For more information about logical attributes, see the *Programming Guide for Java*.

system

Indicates that only CA Identity Manager used attributes. Users must not modify the attributes in the User Console, as follows:

- True—Users cannot modify the attribute. The attribute will not appear in the User Console.
- False—Users can modify this attribute and is available to add to task screens in the User Console (default).

validationruleset

Associates a validation rule set with the attribute.

Make sure that the validation rule set that you specify is defined in a ValidationRuleSet element in the directory configuration file.

delimiter

Defines the character that separates values when multiple values are stored in a single column.

Important! Make sure that the multivalued parameter is set to true for the delimiter parameter to apply.

Note: To prevent displaying sensitive information, such as passwords or salaries, in the User Console, you can specify DataClassification (see page 69) parameters.

Managing Sensitive Attributes

CA Identity Manager provides the following methods for managing sensitive attributes:

- Data classifications for attributes

  Data classifications allow you to specify display and encryption properties for attributes in the directory configuration file (directory.xml).

  You can define data classifications that manage sensitive attributes as follows:

  - In CA Identity Manager task screens, display the value of an attribute as a series of asterisks.

    For example, you can display passwords as asterisks instead of displaying them in clear text.

  - In View Submitted Task screens, hide the attribute value.

    This option enables you to hide attributes from administrators. For instance, hiding salary details such as the salary from administrators who view task status in CA Identity Manager but do not need to view salary details.

  - Ignore certain attributes when creating a copy of an existing object.
  - Encrypt an attribute

- Field styles in task profile screens

  If you do not want to modify an attribute in the directory.xml file, set the display property for the attribute in screen definitions where the sensitive attribute appears.

  The field style enables you to display attributes, such as passwords, as a series of asterisks instead of clear text.

  Note: For more information about the field style for sensitive attributes, search for field styles in the User Console help.

Data Classification Attributes

The Data Classification element provides a way to associate additional properties with an attribute description. The values in this element determine how CA Identity Manager handles the attribute. This element supports the following parameters:

- sensitive

  Causes CA Identity Manager to display the attribute as a series of asterisks (*) in View Submitted Tasks screens. This parameter prevents old and new values for the attribute from appearing in clear text in View Submitted tasks screens.

  Additionally, if you create a copy of an existing user in the User Console, this parameter prevents the attribute from being copied to the new user.

- vst_hide

  Hides the attribute in the Event Details screen for the View Submitted Tasks tab. Unlike sensitive attributes, which are displayed as asterisks, vst_hidden attributes are not displayed.

  You can use this parameter to prevent changes to an attribute, such as the salary, from displaying in View Submitted Tasks.

- ignore_on_copy

  Causes CA Identity Manager to ignore an attribute when an administrator creates a copy of an object in the User Console. For example, assume that you have specified ignore_on_copy for the password attribute on a user object. When copying a user profile, CA Identity Manager does not apply the password of the current user to the new user profile.

- AttributeLevelEncrypt

  Encrypts attribute values when they are stored in the user store. If CA Identity Manager is FIPS 140-2 enabled, CA Identity Manager uses RC2 encryption or FIPS 140-2 encryption.

  For more information about FIPS 140-2 support in CA Identity Manager, see the *Configuration Guide*.

  The attributes appear in clear text during runtime.

  Note: To prevent attributes from appearing in clear text in screens, you can also add a sensitive data classification element to encrypted attributes. For more information, see How to Add Attribute-Level Encryption (see page 70).

- PreviouslyEncrypted

  Causes CA Identity Manager to detect and decrypt any encrypted values in the attribute when it accesses the object in the user store.

  You use this data classification to decrypt any previously encrypted values.

  The clear text value is saved to the store when you save the object.

Configure Data Classification Attributes

Follow these steps:

1. Locate the attribute in the directory configuration file.
2. After the attribute description, add the following attribute:

   <DataClassification name="*parameter*">

   parameter

   Represents one of the following parameters:

   sensitive

   vst_hide

   ignore_on_copy

   AttributeLevelEncrypt

   PreviouslyEncrypted

   For example, an attribute description that includes the vst_hide data classification attribute resembles the following code:

```
<ImsManagedObjectAttr physicalname="salary" displayname="Salary"
description="salary" valuetype="String" required="false" multivalued="false"
maxlength="0">
    <DataClassification name="vst_hide"/>
```

Attribute-Level Encryption

You can encrypt an attribute in the user store by specifying an AttributeLevelEncypt data classification for that attribute in the directory configuration file (directory.xml). When attribute-level encryption is enabled, CA Identity Manager encrypts the value of that attribute before storing it in the user store. The attribute is displayed as clear text in the User Console.

Note: To prevent attributes from appearing in clear text in screens, you can also add a sensitive data classification element to encrypted attributes. For more information, see How to Add Attribute-Level Encryption (see page 70).

If FIPS 140-2 support is enabled, the attribute is encrypted using RC2 encryption or FIPS 140-2 encryption.

Before you implement the attribute-level encryption, note the following points:

- CA Identity Manager cannot find encrypted attributes in a search.

  Assume that an encrypted attribute is added to a member, admin, owner policy, or an identity policy. CA Identity Manager cannot resolve the policy correctly because it cannot search the attribute.

  Consider setting the attribute to searchable="false" in the directory.xml file—For example:

  <ImsManagedObjectAttr physicalname="title" description="Title" displayname="Title" valuetype="String" maxlength="0" searchable="false">

  <DataClassification name="AttributeLevelEncrypt"/>

  </ImsManagedObjectAttr>

- If CA Identity Manager uses a shared user store and Provisioning Directory, do not encrypt Provisioning Server attributes.

- Do not enable AttributeLevelEncrypt for user passwords in environments that meet the following criteria:

  - include CA SiteMinder integration, and
  - Store users in a relational database

  When CA Identity Manager integrates with CA SiteMinder, encrypted passwords cause issues when new users try to log in, and enter passwords in clear text.

- If you enable attribute-level encryption for a user store that is used by applications other than CA Identity Manager, the other applications cannot use the encrypted attribute.

How To Add Attribute-Level Encryption

Assume that you have added an attribute-level encryption to a CA Identity Manager directory. CA Identity Manager automatically encrypts existing clear text attribute values when you save the object which is associated with the attribute. For example, encrypting the password attribute encrypts the password when it saves the profile of the user.

Note: To encrypt the attribute value, the task that you use to save the object must include the attribute. To encrypt the password attribute in the previous example, make sure that the password field is added to the task you use to save the object, such as the Modify User task.

All new objects are created with encrypted values in the user store.

Follow these steps:

1. Complete one of the following tasks:
   - Create a CA Identity Manager directory
   - Update an existing directory by exporting the directory settings.
2. Add the following data classification attributes to the attribute that you want to encrypt in the directory.xml file:

   AttributeLevelEncrypt

   Persists the attribute value in an encrypted form in the user store.

   sensitive (optional)

   Hides the attribute value in CA Identity Manager screens. For example, a password is displayed as asterisks (*).

   For example:
   ```
   <ImsManagedObjectAttr physicalname="salary" displayname="Salary"
   description="salary" valuetype="String" required="false"
   multivalued="false" maxlength="0" searchable="false">
   <DataClassification name="AttributeLevelEncrypt"/>
   <DataClassification name="sensitive"/>
   ```
3. If you have created a CA Identity Manager Directory, associate the directory with an environment.
4. To force CA Identity Manager to encrypt all values immediately, modify all objects using the Bulk Loader.

   Note: For more information about the Bulk Loader, see the *Administration Guide*.

How to Remove Attribute-Level Encryption

If you have an encrypted attribute in the CA Identity Manager Directory and it is stored with the value of that attribute as a clear text, then you can remove the AttributeLevelEncrypt data classification.

Once the data classification has been removed, CA Identity Manager stops encrypting the new attribute values. Existing values are decrypted when you save the object which is associated with the attribute.

Note: To decrypt the attribute value, the task that you use to save the object must include the attribute. For example, to decrypt a password for an existing user, you save the user object with a task that includes the password field, such as the Modify User task.

To force CA Identity Manager to detect and decrypt any encrypted values that remain in the user store for the attribute, you can specify another data classification, PreviouslyEncrypted. The clear text value is saved to the user store when you save the object.

Note: Adding the PreviouslyEncrypted data classification adds extra processing on every object load. To prevent performance issues, consider adding the PreviouslyEncrypted data classification, loading and saving each object that is associated with that attribute, and then removing the data classification. This method automatically converts all stored encrypted values to stored clear text.

Follow these steps:

1. Export the directory settings for the appropriate CA Identity Manager Directory.
2. In the directory.xml file, remove the data classification, AttributeLevelEncrypt, from attributes that you want to decrypt.
3. If you want to force CA Identity Manager to remove previously encrypted values, add the PreviouslyEncrypted data classification attribute.

   For example:

   ```
   <ImsManagedObjectAttr physicalname="salary" displayname="Salary"
   description="salary" valuetype="String" required="false" multivalued="false"
   maxlength="0" searchable="false">
   <DataClassification name="PreviouslyEncrypted"/>
   ```
4. To force CA Identity Manager to decrypt all values immediately, modify all objects using the Bulk Loader.

   Note: For more information about the Bulk Loader, see the *Administration Guide*.

Custom Operations

You can define custom operations for certain managed objects to do the following tasks:

- Use stored procedures
- Optimize queries for their database structure.
- Retrieve a database-generated unique identifier

Custom operations apply only to attributes.

When specifying custom operations, remember the following points:

- Users who specify custom operations must be familiar with SQL.
- CA Identity Manager does not validate custom operations. Until runtime, Syntax errors and invalid queries are not reported.
- If you specify a custom operation for an attribute, you cannot use that attribute in search filters in CA Identity Manager tasks.
- Custom operations must conform to XML standards. Represent special characters using XML syntax. For example, specify a single quotation mark (’) as '

To specify a custom operation, use the Operation element.

Operation Element

The Operation element defines a SQL statement which can execute a custom query or calls a stored procedure for creating, retrieving, modifying, or deleting an attribute. Operation element is a subelement of the IMSManagedObjectAttr element, as shown in the following example:

```
<ImsManagedObjectAttr physicalname="tblUsers.id" displayname="User Internal ID"
description="User Internal ID" valuetype="Number" required="false"
multivalued="false" maxlength="0" hidden="false" permission="READONLY">
  <Operation name="GetDb" value="select @@identity" />
```

Operation element parameters are as follows:

name

Specifies a predefined name for an operation. The valid operations are as follows:

- Create
- Get
- Set
- Delete
- GetDB

  The GetDB operation retrieves a unique identifier from the database during a Create task, when the unique identifier is generated through the database or from a stored procedure.

value

Defines the SQL statement or stored procedure to execute. The valid values are as follows:

- INSERT
- SELECT
- UPDATE
- DELETE
- CALL (for stored procedures)

Note: The parameters are optional unless otherwise specified.

The Operation element can contain one or more Parameter elements.

Parameter Element

A Parameter element specifies values that are passed to the query. When multiple Parameter elements are defined, the values are passed to the query in the given listed order.

A Parameter element requires the name parameter. The value of the name parameter can be a physical attribute or a well-known attribute (see page 74).

Note: CA Identity Manager must understand the values that are passed to a query in the Parameter element. For example, the value can be a physical name or a well-known attribute that is defined in the ImsManagedObjectAttr attributes.

When you specify a physical attribute, note the following points:

- Use the following syntax to specify a physical attribute:

  tablename.columnname

  – *tablename*

    Provides the name of the table where the attribute is located. The table you specify must be the primary table.

  – *columnname*

    Provides the name of the column that stores the attribute.

- The attribute that you specify must exist in the database and it is defined in the directory configuration file, as described in How to Modify Attribute Descriptions (see page 114).

Example: Custom Operations for the Business Number Attribute

In the following example, the Business Number attribute is generated by calling a stored procedure; it is not a physical attribute in the database.

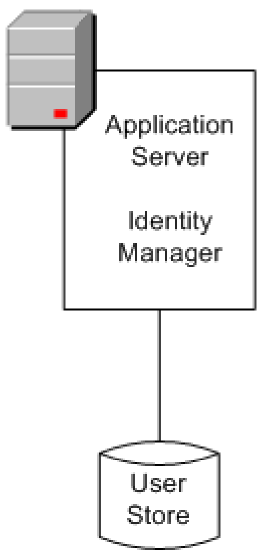
```
<ImsManagedObjectAttr wellknown="%BUSINESS_NUMBER%"     displayname="Business
Number" description="Business Number"     valuetype="String" required="false"
multivalued="false"     maxlength="0">
<Operation name="Get" value="call sp_getbusinessnumber(?)">
    <Parameter name="%USER_ID%"/>
</Operation>
<Operation name="Set" value="call sp_setbusinessnumber(?,?)">
    <Parameter name="%USER_ID%"/>
    <Parameter name="%BUSINESS_NUMBER%"/>
</Operation>
<Operation name="Delete" value="call sp_deletebusinessnumber(?)">
    <Parameter name="%USER_ID%"/>
</Operation>
```

Note the following points:

- sp_getbusinessnumber, sp_setbusinessnumber, and sp_deletebusinessnumber are user-defined stored procedures.
- The value that is returned from the Get operation is mapped to the %BUSINESS_NUMBER% attribute.
- The question mark (?) indicates substitutions that are made at runtime before the query is executed. For example, in the Get operation the %USER_ID% well-known attribute is passed to the sp_getbusinessnumber stored procedure.

Connection to the User Directory

CA Identity Manager connects to a user directory to store information such as a user, group, and organizational information as shown in the following illustration:



A new directory or database is not required. However, the existing directory or database must be on a system that has a fully qualified domain name (FQDN).

For a list of supported directory and database types, see the CA Identity Manager support matrix on the CA Support Site.

You configure a connection to the user store when you create a CA Identity Manager directory in the Management Console.

If you export the directory configuration after creating a CA Identity Manager directory, the user directory connection information is displayed in the Provider element of the directory configuration file.

Description of a Database Connection

To describe a database connection, use the Provider element and its subelements in the directory.xml file.

Note: If you are creating a CA Identity Manager directory, you do not need to provide directory connection information in the directory.xml file. You provide connection information in the CA Identity Manager Directory wizard in the Management Console.

Modify the Provider element for updates only.

Provider Element

The Provider element includes the following subelements:

JDBC (required)

Identifies the JDBC data source to use when connecting to the user store. Specify the JNDI name that you have provided when creating the JDBC data source (see page 101).

Credentials (required)

Supplies the username and password for accessing the database.

DSN

Identifies the ODBC data source to use when connecting to the user store.

Note: This subelement only applies when CA Identity Manager integrates with SiteMinder. In CA Identity Manager environments that do not include SiteMinder, this subelement is ignored.

SiteMinderQuery

Specifies custom query schemes for locating user information in a relational database.

Note: This subelement only applies when CA Identity Manager integrates with SiteMinder. In CA Identity Manager environments that do not include SiteMinder, this subelement is ignored.

A completed database connection resembles the following example:

```
<Provider type="RDB" userdirectory="@SMDirName">
        <JDBC datasource="@SMDirJDBCDataSource"/>
        <Credentials user="@SMDirUser"
                cleartext="true">@SMDirPassword</Credentials>
        <DSN name="@SMDirDSN" />
        <SiteMinderQuery name="AuthenticateUser" query="SELECT TBLUSERS.LOGINID
FROM    TBLUSERS WHERE TBLUSERS.LOGINID='%s' AND TBLUSERS.PASSWORD='%s'" />
</provider>
```

The attributes for the Provider element are as follows:

type

Specifies the type of database. For Microsoft SQL Server and Oracle databases, specify RDB (default).

userdirectory

Specifies the name of the user directory connection. This parameter corresponds to the Connection Object name that you supply during directory creation.

If CA Identity Manager integrates with SiteMinder for authentication, it creates a user directory connection in SiteMinder with the name you specify for the Connection Object during installation. If you want to connect to an existing SiteMinder user directory, enter the name of that user directory when prompted for the Connection Object. CA Identity Manager populates the userdirectory parameter with the name you specify.

If CA Identity Manager does not integrate with SiteMinder, the value of the userdirectory parameter is any name that you give the JDBC connection to the user store.

Note: Do not specify a name for the user directory connection in the directory.xml file. CA Identity Manager prompts you to supply the name during directory creation.

Database Credentials

To connect to the database, CA Identity Manager must provide valid credentials to the data source. The credentials are defined in the Credentials element, which resembles the following example:

```
<Credentials user="@SMDirUser" cleartext="true">
  "MyPassword"
</Credentials>
```

If you do not specify a password in the Credentials element and trying to create the CA Identity Manager directory in the Management Console, it prompts for the password credentials.

Note: We recommend specifying the password in the Management Console.

If you specify the password in the Management Console, CA Identity Manager encrypts the password for you. Otherwise, if you do not want the password to appear in clear text, encrypt the password using the password tool that is installed with CA Identity Manager. SiteMinder Passwords have instructions on using the password tool.

Note: You can specify only one set of credentials. When you define multiple data sources, the credentials that you specify must apply to all data sources.

The credential parameters are as follows:

user

Defines the login ID for an account that can access the data source.

Do not specify a value for the user parameter in the directory.xml file. CA Identity Manager prompts you to supply the login ID when you create the CA Identity Manager directory in the Management Console.

cleartext

Determines whether the password is displayed in clear text in the directory.xml file:

- True—The password is displayed in clear text.
- False—The password is encrypted (default).

Note: These parameters are optional.

Data Source Name (DSN)

The DSN element in the directory.xml file has one parameter—the name of the ODBC data source that CA Identity Manager uses to connect to the database. The value of the name parameter must match the name of an existing data source.

Note: This element applies only when CA Identity Manager integrates with SiteMinder. If CA Identity Manager does not integrate with SiteMinder, this element is ignored.

If the value of the name parameter is @SmDirDSN, you do not have to specify a DSN name in the directory.xml file. CA Identity Manager prompts you to supply the DSN name when you import the directory.xml file.

To configure a failover, define multiple DSN elements. If the primary data source fails to respond to a request, the next data source that is defined responds to the request.

For example, assume that you have configured the failover in the following way:

```
<DSN name="DSN1">
<DSN name="DSN2">
```

CA Identity Manager uses the data source DSN1 to connect to the database. If there is a problem with DSN1, CA Identity Manager tries to connect to the database using DSN2.

Note: The credentials you specify in the Credentials element (see page 129) must apply to all the DSNs that you define.

SQL Query Schemes

CA Identity Manager uses query schemes to find a user and group information in a relational database.

Note: This element applies only when CA Identity Manager integrates with SiteMinder. In environments that do not include SiteMinder, this parameter is ignored.

When you create a CA Identity Manager directory in the Management Console, CA Identity Manager generates a set of query schemes that are based on the required query schemes in SiteMinder. (For complete information about SiteMinder query schemes, see the *CA SiteMinder Web Access Manager Policy Server Configuration Guide*.) The table and column names in the SiteMinder query schemes are replaced with data that you specify in the directory configuration file.

How to Define Custom Query Schemes

Query schemes are defined in SiteMinderQuery elements in the directory configuration file. A SiteMinderQuery element resembles the following:

```
<SiteMinderQuery name="SetUserProperty" query="update tblUsers set %s =
&apos;%s&apos; where loginid = &apos;%s&apos;" />
```

Note: In the sample query, ' is the XML syntax for the single quote (’).

The SiteMinderQuery element only applies when CA Identity Manager integrates with SiteMinder.

The query scheme parameters are as follows:

name

Specifies the redefined name of a SiteMinder query scheme.

Do not modify this value.

query

Specifies the SQL statement or stored procedure to execute. The valid values are as follows:

- INSERT
- SELECT
- UPDATE
- DELETE
- CALL (for stored procedures)

Note: These parameters are required for the SiteMinderQuery element.

Before you customize query schemes, do the following points:

- Become familiar with the default query schemes.

  Note: For more information on SQL query schemes see the *CA SiteMinder Web Access Manager Policy Server Configuration Guide*.

- Acquire extensive experience developing SQL queries.

Modify the Default Query Schemes

Perform the following procedure to modify the default query schemes.

Follow these steps:

1. Export the directory configuration file.

   CA Identity Manager generates a directory configuration file that contains all the current settings for the CA Identity Manager directory, including the generated query schemes.

2. Save the directory configuration file.

   Note: If you want to create a backup of the original directory configuration file, save the file with a different name or in a different location before saving the exported file.

3. Locate the CA Identity Manager-generated query scheme that you want to modify.
4. Enter the query scheme or stored procedure to execute in the query parameter.

   Note: Do not modify the query name.

5. After the necessary changes made, save the directory configuration file.

   Import the file to update the CA Identity Manager Directory (see page 173).

Well-Known Attributes for a Relational Database

Well-known attributes have special meaning in CA Identity Manager. They are identified by the following syntax:

%*ATTRIBUTENAME*%

In this syntax, *ATTRIBUTENAME* must be uppercase.

A well-known attribute is mapped to one physical attribute using an attribute description (see page 114).

In the following attribute description, the attribute tblUsers.password is mapped to the well-known attribute %PASSWORD% so that CA Identity Manager treats the value in tblUsers.password as a password:

```
<ImsManagedObjectAttr
        physicalname="tblUsers.password"
        displayname="Password"
        description="Password"
        valuetype="String"
        required="false"
        multivalued="false"
        wellknown="%PASSWORD%"
        maxlength="0" />
```

Some well-known attributes are required; others are optional.

User Well-Known Attributes

A list of user well-known attributes follows:

%ADMIN_ROLE_CONSTRAINT%

Contains the list of admin roles (see page 136) that are assigned to the administrator (see page 136)..

The physical attribute which is mapped to %ADMIN_ROLE_CONSTRAINT% must be multivalued to accommodate multiple roles.

We recommend indexing the attribute that is mapped to %ADMIN_ROLE_CONSTRAINT%.

%CERTIFICATION_STATUS%

(Required for using the user certification feature)

Contains the certification status of a user.

Note: For more information about user certification, see the *Administration Guide*.

%DELEGATORS%

Maps to a list of users who have delegated work items to the current user.

This attribute is required to use delegation. The physical attribute that mapped to %DELEGATORS% must be multivalued and capable of holding strings.

Important! Editing this field directly using CA Identity Manager tasks or an external tool can cause significant security implications.

%EMAIL%

(Required for enabling the email notification feature)

Stores email address of a user.

%ENABLED_STATE%

(Required)

Tracks the status of a user.

Note: The data type of the physical attribute which is mapped to %ENABLED_STATE% must be String.

%FIRST_NAME%

Contains first name of a user.

%FULL_NAME%

(Required)

Contains first and last name of a user.

%IDENTITY_POLICY%

Contains the list of identity policies that have been applied to a user account.

CA Identity Manager uses this attribute to determine whether an identity policy must be applied to a user. If the policy has the Apply Once setting enabled and the policy is listed in the %IDENTITY_POLICY% attribute, CA Identity Manager does not apply the changes in the policy to the user.

Note: For more information about identity policies, see the *Administration Guide*.

%LAST_CERTIFIED_DATE%

(Required for using the user certification feature)

Contains the date when the role of a user were certified.

Note: For more information about user certification, see the *Administration Guide*.

%LAST_NAME%

Contains last name of a user.

%ORG_MEMBERSHIP%

(Required when organizations are supported)

Contains the unique identifier for the organization to which the user belongs.

%ORG_MEMBERSHIP_NAME%

(Required when organizations are supported)

Contains the user-friendly name of the organization to which the user belongs.

%PASSWORD%

Contains a user password.

Note: The value of the %PASSWORD% attribute is always displayed as a series of asterisk (*) characters in CA Identity Manager screens, even when the attribute or field is not set to hide passwords.

%PASSWORD_DATA%

(Required for password policy support)

Specifies the attribute that tracks password policy information.

Note: The value of the %PASSWORD_DATA% attribute is always displayed as a series of asterisk (*) characters in CA Identity Manager screens, even when the attribute or field is not set to hide passwords.

%PASSWORD_HINT%

(Required)

Contains user-specified question and answer pairs. The question and answer pairs are used in case of forgotten passwords.

Note: The value of the %PASSWORD_HINT% attribute is always displayed as a series of asterisk (*) characters in CA Identity Manager screens, even when the attribute or field is not set to hide passwords.

%USER_ID%

(Required)

Stores a user login ID.

Group Well-Known Attributes

A list of group well-known attributes follows:

%GROUP_ADMIN%

Contains administrators of a group.

Note: The %GROUP_ADMIN% attribute must be multivalued.

%GROUP_DESC%

Contains description of a group.

%GROUP_ID%

Contains the unique identifier for a group.

%GROUP_MEMBERSHIP%

(Required)

Contains a list of member of a group.

Note: The %GROUP_MEMBERSHIP% attribute must be multivalued.

%GROUP_NAME%

(Required)

Stores the name of a group.

%ORG_MEMBERSHIP%

(Required when organizations are supported).

Contains the unique identifier for the organization to which the group belongs.

%ORG_MEMBERSHIP_NAME%

(Required when organizations are supported).

Contains the user-friendly name of the organization to which the group belongs.

%SELF_SUBSCRIBING%

Determines whether users can subscribe to a group.

%Admin_Role_Constraint% Attribute

When you create an admin role, you specify one or more rules for role membership. Users who satisfy the membership rules have the role. For example, if the membership rule for the User Manager role is title=User Manager, users who have the title User Manager have the User Manager role.

Note: For more information about rules, see the *Administration Guide*.

%ADMIN_ROLE_CONSTRAINT% lets you designate one profile attribute to store all of the admin roles of an administrator.

How to Use the %ADMIN_ROLE_CONSTRAINT% Attribute

To use the %ADMIN_ROLE_CONSTRAINT% as the constraint for all admin roles, do the following tasks:

- Pair the %ADMIN_ROLE_CONSTRAINT% well-known attribute with a multivalued profile attribute to accommodate multiple roles.

- When you configure an admin role in the CA Identity Manager user interface, the following scenario can be a constraint:

  Admin Roles equals *role name*

  role name

  Defines the name of the role for which you are supplying the constraint.

  For example, Admin Roles equals User Manager

Note: Admin Roles is the default displayname for the %ADMIN_ROLE_CONSTRAINT% attribute.

Configure Well-Known Attributes

Perform the following procedure to configure well-known attributes.

Follow these steps:

1. In the directory configuration file, search for the following sign:

   ##

   Required values are identified by two pound signs (##).

2. Replace the value that begins with ## with the physical name of the attribute you want as it exists in the database. Supply the attribute name using the following format:

   tablename.columnname

   For example, if the password attribute is stored in the password column in the tblUsers table, specify it in the following way:

   `tblUsers.password`

3. Repeat Steps 1 and 2 until you have replaced all required values and included optional values that you want.

4. Map optional well-known attributes to physical attributes, as necessary.

5. Save the directory configuration file.

How to Configure Self-Subscribing Groups

You can enable self-service users to join groups by configuring support for self-subscribing groups in the directory configuration file.

Follow these steps:

1. In the Self-subscribing Groups section, add a SelfSubscribingGroups element as follows:

   ```
   <SelfSubscribingGroups type=search_type org=org_dn>
   ```

2. Type values for the following parameters:

   type

   Indicates where CA Identity Manager searches for self-subscribing groups. The valid values are as follows:

   - NONE—CA Identity Manager does not search for groups. Specify NONE to prevent users from subscribing to groups.
   - ALL—CA Identity Manager searches all groups in the user store. Specify ALL when users can subscribe to all groups.
   - INDICATEDORG (*for environments that support organizations only*)—CA Identity Manager searches for self-subscribing groups in the organization of a user and its suborganizations. For example, when the profile of a user is in the Marketing organization, CA Identity Manager searches for self-subscribing groups in the Marketing organization, and in all suborganizations.
   - SPECIFICORG (*for environments that support organizations only*)—CA Identity Manager searches in a specific organization. Supply the unique identifier of the specific organization in the org parameter.

   org

   Defines the unique identifier of the organization where CA Identity Manager searches for self-subscribing groups.

   Note: Make sure that you specify the org parameter if type=SPECIFICORG.

3. Restart the SiteMinder Policy Server if you changed any of the following items:

   - The type parameter to or from SPECIFICORG
   - The value of the org parameter

Once support for self-subscribing groups is configured in the CA Identity Manager directory, CA Identity Manager administrators can specify which groups are self-subscribing in the User Console.

When a user self-registers, CA Identity Manager looks for groups in the specified organizations and it displays the self-subscribing groups to the user.

Validation Rules

A validation rule enforces requirements on data that a user types in a task screen field. The requirements can enforce a data type or format or can make sure that the data is valid in the context of other data on the task screen.

Validation rules are associated with profile attributes. Before a task is processed, CA Identity Manager ensures that the data entered for a profile attribute satisfies any associated validation rules.

You can define validation rules and can associate them with profile attributes in the directory configuration file.

Organization Management

For relational databases, CA Identity Manager has the option to manage organizations. When your database supports organizations, the following points are true:

- Organizations have a hierarchical structure.
- All managed objects, such as users, groups, and other organizations belong to an organization.
- When you delete an organization, the objects that belong to that organization are also deleted.

You configure the organization object in the same way that you configure the user and group objects with a few additional steps.

How to Set Up Organization Support

Implement the following steps to set up organization support:

1. Configure Organization Support in the Database (see page 140).
2. Describe the organization object in the ImsManagedObject (see page 110).

   Be sure to configure the Table and UniqueIdentifier subelements.
3. Configure the top-level organization (see page 140).
4. Describe the attributes (see page 114) that constitute an organization.
5. Define the well-known attributes for the organization object (see page 141).

Configure Organization Support in the Database

Follow these steps:

1. Open one of the following SQL scripts in an editor:
   - Microsoft SQL Server databases:

     ims_mssql_rdb.sql
   - Oracle databases:

     ims_oracle_rdb.sql

     These files are placed in the following location:

     admin_tools\directoryTemplates\RelationalDatabase

     admin_tools refers to the installed location of the Administrative Tools, which are installed by default in one of the following locations:

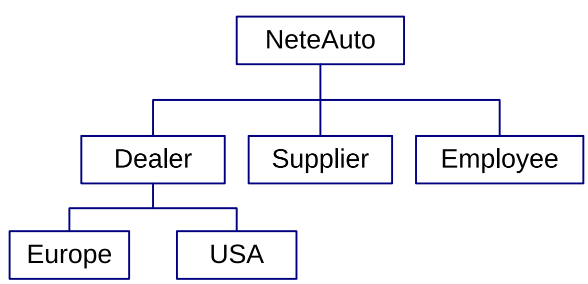
     Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools

     UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools
2. In the SQL script, search for and replace <@primary organization table@> with the name of the primary table for the organization object. Save the SQL script.
3. Run the SQL script against the database.

Root Organization Specification

The root organization serves as the top-level or parent organization in the directory. All organizations relate to the root organization.

In the following illustration, NeteAuto is the root organization. The other organizations are suborganizations of NeteAuto:



A complete root organization definition resembles the following sample:

```
<ImsManagedObject name="Organization" description="My  Organizations"
objecttype="ORG">
   <RootOrg value="select orgid from tblOrganizations where parentorg is null">
      <Result name="%ORG_ID%" />
   </RootOrg>
```

After you define the basic information for the organization object including the tables that constitute the organization profile and the unique identifier of the organization object, specify the root organization in the directory.xml file:

- In the value parameter of the RootOrg element, define the query that CA Identity Manager uses to retrieve the root organization, as in the following example:

  ```
  <RootOrg value="select orgid from tblOrganizations where parentorg  is null">
  ```

- In the name parameter of the Result element, type the unique identifier of the organization, as in the following example:

  ```
  <Result name="%ORG_ID%" />
  ```

  Note: The value of the name parameter must be the unique identifier for the organization object.

Well-Known Attributes for Organizations

Define well-known attributes for the attributes in the profile of an organization profile as described in Well-Known Attributes (see page 74).

The required and optional organization well-known attributes are as follows:

%ORG_DESCR%

Contains description of an organization.

%ORG_MEMBERSHIP%

(Required)

Contains the parent organization of an organization.

Note: See How to Define the Organizational Hierarchy (see page 142) for more information about the %ORG_MEMBERSHIP% attribute.

%ORG_MEMBERSHIP_NAME%

(Required)

Contains the user-friendly name of the parent organization (see page 142) of an organization.

%ORG_NAME%

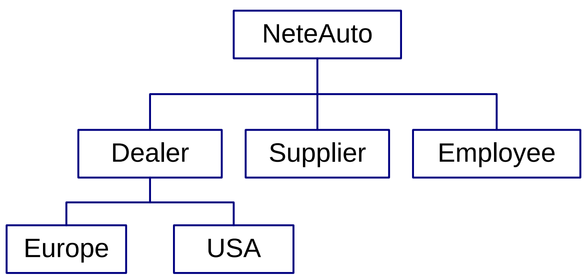
(Required)

Contains the name of the organization.

How to Define the Organizational Hierarchy

In CA Identity Manager, organizations have a hierarchical structure that includes a root organization and suborganizations. The suborganizations may also have suborganizations.

Each organization, except the root organization, has a parent organization. For example, in the following illustration, Dealer is the parent organization for the USA and Europe organizations:



The unique identifier of the parent organization is stored in an attribute in the profile of an organization. Using the information in this attribute, CA Identity Manager can construct the organization hierarchy.

To specify the attribute that stores the parent organization, use the %ORG_MEMBERSHIP% and %ORG_MEMBERSHIP_NAME% well-known attributes with the physical attribute storing the name of the parent organization in an attribute description as follows:

```
<ImsManagedObjectAttr physicalname="tblOrganizations.parentorg"
displayname="Organization" description="Parent Organization" valuetype="Number"
required="false" multivalued="false" wellknown="%ORG_MEMBERSHIP%" maxlength="0" />
```

How to Improve Directory Search Performance

To improve the performance of directory searches for users, organization, and groups, do the following tasks:

- Index the attributes that administrators can specify in search queries.

- Override the default directory timeout setting by specifying values for the timeout search parameters in a directory configuration file (directory.xml).
- Tune the user directory. See the documentation for the database that you are using.

Configure database-specific options in the ODBC data source. For more information, see the documentation for the data source.

How to Improve Performance for Large Searches

When CA Identity Manager manages a huge user store, searches that return many results can cause the system to run out of memory.

The following two settings determine how CA Identity Manager handles large searches:

- Maximum number of rows

  Specifies the maximum number of results that CA Identity Manager can return when searching a user directory. When the number of results exceeds the limit, an error is displayed.

- Page size

  Specifies the number of objects that can be returned in a single search. If the number of objects exceeds the page size, CA Identity Manager performs multiple searches.

  Note: If the user store does not support paging and a value for maxrows is specified, CA Identity Manager uses only the maxrows value to control search size.

You can configure maximum row limits and page size in the following places:

- User Store

  In most user stores and databases, you can configure search limits.

  Note: For more information, see the documentation for the user store or database that you are using.

- CA Identity Manager Directory

  You can configure the DirectorySearch element (see page 55) in the directory configuration file (directory.xml) that you use to create the CA Identity Manager Directory.

  By default, the value for maximum rows and page size is unlimited for existing directories. For new directories, the value for maximum rows is unlimited and the value for page size is 2000.

- Managed object definition

  To set maximum row limits and page sizes that apply to one type of object instead of an entire Directory, configure the *managed object definition* (see page 57) the directory.xml file that you use to create the CA Identity Manager Directory.

  Setting limits for a managed object type allows you to do adjustments that are based on business requirements. For example, most companies have more users than groups. Those companies can set limits for the user object searches only.

- Task Search screens

  You can control the number of search results that users see in search and list screens in the User Console. If the number of results exceeds the number of results per page that is defined for the task, users see links to additional pages of results.

  This setting does not affect the number of results that are returned by a search.

  Note: For information on setting page size in search and list screens, see the *Administration Guide*.

If maximum row limits and page sizes are defined in multiple places, the most specific setting applies. For example, managed object settings take precedence over directory level settings.

Chapter 5: CA Identity Manager Directories

A CA Identity Manager directory provides information about a user directory that CA Identity Manager manages. This information describes how objects such as users, groups, and organizations are stored in the user store and displayed in CA Identity Manager.

You create, view, export, update, and delete CA Identity Manager directories in the CA Identity Manager directory section of the Management Console.

Note: If CA Identity Manager uses a cluster of SiteMinder Policy Servers, stop all but one Policy Server before creating or updating CA Identity Manager directories.

This section contains the following topics:

Prerequisites to Creating a CA Identity Manager Directory

Before you create a CA Identity Manager Directory, you must do the following:

- Stop all but one CA Identity Manager node before you create or modify a CA Identity Manager Directory.

  Note: When you have a cluster of CA Identity Manager nodes, only one CA Identity Manager node can be enabled when you make changes in the Management Console.

- Stop all but one Policy Server before creating or updating CA Identity Manager Directories.

  Note: When you have a cluster of SiteMinder Policy Servers, only one SiteMinder policy server can be enabled when you make changes in the Management Console.

How to Create a Directory

In the Management Console, you create a CA Identity Manager Directory, which describes the structure and content of the user store, and the Provisioning Directory, which stores required information for the Provisioning Server. These directories are associated with CA Identity Manager environment.

You can use one of the following methods to create directories:

- Use the Directory Configuration Wizard

  Guides administrators through the process of creating a directory for their user store. This method helps reduce possible configuration errors.

  Note: Use the Directory Configuration Wizard to create new directories for the LDAP user stores only. To create a directory for a relational database, or to update an existing directory, import a directory.xml file directly.

- Use an XML Configuration File

  Allows administrators to select a fully configured XML file to create or modify the user store or Provisioning Server.

  Select this method if you are creating a directory for a relational database, or if you are updating an existing directory.

More Information:

Create a Directory with an XML Configuration File (see page 158)
Creating a Directory Using the Directory Configuration Wizard (see page 146)

Creating a Directory Using the Directory Configuration Wizard

The Directory Configuration Wizard walks administrators through the process of creating a directory for their user store and helps reduce configuration errors. Before you launch the wizard, you must first upload CA Identity Manager LDAP directory configuration template. These templates are preconfigured with well-known and required attributes. After entering connection details for your LDAP user store or Provisioning Directory, you can select the LDAP attributes, map well-known attributes, and can enter metadata for the attributes. When you are done mapping attributes, click Finish to create the directory.

Launch the Directory Configuration Wizard

The Directory Configuration Wizard lets an administrator select a CA Identity Manager template and modify that template for use in your environment.

Follow these steps:

1. From the Management Console, click Directories and select Create from Wizard.

   You are prompted to select a directory configuration file to configure the user store.

2. Click Browse to select the configuration file to configure the user store or Provisioning Server from the following default location and click Next.

   ```
   admin_tools\directoryTemplates\directory\
   ```

   Note: admin_tools specifies the directory where the Administrative Tools are installed and directory specifies the name of the LDAP vendor.

   The Administrative Tools are placed in the following default locations:

   - Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
   - UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

3. On the Connection Details screen, specify connection information for the LDAP directory or Provisioning Server, the directory search parameters, and failover connections information and click Next.

4. On the Configure Managed Object screen, specify the objects to configure and click Next. You can select from the following objects:

   - Configure User Managed Object
   - Configure Group Managed Object
   - Configure Organization Object
   - Show summary and deploy directory

     Note: Choose summary and deploy directory only when you have finished configuring the directory.

   a. On the Select Attribute screen, view and modify the structural and auxiliary classes as needed and click Next.

   b. On the Select Attributes: Mapping Well-Knowns screen, map the CA Identity Manager well-known aliases to selected LDAP attributes and click Next.

   c. (Optional) On the Describe User Attributes screen, view, and modify the attribute definitions and click Next. You can modify the display name and description.

   d. (Optional) On the User Attribute Details screen, define the metadata for each selected attribute to manage and click Next.

      The Managed Object Selection Screen appears.

      To configure Groups or Organizations, select the managed object and click Next to walk through Attributes screens for these objects.

5. Select Show summary and deploy directory from the list and click Next.

   The Confirmation Screen Appears.

6. View the directory details.

   If there are any errors, click the Back button to modify on the appropriate screens. Click Finish to apply the changes.

   CA Identity Manager validates the configuration and create the directory. You are then taken back to the Directories listing screen where you can view the new directory.

Select Directory Template Screen

Use this screen to select a directory XML file for LDAP to configure a user store or Provisioning Server.

Click the Browse button to select the configuration file to configure the user store or Provisioning Server from the following default location:

```
admin_tools\directoryTemplates\directory\
```

Note: admin_tools specifies the directory where the Administrative Tools are installed and directory specifies the name of the LDAP vendor.

The Administrative Tools are placed in the following default locations:

- Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

After you select the directory XML file, click Next to continue to the Connection Details Screen.

Connection Details Screen

Use this screen to enter the configuration credentials for your user store. You can also enter the directory search parameters, and add failover connections. After you enter connection information, click Next to select the objects to manage.

Note: The fields that appear on this screen depend on the type of user store, and whether you are creating the connection using the directory configuration wizard or directly importing an XML file.

The following fields are available in this screen:

Name

Specifies the name of the user directory to which you are connecting.

Description

Specifies a description of the user directory.

Host

Specifies the host name for the computer where the user store is located.

Port

Specifies the port for the computer where the user store is located.

User DN

Specifies the user domain name for accessing the LDAP user store.

JDBC Data Source JNDI Name

Specifies the name of an existing JDBC data source that CA Identity Manager uses to connect to the database.

Username

Specifies the username for accessing the Provisioning Server.

Note: For Provisioning Servers only.

Domain

Specifies the domain name for accessing the Provisioning Server.

Note: For Provisioning Servers only.

Password

Specifies the password for accessing the LDAP user store/Provisioning Server.

Confirm Password

Confirms the password for accessing the LDAP user store/Provisioning Server.

Secure Connection

When selected, forces a Secure Sockets Layer (SSL) connection to the LDAP user directory.

Search Root

Specifies the location in an LDAP directory that serves as the starting point for the directory—typically, an organization (o) or organizational unit (ou).

Note: For the LDAP user stores only.

Search Maximum Rows

Specifies the maximum number of results that CA Identity Manager can return when searching a user directory. When the number of results exceeds the limit, an error is displayed.

Setting maximum rows can override the settings in the LDAP directory that limit search results. When conflicting settings apply, the LDAP server uses the lowest setting.

Search Page Size

Specifies the number of objects that can be returned in a single search. If the number of objects exceeds the page size, CA Identity Manager performs multiple searches.

Note the following points when specifying Search Page Size:

- To use the Search Page Size option, the user store that CA Identity Manager manages must support paging. Some user store types may require additional configuration to support paging. For more information, see the *Configuration Guide*.
- If the user store does not support paging and a value for Search Maximum Rows is specified, CA Identity Manager uses only the Search Maximum Rows value to control search size.

Search Timeout

Specifies the maximum number of seconds that CA Identity Manager searches a directory before terminating the search.

Failover Host

Specifies the host name of the system where a redundant user store or an alternate Provisioning Server exists, in case the primary system is unavailable. If multiple servers are listed, CA Identity Manager attempts to connect to the systems in the same listed order.

Failover Port

Specifies the port of the system where a redundant user store or an alternate Provisioning Server exists, in case the primary system is unavailable. If multiple servers are listed, CA Identity Manager attempts to connect to the systems in the same listed order.

Add Button

Click to add additional failover host name and port numbers.

Configure Managed Objects Screen

Use this screen to select an object to configure.

The following list is the fields in this screen:

Configure User Managed Object

Describes how users are stored in the user store and how they are represented in CA Identity Manager.

Configure Group Managed Object

Describes how groups are stored in the user store and how they are represented in CA Identity Manager.

Configure Organization Managed Object

If the user store includes organizations, describes how organizations are stored and represented in CA Identity Manager.

Show Summary and Deploy Directory

Specifies that all managed objects have been defined and you want to deploy the directory. After you select the Show summary and the deploy directory, click Next and you are taken to a summary page.

Save Button

Click to save your xml file.

Back Button

Click to go back to the Connection Details screen to modify.

Next Button

Click to continue to the Select Attributes screen to select the user, group, or organization attributes to configure.

Select Attributes Screen

Use this screen to change or add structural and auxiliary classes for your User, Group, or Organization objects. This screen is preconfigured with values based on common directory schemas and best practices for the type of directory you are using. An administrator can change the structural class by selecting a new class from the drop-down menu. Selecting a class updates the table with attributes belonging to the new structural class.

An auxiliary class can be added by selecting one from the drop-down menu. Selecting an auxiliary class updates the table with attributes belonging to the new auxiliary class.

The following list is the fields appear in this screen:

Structural Class Name

Specifies the structural class of the attribute to configure.

Change Button

Click to change the structural class.

Auxiliary Class Name

Specifies the auxiliary class of the attribute to configure.

Add Button

Click to add an auxiliary class to configure.

Object Class

Specifies the container object class.

ID

Specifies the container ID.

Name

Specifies the container name.

Attributes Table

Specifies the physical name, object class, whether the attribute is multi-valued, and the data type of the selected attributes. Attributes in this table can be sorted by Selected, Object Class, Multi-Valued, and Data Type.

Back Button

Click to go back to the Configured Managed Objects screen.

Next

Click to continue to the Well-Known Mapping screen to map the required and optional well-known aliases.

Well-Known Mapping Screen

Use this screen to map CA Identity Manager well-known attributes to selected LDAP attributes. An administrator can add to the list of well-known attributes (if they are required for custom code) by typing a new well-known attribute into the text field and clicking the Add button. The screen refreshes so you can continue adding as many well-known attributes as needed.

The following list is the fields appear in this screen:

Required Well-Knowns

Specifies the well-known attributes for Users, Groups, or Organizations (if applicable) that are required to be mapped to LDAP attributes.

Optional Well-Knowns

Specifies the well-known attributes for Users, Groups, or Organizations (if applicable) that can be mapped optionally.

New Well-Known

Specifies a well-known attribute as referenced by custom code.

Add Button

Click to add a new well-known attribute to the Optional Well-Knowns table.

Back Button

Click to go back to the Select User Attributes screen to select more attributes. The mappings that you have already made are saved and available when you return to this screen.

Next Button

Click to continue to the Basic Object Attribute Definition screen to specify basic attribute definitions.

More information

Well-Known Attributes for an LDAP User Store (see page 74)
Group Well-Known Attributes (see page 78)
User Well-Known Attributes (see page 75)
Organization Well-Known Attributes (see page 79)

Basic Object Attribute Definition Screen

Use this screen to view and modify the commonly defined definitions: Display Name and Description.

The following list is the fields appear in this screen:

Managed Object Table

Specifies the display name, physical name, well-known name, and description of the managed object. Use the drop-down menu to change the description, if needed. Once you have made your changes, click Next to continue.

Back Button

Click to go back to the Well-Known Mapping screen to modify the details of the mappings.

Next Button

Click to continue to the Detailed Object Attribute Definition screen where you can specify additional attribute definitions.

Detailed Object Attribute Definition Screen

Use this screen to specify other attribute definitions. An administrator can define the metadata for each selected attribute by modifying the display name, managing the attribute in the User Console screens, the data type of the value, the maximum length, and the validation rule set. Once you have specified the attribute definitions, click Next to continue.

The fields in this screen are listed below:

Display Name

Specifies the unique name for the managed object attribute. This is the name that is displayed in the User Console.

Tags

Specifies the data classification tags for the managed object attribute value. The tags are all optional and all default to false except for searchable. The following tags can be selected:

Required

Indicates that the attribute is mandatory when creating objects.

Multiple Values

Indicates that the attribute appears as multi-valued.

Hidden

Indicates that the attribute is hidden.

System

Indicates that the attribute is a system attribute and is not added to the task screens.

Searchable

Indicates that the attribute is added to search filters. Defaults to true.

Sensitive Encrypt

Indicates that the attribute is sensitive and is displayed as a series of asterisks (*).

Hide in VST

Indicates that the attribute is hidden in the Event Details screen for View Submitted Tasks.

Do not copy

Indicates that the attribute must be ignored when an administrator creates a copy of an object.

Previously encrypted

Indicates that the attribute being accessed in the user store was previously encrypted and it requires decryption. The clear text value is saved to the user store when the object is saved.

Untagged encrypted

Indicates that the attribute was previously encrypted in the user store and does not have the encryption algorithm tagname at the beginning of the encrypted text.

Data Type

Specifies the data type of the value for the managed object attribute in the User Console. You can select from the following list:

- READONLY
- WRITEONCE
- READWRITE

Maximum Length

Specifies the maximum length of the value for the managed object attribute

Default: 0

Validation Rule Set

Specifies the validation rule sets to validate the value of the managed object attribute. You can select from the following list:

- User Validation
- Phone Format
- International Phone Format

Back Button

Click this button to go back to the Basic Object Attribute Definition Screen to modify.

Next Button

Click this button to continue to the Configure Managed Objects Screen. In this screen, you can select the next managed object to configure. Once you have configured your managed objects, select the Show summary and the deploy directory to view your directory information and deploy the directory.

More information

Managing Sensitive Attributes (see page 67)

Confirmation Screen

This screen shows a summary of the directory details.

The following list is the fields appear on this screen:

Connection Details

Specifies the connection details for the user directory.

User/Group/Organization Details

Specifies the changes that are made to the directory.xml.

Back Button

Click to modify any details in the wizard.

Save Button

Click to save your selections.

Finish Button

Click if all of the directory details are correct to exit the wizard.

The configuration is validated and the directory is created. You are then taken back to the Directories listing page where the new directory is listed. To edit or export the new directory, select it from the directory list.

Create a Directory with an XML Configuration File

You can create or update a CA Identity Manager Directory by importing a completed directory.xml file in the Management Console.

Note: If you are creating a directory using a directory.xml file instead of using the Directory Configuration Wizard, make sure that you have modified the default configuration template. For more information, see the *Configuration Guide*.

Follow these steps:

1. Open the Management Console by typing the following URL in a browser:

   http://*hostname*:*port*/iam/immanage

   hostname

   Defines the fully qualified domain name of the server where CA Identity Manager is installed.

   port

   Defines the application server port number.

2. Click Directories.

   The CA Identity Manager Directories window appears.

3. Click Create or Update from XML.

4. Type the path and filename of the directory configuration XML file for creating the CA Identity Manager Directory, or browse for the file. Click Next.

5. Supply values for the fields in this window as follows:

   Note: The fields that appear in this window depend on the user store type and the information you provided in the directory configuration file in Step 4. If you provided values for any of these fields in the directory configuration file, CA Identity Manager does not prompt you to supply these values again.

   Name

   Determines the name of the CA Identity Manager Directory that you are creating.

   Description

   (Optional) Describes the CA Identity Manager Directory.

   Connection Object Name

   Specifies the name of the user directory that the CA Identity Manager Directory describes. Enter *one* of the following details:

   - If CA Identity Manager does not integrate with SiteMinder, specify any meaningful name for the object that CA Identity Manager uses to connect to the user store.
   - If CA Identity Manager integrates with SiteMinder and you want to create a user directory connection object in SiteMinder, specify any meaningful name. CA Identity Manager creates the user directory connection object in SiteMinder with the name you specify.
   - If CA Identity Manager integrates with SiteMinder and you want to connect to an existing SiteMinder user directory, specify the name of the SiteMinder user directory connection object exactly as it appears in the Policy Server user interface.

   JDBC Data Source JNDI Name (for relational directories only)

   Specifies the name of an existing JDBC data source that CA Identity Manager uses to connect to the database.

   Host (for LDAP directories only)

   Specifies the host name or IP address of the system where the user directory is installed.

   For CA Directory user stores, use the full domain name of the host system. Do not use localhost.

   For Active Directory user stores, specify the domain name, not the IP address.

Port (for LDAP directories only)

Specifies the port number of the user directory.

Provisioning Domain

Provisioning Domain that CA Identity Manager manages.

Note: The Provisioning Domain name is case-sensitive.

Username/User DN

Specifies the user name for an account that can access the user store.

For Provisioning user stores, the user account you specify must have the Domain Administrator profile, or an equivalent set of privileges for the Provisioning Domain.

Password

Specifies the password for the user account that you specified in the Username (for Relational Databases) or User DN field (for LDAP directories).

Confirm Password

Enter the password that you typed in the Password field again for confirmation.

Secure Connection (for LDAP directories only)

Indicates whether CA Identity Manager uses a secure connection.

Be sure to select this option for Active Directory user stores.

Click Next.

6. Review the settings for the CA Identity Manager Directory. Click Finish to create the CA Identity Manager Directory with the current settings or click Previous to modify.

   Status information is displayed in the Directory Configuration Output window.

7. Click Continue to exit.

   CA Identity Manager creates the directory.

Enable Provisioning Server Access

Enabling access to Provisioning Server helps you connect to the Provisioning Sserver so that you can access the accounts available in the provisioning directory.

Follow these steps:

1. Open the Management Console by typing the following URL in a browser:

   `http://`*`hostname`*`:`*`port`*`/iam/immanage`

   hostname

   Defines the fully qualified host name of the system where the CA Identity Manager server is installed.

   port

   Defines the application server port number.

2. Click Directories.

3. Click Create from Wizard.

4. Type the path and filename of the directory XML file for configuring the Provisioning Directory. The XML file is stored in the directoryTemplates\ProvisioningServer in the Administrative Tools folder. The default location of that folder is:

   ■ Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools

   ■ UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

   Note: You can use this directory configuration file as installed with no modification.

5. Click Next.

6. Enter the values for the fields on this window as follows:

   Name

   Specifies a name for the Provisioning Directory that is associated with the Provisioning Server that you are configuring.

   – If CA Identity Manager does not integrate with SiteMinder, specify a meaningful name for the object that CA Identity Manager uses to connect to the user directory.

   – If CA Identity Manager integrates with SiteMinder, you have two choices:

     – If you want to create a user directory connection object in SiteMinder, specify any meaningful name. CA Identity Manager creates this object in SiteMinder with the name you specify.

     – If you want to connect to an existing SiteMinder user directory, specify the name of the SiteMinder user directory connection object exactly as it appears in the Policy Server user interface.

   Host

   Specifies the host name or IP address of the system where the user directory is installed.

Port

Specifies the port number of the user directory.

Domain

Specifies the name of the provisioning domain that CA Identity Manager manages.

Important! When creating a Provisioning Directory in the Management Console with the foreign language characters as the domain name, the Provisioning Directory creation fails.

The name must match the name of the provisioning domain that you specified during the installation.

Note: The domain name is case-sensitive.

Username

Specifies a user that can log in to the Provisioning Manager.

The user must have the Domain Administrator profile, or an equivalent set of privileges for the Provisioning Domain.

Password

Specifies the password for the global user that you specified in the Username field.

Confirm Password

Enter the password that you typed in the Password field again for the confirmation.

Secure Connection

Indicates whether CA Identity Manager uses a secure connection.

Be sure to select this option for Active Directory user stores.

Directory Search Parameters

maxrows

Defines the maximum number of results that CA Identity Manager can return when searching a user directory. This value overrides any limit that is set in the LDAP directory. When conflicting settings apply, the LDAP server uses the lowest setting.

Note: The maxrows parameter does not limit the number of results that are displayed on the CA Identity Manager task screen. To configure display settings, modify the list screen definition in the CA Identity Manager User Console.

timeout

Determines the maximum number of seconds that CA Identity Manager searches a directory before terminating the search.

Failover Connections

Specifies the hostname and port number of one or more optional systems that are alternate Provisioning Servers. If multiple servers are listed, CA Identity Manager attempts to connect to the systems in the listed order.

The alternate Provisioning Servers are used if the primary Provisioning Server fails. When the primary Provisioning Server becomes available again, the alternate Provisioning Server continues to be used. If you want to return to using the Provisioning Server, restart the alternate Provisioning Servers.

Click Next.

7. Select the objects for managing, such as Users or Groups.
8. After you have configured the objects as needed, click Show summary deploy directory and review the settings for the Provisioning Directory.
9. Click one of these actions:
   a. Click Back to modify.
   b. Click Save to save the directory information if you want to come back later to deploy.
   c. Click Finish to complete this procedure and start configuring an environment with provisioning (see page 182).

View a CA Identity Manager Directory

Perform the following procedure to view a CA Identity Manager Directory.

Follow these steps:

1. In the CA Identity Manager Management Console, click Directories.
2. Click the name of the CA Identity Manager directory to view. The Directory Properties window appears, showing the CA Identity Manager directory properties.

CA Identity Manager Directory Properties

The CA Identity Manager Directory properties are as follows:

Note: The properties that are displayed depend on the type of database or directory that is associated with the CA Identity Manager directory.

Name

Defines the unique name of the CA Identity Manager Directory.

Description

Provides a description for the CA Identity Manager Directory.

Type

Defines the type of directory provider.

Connection Object Name

Displays the name of the user directory that the CA Identity Manager Directory describes.

If CA Identity Manager integrates with SiteMinder, the connection object name matches the name of the SiteMinder user directory connection.

Root Organization (for user stores that include organizations)

Specifies the entry point into the user store.

For LDAP directories, the root organization is specified as a DN. For relational databases, the unique identifier for the root organization is displayed.

JDBC Data Source

Specifies the name the JDBC data source that CA Identity Manager uses to connect to the database.

URL

Provides the URL or IP address of the user store.

Username

Specifies the user name for an account that can access the user store.

Search Maximum Rows

Indicates the maximum number of returned rows as the result of a search.

Search Page Size

Specifies the number of objects that can be returned in a single search. If the number of objects exceeds the page size, CA Identity Manager performs multiple searches.

Note: The user store that CA Identity Manager manages must support paging. Some user store types may require additional configuration to support paging. For more information, see the *Configuration Guide*.

Supports Paging

Indicates that the directory supports paging.

Search Timeout (For LDAP directories only)

Specifies the maximum number of seconds that CA Identity Manager searches a user store before terminating the search.

Provisioning Domain (For Provisioning Server directories only)

Provisioning Domain that CA Identity Manager manages.

CA Identity Manager Directory Properties Window

General information about a CA Identity Manager directory is presented in the properties window for the directory that you select. The Directory Properties window is divided into the following sections:

Directory Properties

Displays basic properties for the CA Identity Manager directory including the associated Provisioning Domain, if Provisioning is enabled for the Environment.

Managed Objects **(see page 166)**

Provides descriptions of the type of user store objects that CA Identity Manager manages.

Validation Rule Sets **(see page 171)**

Lists validation rule sets that apply to the CA Identity Manager directory.

Environments

Lists the environments that are associated with the CA Identity Manager directory. A directory can be associated with multiple CA Identity Manager environments.

To view more information about a CA Identity Manager environment, click the name of the environment.

To modify properties in a CA Identity Manager directory, import a directory configuration file as described in Update a CA Identity Manager Directory (see page 173).

In addition to viewing properties, you can also perform the following actions:

Update Authentication

Allows administrators to change the directory that CA Identity Manager uses to authenticate Management Console administrators. Administrators can also add additional Management Console administrators in the existing authentication directory.

Note: The Update Authentication options apply only when native CA Identity Manager security protects the Management Console. For information about enabling native security or about using a different security method, see the *Configuration Guide*.

Export **(see page 172)**

Exports the directory definition as an XML file. After you export the directory settings, you can modify the XML file and then reimport it to update the directory. You can also import the XML file to another directory to configure the same settings for that directory.

Update **(see page 173)**

Allows administrators to add or change managed object definitions, such as the attributes of an object, set search parameters, and change directory properties.

How to View Managed Object Properties and Attributes

A managed object describes a type of entry in the user store, such as a user, group, or organization. The properties and attributes that apply to a managed object apply to all entries of that type. For example, a user profile consists of all the properties and attributes of the User managed object.

To view the details of a managed object, click the name of the object to open the Managed Object Properties window.

Managed Object Properties

The Managed Object Properties window describes the properties and attributes for a type of managed object.

The information about the Managed Object Properties window depends on the type of user store you are managing. A managed properties of an object are as follows:

Description

Provides a description of the managed object.

Type

Indicates the type of entry that the managed object represents. An object type can be one of the following types:

- User
- Group
- Organization

Object Class (for LDAP directories only)

Specifies the object classes for the managed object. A managed object may have multiple object classes.

Sort Order (for LDAP directories only)

Specifies the attribute that CA Identity Manager uses to sort search results in custom business logic. Sort Order does not affect the order of search results in the User Console.

For example, when you specify the cn attribute for the user object, CA Identity Manager sorts the results of a search for users alphabetically by the cn attribute.

Primary Table (for relational databases only)

Specifies the table that contains the unique identifier for the managed object.

Maximum Rows

Specifies the maximum number of results that CA Identity Manager can return when searching for objects of this type. When the number of results exceeds the limit, an error is displayed.

Setting maximum rows can override the settings in the LDAP directory that limit search results. When conflicting settings apply, the LDAP server uses the lowest setting.

Page Size

Specifies the number of objects that can be returned in a single search. If the number of objects exceeds the page size, CA Identity Manager performs multiple searches.

Note: The user store that CA Identity Manager manages must support paging. Some user store types may require additional configuration to support paging. For more information, see the *Configuration Guide*.

Container Properties (for LDAP Directories Only)

In an LDAP directory, the *container* groups contains objects of a specific type. When a container is specified, CA Identity Manager handles only entries in the container. For example, when you specify the container ou=People, CA Identity Manager handles users existing in the People container only.

Note: Users and groups that exist in the LDAP directory but not in the defined container may appear in the User Console. You may experience problems when managing those users and groups.

Containers group users and groups only. You cannot specify a container for organizations.

The properties of a container are as follows:

objectclass

Specifies the LDAP object class of the container where objects of a specific type are created. For example, the default value for the user container is "top,organizationalUnit," which indicates that users are created in LDAP organizational units (ou).

ID

Specifies the attribute that stores the container name, for example, ou. The attribute is paired with the Name value to form the relative DN of the container, as in the following example:

ou=People

Name

Specifies the container name.

Secondary Table Properties (for Relational Databases Only)

Secondary tables contain additional attributes for a managed object. For example, a secondary table named tblUserAddress may contain the street, city, state, and ZIP code attributes for the User managed object.

The following properties are displayed for secondary tables:

Table

Specifies the name of the table.

Reference

Describes the mapping between the primary table and the secondary table.

The reference is displayed using the following format:

primarytable.attribute=secondarytable.attribute

For example, tblUsers.id = tblUserAddress.userid indicates that the id attribute in the primary table, tblUsers, maps to the userid attribute in the tblUserAddress table.

Attribute Properties in the Managed Object Properties Window

The following properties are displayed for attributes in the Managed Object Properties window:

Display Name

The user-friendly name of the attribute. This name appears in the list of available attributes when you design a task window for a particular task in the User Console.

Physical Name

The name of the attribute in the user store.

Well-Known Name

Well-known names indicate attributes that have special meaning in CA Identity Manager, such as the attribute used to store user passwords.

Attribute Properties in the Attribute Properties Windows

You can see additional details about an attribute by clicking its name to open the Attribute Properties window.

The following attribute properties are displayed in the Attribute Properties window:

Description

Provides a description for the attribute.

Physical Name

Specifies the name of the attribute in the user store.

Object Class (for user, group, and organization attributes in LDAP directories only)

The LDAP auxiliary class for a user attribute, when the attribute is not part of the primary object class that is specified for the User object.

You can specify an auxiliary object class for the User and Group objects only.

Well-Known Name

Indicates attributes that have special meaning in CA Identity Manager, such as the attribute used to store user passwords.

Required

Indicates whether a value is required for the attribute, as follows:

- True indicates that the attribute must have a value.
- False indicates that a value is optional.

Read Only

Indicates the permission level for an attribute, as follows:

- True indicates that the attribute cannot be modified.
- False indicates that the attribute can be modified.

Hidden

Indicates whether an attribute can be displayed in a task window for a particular task.

Hidden attributes are often used in logical attribute schemes.

Note: For more information, see the *Programming Guide for Java*.

Supports Multiple Values

Indicates whether the attribute can have multiple values or not, as follows (for example, the attribute that is used to store the members of a group is multivalued):

- True indicates that the attribute can support multiple values.
- False indicates that the attribute can have only a single value.

Multiple Value Delimiter (for relational databases only)

The character that separates values when multiple values are stored in a single column.

System Attribute

Indicates whether the attribute is used only by CA Identity Manager or not, as follows:

- True indicates that the attribute is a system attribute. The attribute is not available to add to task windows.
- False indicates that the users can use this attribute. The attribute may appear on task windows.

Data Type

Specifies the data type of the attribute. The default value is String.

Maximum Length

Specifies the maximum length that an attribute value can have. If set to 0, there is no limit on the length of the value.

Validation Rule Set

Specifies the name of a validation rule set, when the attribute is associated with one.

Validation Rule Sets

A validation rule enforces requirements on data that a user types in a task window field. The requirements can enforce a data type or format or can ensure that the data is valid in the context of other data in the task window.

One or more validation rules are grouped in a validation rule set. A validation rule set is then associated with a profile attribute. For example, you can create a validation rule set that contains a Format Date validation rule, which enforces a date format of mm-dd-yyyy. You can then associate the validation rule set with the attribute that stores the start date of an employee.

Note: You create validation rules and rule sets in the directory configuration file or in the User Console.

The Managed Object Properties window displays a list of validation rule sets that apply to the CA Identity Manager directory. To view the details of a validation rule set, click the name of the rule set to open the Validation Rule Set Properties window.

Validation Rule Properties

The following information is displayed in the Validation Rule Properties window:

Name

Provides the name of the validation rule.

Description

Provides a description of the rule.

Class

Provides the name of the Java class that implements the validation rule.

This field does not appear unless the validation rule is defined in a Java class.

Filename

Provides the name of the file that contains the JavaScript implementation of the validation rule.

This field does not appear unless the validation rule is defined in a file.

Regular Expression

Provides the regular expression that implements the validation rule.

This field does not appear unless the validation rule is defined as a regular expression.

Validation Rule Set Properties

The following information is displayed in the Validation Rule Set Properties window:

Name

Specifies the name of the validation rule set.

Description

Provides a description for the validation rule set.

The Validation Rule Set Properties page also includes a list of validation rules in the set. You can click the name of the validation rule to open the Validation Rule Properties window.

How to Update Settings for a CA Identity Manager Directory

To view the current settings of a CA Identity Manager directory, export the directory settings and save it as an XML file.

After you export the directory settings, you can modify and again import the XML file to update the directory. You can also import the XML file to another directory to configure the same settings for that directory.

Export a CA Identity Manager Directory

Perform the following procedure to export a CA Identity Manager directory.

Follow these steps:

1. Click Directories.

   The list of CA Identity Manager directories appears.

2. Click the name of the directory to export.

   The Properties for the CA Identity Manager directory window appear.

3. At the bottom of the properties window, click Export.
4. When prompted, save the XML file.

Update a CA Identity Manager Directory

The purpose of updating a CA Identity Manager directory is to:

- Add or change managed object definitions, including the attributes of an object.
- Set search parameters
- Change directory properties

Note: CA Identity Manager does not delete object or attribute definitions.

The directory configuration file can contain only the changes that you want to make. You do not have to include properties or attributes that are already defined.

Note: When you have a cluster of CA Identity Manager nodes, only one CA Identity Manager node can be enabled when you make changes in the Management Console. Stop all but one CA Identity Manager node before you create or modify a CA Identity Manager directory.

Follow these steps:

1. Export the current CA Identity Manager directory settings to an XML file.
2. Modify the XML file to reflect your changes.
3. Click Directories.

   The list of CA Identity Manager directories appears.
4. Click the name of the directory to update.

   Properties for the CA Identity Manager directory appear.
5. At the bottom of the properties window, click Update.
6. Type the path and file name of the XML file for updating the CA Identity Manager directory, or browse for the file. Click Finish.

   Status information is displayed in the Directory Configuration Output field.
7. Click Continue.

Delete a CA Identity Manager Directory

Before you delete a CA Identity Manager Directory, delete any CA Identity Manager Environments that are associated with it.

Follow these steps:

1. In the Management Console, click Directories.

   The list of CA Identity Manager Directories appears.

2. Select the check box to the left of the directory (or directories) to delete.
3. Click Delete.

   A confirmation message appears.
4. Click OK to confirm the deletion.

Chapter 6: CA Identity Manager Environments

This section contains the following topics:

CA Identity Manager Environments

A CA Identity Manager environment is a view of a user store. In a CA Identity Manager environment, you can manage users, groups, organizations, tasks, and roles. You can also give users accounts in managed endpoints, such as email accounts or other applications.

Using the Management Console, you can do the following tasks:

- Create, modify, or delete a CA Identity Manager environment.
- Export and import a CA Identity Manager environment.
- Configure advanced settings
- Import roles and tasks
- Reset the System Manager account

Prerequisites to Creating a CA Identity Manager Environment

Before you begin, use the worksheet in the following table to collect the information that you need:

CA Identity Manager Environment Configuration Worksheet

Required Information	Value
A meaningful CA Identity Manager environment name that you choose. For example: MyEnvironment	
A base URL which CA Identity Manager uses to form the Redirect URL for the default password policy for the environment. For example: http://server.yourcompany.org	
An alias that is added to the URL for accessing protected tasks in the environment. For example: http://server.yourcompany.org/iam/im/alias	
An alias that is added to the URL for accessing public tasks, such as self-registration and forgotten password tasks. For example: http://server.yourcompany.org/iam/im/public_alias/index.jsp?task.tag=SelfRegistration **Note:** When your environment does not include public tasks, you do not need to specify a public alias.	
If you supplied a public alias, the name of an existing user who serves as the public user. CA Identity Manager uses credentials of the public user in place of user-supplied credentials when accessing public tasks.	
The name of a CA Identity Manager (see page 97)	
The name of the provisioning directory, when the CA Identity Manager environment supports provisioning.	

CA Identity Manager Environment Configuration Worksheet	
Required Information	**Value**
The unique identifier for an existing user who administers the CA Identity Manager environment. For example: myadmin	
The name of the SiteMinder agent or agent group that protects the CA Identity Manager environment if CA Identity Manager integrates with SiteMinder.	

Create a CA Identity Manager Environment

CA Identity Manager environments let you manage objects in a directory with a set of roles and tasks. Use the CA Identity Manager environment wizard to guide you through the steps to create a CA Identity Manager environment.

Note the following points before creating a CA Identity Manager environment:

- Assume that you are using an LDAP user store and you have configured a user container such as ou=People in the directory configuration file (directory.xml) for your CA Identity Manager directory. Verify that the users you select when you create the CA Identity Manager environment exist in that container. Selecting a user account that does not exist in the user container may cause failures.
- When you configure a CA Identity Manager environment to manage an LDAP user directory with a flat or flat user structure, the profile for the selected user must include the organization of the user. To help ensure that the profile of a user is configured correctly, add the name of the organization of the user to the physical attribute corresponding to the %ORG_MEMBERSHIP% well-known attribute in the directory.xml file (see page 81). For example, when the physical attribute description is mapped to the %ORG_MEMBERSHIP% well-known attribute in the directory.xml file and the user belongs to the Employees organization, the profile of the user must contain the attribute/value pair description=Employees.

Follow these steps:

1. If CA Identity Manager uses a cluster of Policy Servers, stop all but one Policy Server.
2. If you have a cluster of CA Identity Manager nodes, stop all but one CA Identity Manager node.
3. In the Management Console, click Environments.

4. Click New.

   The CA Identity Manager environment wizard opens.

5. Supply the following information:

   - **Environment name**

     Specifies a unique name for the environment

   - **Description**

     Describes the environment

   - **Protected alias**

     Specifies a unique name, such as employees. This alias is added to the URL for accessing protected tasks in the CA Identity Manager environment. For example, when the alias is employees, the URL for accessing the employee environment is http://my*server.mycompany.com*/iam/im/employees

     Note: The alias is case-sensitive and cannot contain spaces. We recommend using lowercase letters without punctuation or spaces when you specify the alias.

   - **Base URL**

     Specifies the URL for CA Identity Manager. The URL requires a hostname; it cannot include localhost. Also, do not include the alias, for example, http://*myserver.mycompany.com*/iam/im.

     If you are using a Web Agent, make sure that the Base URL is changed to reflect the URL of the Web Agent.

     Note: If you are using a Web Agent to protect CA Identity Manager resources, do not specify a port number in the Base URL field. If you are using a Web Agent and the Base URL contains a port number, the links to CA Identity Manager tasks doesnt not work properly.

     For more information about protecting CA Identity Manager resources, see the *Installation Guide* for your application server.

     Click Next.

6. Select a CA Identity Manager directory to associate with the environment you are creating, and click Next.

7. When the CA Identity Manager environment supports provisioning, select the appropriate provisioning server to use.

   Note: You are not prompted to select a provisioning server if you have selected a Provisioning directory as the CA Identity Manager directory.

8. Configure support for public tasks. Typically, these tasks are self-service tasks, such as self-registration or forgotten password tasks. Users do not need to log in to access public tasks.

   Note: To enable users to use self-service tasks, configure public task support.

   a. Specify a unique name that is added to the URL for accessing public tasks.

      Example: You would use the following URL to access the default self-registration task:

      http://*myserver.mycompany.com*/iam/im/*alias*/index.jsp?task.tag=SelfRegistration

      In this URL, *alias* is the unique name that you supply.

   b. Specify one of the following existing user accounts that serves as the public user account. CA Identity Manager uses this account to allow unknown users to access public tasks without having to supply credentials.

      – LDAP users enter the unique identifier or relative DN of the public user account. Make sure that this value is mapped to the %USER_ID% well-known (see page 74). For example, if the DN of the user DN is uid=Admin1, ou=People, ou=Employees, ou=NeteAuto, type Admin1.

      – Relational database users type the value that is mapped to the %USER_ID% well-known attribute in the directory configuration file, or the unique identifier for the user.

   Click Validate to view the full identifier of the user.

9. Select the tasks and roles to create for this environment. You can do the following tasks:

   - **Create default roles**

     Creates a set of default tasks and roles that are initially available in the environment. Administrators can use these tasks and roles as templates for creating new tasks and roles in the User Console.

   - **Create only the system manager role**

     Creates only the System Manager role and the tasks which are associated with it.

     The System Manager role is required to access the environment.

     A System Manager can create new tasks and roles in the User Console.

- **Import roles from the file**

  Imports a role definition file that you exported from another CA Identity Manager environment.

  Note: To use the CA Identity Manager environment, the role definitions file must include at least the System Manager role or a role that includes similar tasks.

  Select the Import roles from the file option button, and type the path and filename of the role definitions file or browse for the file to import.

10. Select Role Definitions files to create sets of default tasks for your environment, and click Next.

    Role Definitions files are XML files that define a set of tasks and roles that are required to support specific features. For example, if you want to manage Active Directory and UNIX NIS endpoints, select those Role Definitions files.

    Note: This step is optional. If you do not want to create additional default tasks to support new functionality, skip this screen.

11. Define a user to serve as the System Manager for this environment as follows:

    a. In the System Manager field, type the value that is mapped to the %USER_ID% well-known attribute in the directory configuration file, or specify one of the following user accounts:

       – LDAP users enter the unique identifier or relative DN of the user. For example, if the DN of the user DN is uid=Admin1, ou=People, ou=Employees, ou=NeteAuto, type Admin1.

       – Relational database users type the unique identifier for the user.

    b. Click Add.

       CA Identity Manager adds the complete identifier of the user to the list of users.

    c. Click Next.

    Note the following points when specifying the System Manager:

    - The System Manager must *not* be the same user as the administrator of the user store.

    - You can specify multiple System Managers for the Environment. However, you can only specify the initial System Manager in the Management Console. To specify additional System Managers, assign the System Manager role to the appropriate users in the User Console.

12. In the Inbound Administrator field, specify a CA Identity Manager administrator account that can execute admin tasks that are mapped to inbound mappings.

    The user must be able to execute all those tasks on any user. The Provisioning Synchronization Manager role contains the provisioning tasks that are included in the default inbound mappings.

13. Enter a password for the keystore, the database of keys that encrypt and decrypt data.

    Defining this password is a prerequisite to defining dynamic keys. You can modify the password after creating the environment using the System, Secret Keys task.

    A page summarizing the settings for the environment appears.

14. Review the settings for the environment. Click Previous to modify or click Finish to create the CA Identity Manager environment with the current settings.

    The Environment Configuration Output screen displays the progress of the environment creation.

15. Click Continue to exit the CA Identity Manager environment wizard.
16. Start the Environment.

    Clicking the environment name, then click Start.

17. If you stopped any Policy Servers in Step 1, restart them now.

How to Access a CA Identity Manager Environment

After you have created a CA Identity Manager environment, you can access it by typing a URL in a browser.

Note: Enable Javascript in the browser that you use to access the Management Console.

The format of the URL depends on how you configured the environment and the type of task that you want to access.

- To access protected tasks from the User Console, use the following URL:

  `http://hostname/iam/im/alias`

  hostname

  Defines the fully qualified domain name of the server where CA Identity Manager is installed—for example, myserver.mycompany.com

  alias

  Defines the alias of the environment alias, for example, employees.

  Log in to the CA Identity Manager Environment with a privileged administrator account, such as the System Manager account that you created for the CA Identity Manager Environment.

  Note: All CA Identity Manager tasks are protected unless you configure public tasks.

- To access public tasks, which do not require users to provide credentials, use a URL with the following format:

  http://*hostname*/iam/im/*alias*/index.jsp?task.tag=*tasktag*

  hostname

  Defines the fully qualified domain name of the server where CA Identity Manager is installed, for example, myserver.mycompany.com.

  alias

  Defines the alias for public tasks, for example, self-service.

  task_tag

  Defines the tag for the task to invoke.

  You specify the task tag when you configure a task in the User Console.

  The task tags for the default self-registration and forgotten password reset tasks are SelfRegistration and ForgottenPasswordReset.

  Note: For more information, see the *Administration Guide*.

How to Configure an Environment for Provisioning

You can configure an environment for provisioning after you have enabled access to the Provisioning Server (see page 160).

Then, you create a special CA Identity Manager user, called the Inbound Administrator, create a connection to the Provisioning Server, and configure inbound synchronization in Provisioning Manager.

Note: Whenever you modify the provisioning properties for an environment, make sure that you restart the application server for the changes to take effect.

Configure the Inbound Administrator

For the inbound synchronization to work, create a special CA Identity Manager user called *inbound administrator*. An inbound administrator is an account that CA Identity Manager uses during synchronization. In the previous releases of CA Identity Manager, the inbound administrator was called the *corporate use*r. No user logs in to this user account; instead, CA Identity Manager uses it internally. Use this procedure for configuring user account with an inbound administrator privileges.

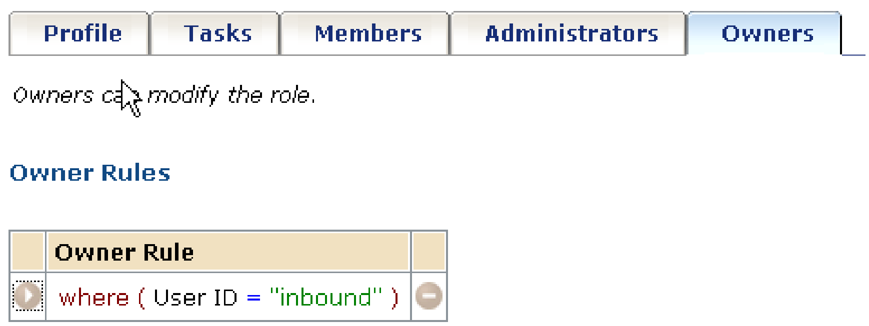
Follow these steps:

1. Log in to the environment as the user with the System Manager role.
2. Create a user. You can name the user **inbound** as a reminder of its purpose.

3. Choose Admin Roles, Modify Admin Role, and select a role that contains the tasks you use for the synchronization.
   - Provisioning Create User
   - Provisioning Enable/Disable User
   - Provisioning Modify User

   Note: If you have not modified the default synchronization tasks, use the Provisioning Synchronization Manager role.

4. On the Members tab, add a member policy that includes:
   - A member rule that the new user meets.
   - A scope rule providing access to all users who are affected by provisioning directory changes that trigger the inbound synchronization. The following screenshots shows the Owners tab where you create the owner rule:



5. In the Management Console:

   a. Select the Environment.

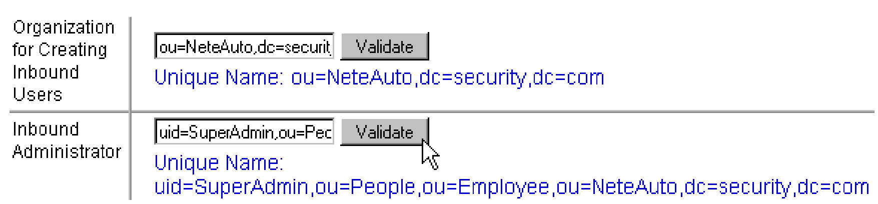
   b. Select Advanced Settings, Provisioning.

   c. Complete the Organization for the Creating Inbound Users field if the CA Identity Manager directory includes an organization.

   This organization is where users are created when the inbound synchronization occurs. For example, when a user is added to the provisioning directory, CA Identity Manager adds the user to this organization.

   d. Complete the Inbound Administrator field with the User ID of the user that you created in Step 2.

   e. Click Validate to confirm the user ID is accepted. As shown in the following example where the complete user ID appears below the user ID entered.



f. Modify other fields on the screen.

When you modify, be sure that you understand how the fields interact. For details on each field, click the Help link on the screen.

Connect an Environment to the Provisioning Server

As a system administrator, assign the provisioning server to an environment for synchronizing the accounts in both the directories.

Follow these steps:

1. In the Management Console, click Environments.
2. Click the name of the environment that you want to associate with the Provisioning Server.
3. Click the right arrow icon in the Provisioning Server field.

   The Provisioning Properties screen opens.
4. Select the Provisioning Server.
5. Click Save at the bottom of the screen.
6. Configure Synchronization in the Provisioning Manager (see page 184).

Configure Synchronization in the Provisioning Manager

Inbound synchronization keeps CA Identity Manager up to date with changes that occur in the provisioning directory. Changes include those made using Provisioning Manager and changes in endpoints for which the Provisioning Server has a connector.

Note: Each Provisioning Server supports a single environment. However, you can configure backup environments on different systems in a cluster in case the current environment is unavailable.

Follow these steps:

1. In the CA Identity Manager server, choose Start, CA Identity Manager, Provisioning Manager.
2. Click System, CA Identity Manager Setup.
3. Complete the Host Name field with the name of the system where the CA Identity Manager Server is installed.

4. Complete the Port field with the application server port number.
5. Complete the Environment name field with the alias for the environment.
6. Select Secured Connection if you want the HTTPS protocol to communicate with the CA Identity Manager server instead of using HTTP and encrypting the individual notifications.
7. Click Add.
8. Repeat steps 3-6 for each of the backup version of the environment.

   If the application server for the current environment is unavailable, the CA Identity Manager failovers to a backup environment. You can reorder the current and backup environments to set the failover order.
9. If it is the first environment, fill in the Shared Secret fields using the password that was entered during the CA Identity Manager installation for the user for embedded components.

   Note: These fields do not apply if FIPS is enabled in this installation.
10. Set the Log Level as follows:
    - No Log--No information is written to the log file.
    - Error--Only error messages are logged.
    - Info--Error and information messages are logged (default).
    - Warning--Error, warning, and information messages are logged.
    - Debug--All information is logged.
11. Restart the application server before you log in to the environment.

Note: For a log of inbound synchronization operations and any problems that are encountered during synchronization, see the following file:

PSHOME`\logs\etanotify<date>.log`

Import Custom Provisioning Roles

When you create the environment, you have the choice to use the default roles or a custom role definition file you create. If you import custom roles definitions, *also* import the Provisioning Only role definitions. After you create the environment, import the role definitions from the ProvisioningOnly-RoleDefinitions.xml file, which is in one of these folders:

admin_tools/ProvisioningOnlyRoleDefinitions/Organization
admin_tools/ProvisioningOnlyRoleDefinitions/NoOrganization

The default location for *admin_tools* is:

- **Windows**: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX**: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

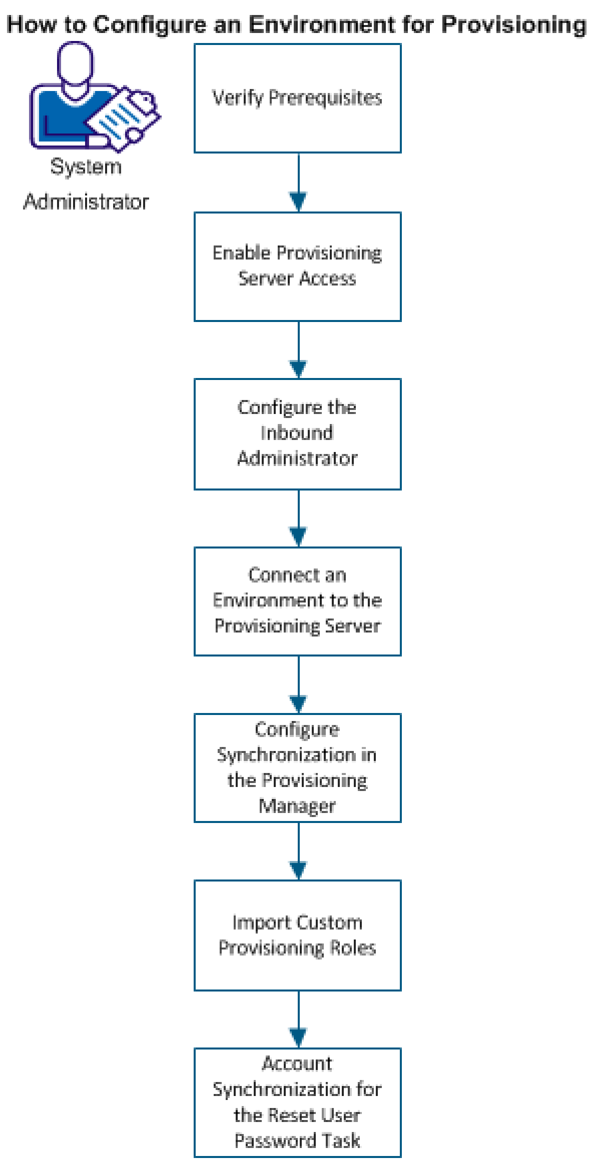
Account Synchronization for the Reset User Password Task

Before you enable provisioning for an environment, the account synchronization setting for the Reset User Password task is set to On Task Completion. However, when you import the ProvisioningOnly-RoleDefinitions.xml configuration file which creates the roles and tasks for user provisioning, account synchronization is disabled.

To use the Reset User Password to trigger the account synchronization, change account synchronization for this task back to On Task Completion.

How to Create and Deploy Connectors Using Connector Xpress

You can configure provisioning for an environment to provide accounts in other systems to users managed by CA Identity Manager. Accounts provide users with access to additional resources, such as an email account. You provide these additional accounts by assigning provisioning roles, which you create through CA Identity Manager.



As an administrator, complete the following steps:

1. Verify Prerequisites (see page 188)
2. Enable Provisioning Server Access (see page 160)
3. Configure the Inbound Administrator (see page 182)

4. Connect an Environment to the Provisioning Server (see page 184)
5. Configure Synchronization in the Provisioning Manager (see page 184)
6. Import Custom Provisioning Roles (see page 186)
7. Account Synchronization for the Reset User Password Task (see page 186)

Verify Prerequisites

Before you configure the environment for provisioning, make sure that the Provisioning Directory is installed on CA Directory. For more information, see the *Installation Guide*.

Enable Provisioning Server Access

Enabling access to Provisioning Server helps you connect to the Provisioning Sserver so that you can access the accounts available in the provisioning directory.

Follow these steps:

1. Open the Management Console by typing the following URL in a browser:

   `http://`*`hostname`*`:`*`port`*`/iam/immanage`

   hostname

   Defines the fully qualified host name of the system where the CA Identity Manager server is installed.

   port

   Defines the application server port number.

2. Click Directories.

3. Click Create from Wizard.

4. Type the path and filename of the directory XML file for configuring the Provisioning Directory. The XML file is stored in the directoryTemplates\ProvisioningServer in the Administrative Tools folder. The default location of that folder is:

   - Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
   - UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

   Note: You can use this directory configuration file as installed with no modification.

5. Click Next.

6. Enter the values for the fields on this window as follows:

   Name

   Specifies a name for the Provisioning Directory that is associated with the Provisioning Server that you are configuring.

   - If CA Identity Manager does not integrate with SiteMinder, specify a meaningful name for the object that CA Identity Manager uses to connect to the user directory.
   - If CA Identity Manager integrates with SiteMinder, you have two choices:
     - If you want to create a user directory connection object in SiteMinder, specify any meaningful name. CA Identity Manager creates this object in SiteMinder with the name you specify.
     - If you want to connect to an existing SiteMinder user directory, specify the name of the SiteMinder user directory connection object exactly as it appears in the Policy Server user interface.

   Host

   Specifies the host name or IP address of the system where the user directory is installed.

   Port

   Specifies the port number of the user directory.

   Domain

   Specifies the name of the provisioning domain that CA Identity Manager manages.

   Important! When creating a Provisioning Directory in the Management Console with the foreign language characters as the domain name, the Provisioning Directory creation fails.

   The name must match the name of the provisioning domain that you specified during the installation.

   Note: The domain name is case-sensitive.

   Username

   Specifies a user that can log in to the Provisioning Manager.

   The user must have the Domain Administrator profile, or an equivalent set of privileges for the Provisioning Domain.

Password

Specifies the password for the global user that you specified in the Username field.

Confirm Password

Enter the password that you typed in the Password field again for the confirmation.

Secure Connection

Indicates whether CA Identity Manager uses a secure connection.

Be sure to select this option for Active Directory user stores.

Directory Search Parameters

maxrows

Defines the maximum number of results that CA Identity Manager can return when searching a user directory. This value overrides any limit that is set in the LDAP directory. When conflicting settings apply, the LDAP server uses the lowest setting.

Note: The maxrows parameter does not limit the number of results that are displayed on the CA Identity Manager task screen. To configure display settings, modify the list screen definition in the CA Identity Manager User Console.

timeout

Determines the maximum number of seconds that CA Identity Manager searches a directory before terminating the search.

Failover Connections

Specifies the hostname and port number of one or more optional systems that are alternate Provisioning Servers. If multiple servers are listed, CA Identity Manager attempts to connect to the systems in the listed order.

The alternate Provisioning Servers are used if the primary Provisioning Server fails. When the primary Provisioning Server becomes available again, the alternate Provisioning Server continues to be used. If you want to return to using the Provisioning Server, restart the alternate Provisioning Servers.

Click Next.

7. Select the objects for managing, such as Users or Groups.
8. After you have configured the objects as needed, click Show summary deploy directory and review the settings for the Provisioning Directory.
9. Click one of these actions:
   a. Click Back to modify.
   b. Click Save to save the directory information if you want to come back later to deploy.
   c. Click Finish to complete this procedure and start configuring an environment with provisioning (see page 182).

Configure the Inbound Administrator

For the inbound synchronization to work, create a special CA Identity Manager user called *inbound administrator*. An inbound administrator is an account that CA Identity Manager uses during synchronization. In the previous releases of CA Identity Manager, the inbound administrator was called the *corporate use*r. No user logs in to this user account; instead, CA Identity Manager uses it internally. Use this procedure for configuring user account with an inbound administrator privileges.

Follow these steps:

1. Log in to the environment as the user with the System Manager role.
2. Create a user. You can name the user **inbound** as a reminder of its purpose.

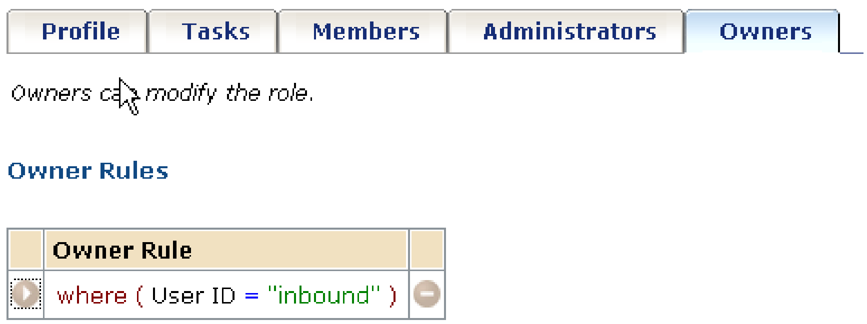
3. Choose Admin Roles, Modify Admin Role, and select a role that contains the tasks you use for the synchronization.

   - Provisioning Create User
   - Provisioning Enable/Disable User
   - Provisioning Modify User

   Note: If you have not modified the default synchronization tasks, use the Provisioning Synchronization Manager role.

4. On the Members tab, add a member policy that includes:

   - A member rule that the new user meets.
   - A scope rule providing access to all users who are affected by provisioning directory changes that trigger the inbound synchronization. The following screenshots shows the Owners tab where you create the owner rule:



5. In the Management Console:

   a. Select the Environment.

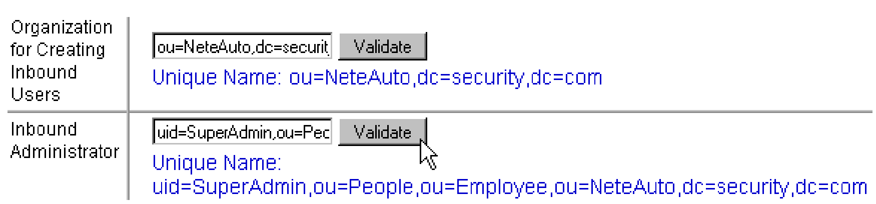
   b. Select Advanced Settings, Provisioning.

   c. Complete the Organization for the Creating Inbound Users field if the CA Identity Manager directory includes an organization.

   This organization is where users are created when the inbound synchronization occurs. For example, when a user is added to the provisioning directory, CA Identity Manager adds the user to this organization.

   d. Complete the Inbound Administrator field with the User ID of the user that you created in Step 2.

   e. Click Validate to confirm the user ID is accepted. As shown in the following example where the complete user ID appears below the user ID entered.



f. Modify other fields on the screen.

   When you modify, be sure that you understand how the fields interact. For details on each field, click the Help link on the screen.

Connect an Environment to the Provisioning Server

As a system administrator, assign the provisioning server to an environment for synchronizing the accounts in both the directories.

Follow these steps:

1. In the Management Console, click Environments.
2. Click the name of the environment that you want to associate with the Provisioning Server.
3. Click the right arrow icon in the Provisioning Server field.

   The Provisioning Properties screen opens.
4. Select the Provisioning Server.
5. Click Save at the bottom of the screen.
6. Configure Synchronization in the Provisioning Manager (see page 184).

Configure Synchronization in the Provisioning Manager

Inbound synchronization keeps CA Identity Manager up to date with changes that occur in the provisioning directory. Changes include those made using Provisioning Manager and changes in endpoints for which the Provisioning Server has a connector.

Note: Each Provisioning Server supports a single environment. However, you can configure backup environments on different systems in a cluster in case the current environment is unavailable.

Follow these steps:

1. In the CA Identity Manager server, choose Start, CA Identity Manager, Provisioning Manager.
2. Click System, CA Identity Manager Setup.
3. Complete the Host Name field with the name of the system where the CA Identity Manager Server is installed.

4. Complete the Port field with the application server port number.
5. Complete the Environment name field with the alias for the environment.
6. Select Secured Connection if you want the HTTPS protocol to communicate with the CA Identity Manager server instead of using HTTP and encrypting the individual notifications.
7. Click Add.
8. Repeat steps 3-6 for each of the backup version of the environment.

   If the application server for the current environment is unavailable, the CA Identity Manager failovers to a backup environment. You can reorder the current and backup environments to set the failover order.
9. If it is the first environment, fill in the Shared Secret fields using the password that was entered during the CA Identity Manager installation for the user for embedded components.

   Note: These fields do not apply if FIPS is enabled in this installation.
10. Set the Log Level as follows:
    - No Log--No information is written to the log file.
    - Error--Only error messages are logged.
    - Info--Error and information messages are logged (default).
    - Warning--Error, warning, and information messages are logged.
    - Debug--All information is logged.
11. Restart the application server before you log in to the environment.

Note: For a log of inbound synchronization operations and any problems that are encountered during synchronization, see the following file:

PSHOME\logs\etanotify<date>.log

Import Custom Provisioning Roles

When you create the environment, you have the choice to use the default roles or a custom role definition file you create. If you import custom roles definitions, *also* import the Provisioning Only role definitions. After you create the environment, import the role definitions from the ProvisioningOnly-RoleDefinitions.xml file, which is in one of these folders:

admin_tools/ProvisioningOnlyRoleDefinitions/Organization
admin_tools/ProvisioningOnlyRoleDefinitions/NoOrganization

The default location for *admin_tools* is:

- **Windows**: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX**: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

Account Synchronization for the Reset User Password Task

Before you enable provisioning for an environment, the account synchronization setting for the Reset User Password task is set to On Task Completion. However, when you import the ProvisioningOnly-RoleDefinitions.xml configuration file which creates the roles and tasks for user provisioning, account synchronization is disabled.

To use the Reset User Password to trigger the account synchronization, change account synchronization for this task back to On Task Completion.

Manage Environments

This section describes how to manage an environment.

Modify CA Identity Manager Environment Properties

The CA Identity Manager Environment Properties screen in the Management Console lets you do the following tasks:

- View the current settings for the environment.
- Modify the description, base URL, and protected and public aliases.
- Import an existing CA Identity Manager Environment after an upgrade.

  Note: For more information on importing existing CA Identity Manager Environments, see the upgrade section of the *Installation Guide*.
- Start and stop the Environment
- Access pages for configuring the following tasks:
  - **Advanced Settings**

    Configures advanced features, including features that are built using the CA Identity Manager APIs.
  - **Role and Task Settings**

    Import a role definition file that you exported from another CA Identity Manager Environment.
  - **System Manager**

    Assigns system manager roles.

Follow these steps:

1. If CA Identity Manager uses a cluster of SiteMinder Policy Servers, stop all but one Policy Server.
2. If you have a cluster of CA Identity Manager nodes, stop all but one CA Identity Manager node.
3. Click Environments.

   The CA Identity Manager Environments screen appears with a list of CA Identity Manager Environments.
4. Click the name of the CA Identity Manager Environment to modify.

   The CA Identity Manager Properties screen appears and displays the following properties:

   OID

   Defines a unique identifier for the Environment. CA Identity Manager generates this identifier when you create a CA Identity Manager environment.

   You use the OID when you configure task removal from a task persistence database. See the *Installation Guide*.

   Name

   Specifies the unique name of the CA Identity Manager Environment.

   Description

   Provides a description of the CA Identity Manager Environment.

   CA Identity Manager Directory

   Specifies the CA Identity Manager directory with which the Environment is associated.

   Enable Verbose Log Output

   Controls how much information CA Identity Manager records and displays in the Environment log when you import an Environment. The Environment log is displayed in the status window in the Management Console when you import an Environment or other object definitions from a file.

   Note: Selecting this check box can significantly impact the performance.

   The verbose log includes validation and deployment messages for each object (task, screen, role and policy) and its attributes in the Environment.

   To see the verbose log, select this check box and save the Environment properties. When you import roles or other settings from a file, the additional information appears in the log.

Provisioning Server

Specifies the Provisioning directory that is used as the provisioning user store.

Click the right arrow button to configure the provisioning directory in the Provisioning Properties page.

Version

Defines the version number of CA Identity Manager.

Base URL

Specifies the portion of the CA Identity Manager URL that does not include the protected or public alias for the environment.

CA Identity Manager uses the base URL to form the Redirect URL to point to the Password Services task in the default password policy for the environment.

Protected Alias

Defines the base URL name for accessing protected tasks in the User Console for a CA Identity Manager environment.

Public Alias

Defines the base URL name for accessing public tasks, such as the self-registration and forgotten password tasks.

Public User

Defines the user account that CA Identity Manager uses in place of user-supplied credentials to access public tasks.

Job Timeout

Determines the amount of time that CA Identity Manager waits after a task is submitted before displaying a status message.

This value is set in the User Console page in Advanced Settings.

Status

Stops or restarts the CA Identity Manager Environment.

Migrate Task Persistence Data from CA Identity Manager 8.1

Migrates data from a CA Identity Manager 8.1 task persistence database to CA Identity Manager 12.6.4 task persistence database.

For more information, see the *Installation Guide*.

Note: The Migrate Task Persistence Data from CA Identity Manager 8.1 button is only visible in environments that were created in previous versions of CA Identity Manager and migrated to CA Identity Manager 12.6.4.

5. Modify the description, base URL, or protected or public alias, as needed.

6. If you modified any Environment properties, restart the CA Identity Manager Environment.
7. If you stopped any Policy Servers in Step 1, restart them now.

Environment Settings

Environment-specific information is stored in three environment settings files:

- *alias*_environment_roles.xml
- *alias*_environment_settings.xml
- *alias*_environment.xml

Note: *alias* refers to the alias for the environment. You specify the alias when you create the environment.

You generate a ZIP file containing these files, which reflect the current configuration, when you export the environment settings.

After you have exported the environment settings, import the settings to accomplish one of the following tasks:

- Manage multiple environments with similar settings. In this case, you create one environment with the settings you need, import those settings to other environments, and then customize the settings in each environment, as needed.
- Migrate an environment from a development system to a production system.
- Update an existing environment after upgrading to a new version of CA Identity Manager.

Export a CA Identity Manager Environment

To deploy a CA Identity Manager environment on a production system, you export the environment from a development or staging system and import that environment to the production system.

Note: When you import a previously exported environment, CA Identity Manager displays a log in a status window in the Management Console. To see validation and deployment information for each managed object and its attributes in this log, select the Enable Verbose Log Output field on the Environment Properties page *before* you export the environment. Make sure that selecting the Enable Verbose Log Output field can cause significant performance issues during the import.

Follow these steps:

1. Click Environments in the Management Console.

   The CA Identity Manager environments screen appears with a list of CA Identity Manager environments.

2. Select the environment that you want to export.
3. Click the Export button.

   A File Download screen appears.

4. Save the ZIP file to a location that is accessible to the production system.
5. Click Finish.

The environment information is exported to a ZIP file that you can import into another environment.

Import a CA Identity Manager Environment

You can import CA Identity Manager environment settings to accomplish one of the following tasks:

- Manage multiple environments with similar settings. In this case, you create one environment with the settings you need, import those settings to other environments, and then customize the settings in each environment, as needed.
- Migrate an environment from a development system to a production system.
- Update an existing environment after upgrading to a new version of CA Identity Manager.

Follow these steps:

1. Click Environments in the Management Console.

   The CA Identity Manager environments screen appears with a list of CA Identity Manager Environments.

2. Click the Import button.

   The Import Environment screen appears.

3. Browse for the ZIP file required to import an environment.
4. Click Finish.

The environment is imported into CA Identity Manager.

Restart a CA Identity Manager Environment

Follow these steps:

1. Click Environments in the Management Console.

   The CA Identity Manager environments screen appears with a list of CA Identity Manager environments.

2. Click the name of the CA Identity Manager environment to start.

   The CA Identity Manager Environment Properties screen appears.

3. Select one of the following options:

   Restart Environment

   Stops and starts an Environment.

   Stop

   Stops an Environment that is currently running.

   Start

   Starts an Environment that is not currently running.

Delete a CA Identity Manager Environment

Use this procedure to remove a CA Identity Manager Environment.

Note: If CA Identity Manager integrates with SiteMinder for advanced authentication, CA Identity Manager also deletes the SiteMinder policy domain protecting the environment and the default authentication schemes that are created for the environment.

Follow these steps:

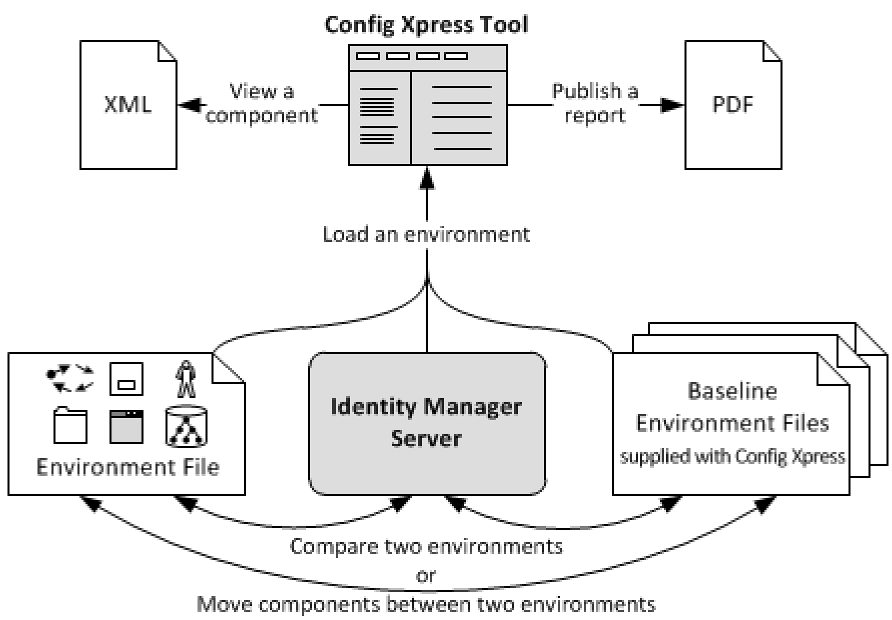
1. In the Environments screen, select the check box for the CA Identity Manager Environments to delete.
2. Click Delete.

   CA Identity Manager displays a confirmation message.
3. Click OK to confirm the deletion.

Manage Configuration

Config Xpress is a tool that is included with CA Identity Manager. You can use this tool to analyze and work with the configurations of your CA Identity Manager environments.

Most importantly, the tool lets you move components between environments. Config Xpress automatically detects any other required components, and prompts you move them too. This assistance can save you work and reduce the risk of problems.



Follow these steps:

1. Set up Config Xpress (see page 202).
2. Before you can use the tool, load a CA Identity Manager environment (see page 203) into Config Xpress for analysis.
3. Use Config Xpress to do these tasks with the loaded environment:
   - Move components between environments (see page 205).
   - Publish a PDF report of the system components (see page 206).
   - Display the XML configuration for a particular component (see page 207).

Set Up Config Xpress

The installation files for Config Xpress are included on the installation drive, but the tool is not installed.

Config Xpress has the following software requirements:

- CA Identity Manager r12.0 and later
- Windows operating system
- Adobe Air Runtime
- PDF reader to view reports

Follow these steps:

1. Download Adobe Air Runtime from http://get.adobe.com/air, then install it.
2. Ensure that the Administration Tools are installed.
3. Look in the following location for the installation file for Config Xpress:

   ```
   C:\Program Files\CA\Identity Manager\IAM Suite\Identity
   Manager\tools\ConfigXpress
   ```
4. Run Config Xpress.air to install Config Xpress.
5. When the installation is complete, Config Xpress starts.

Load an Environment into Config Xpress

Before you can use Config Xpress, load one or more environments into the tool. This task lets you work with the environment in Config Xpress.

You can load an environment into Config Xpress directly from a live CA Identity Manager server, or you can load it from an environment file. If you use one of the baseline environment files that are installed with Config Xpress, you can compare your environment to the out-of-the-box configuration.

The process of loading an environment can take a few minutes.

Follow these steps:

1. Open Config Xpress.
2. To load a **live environment** directly from a CA Identity Manager server:
    a. Click the Server (Network) tab.
    b. Enter the name and port of the CA Identity Manager server. For example:

       `servername.ca.com:8080`

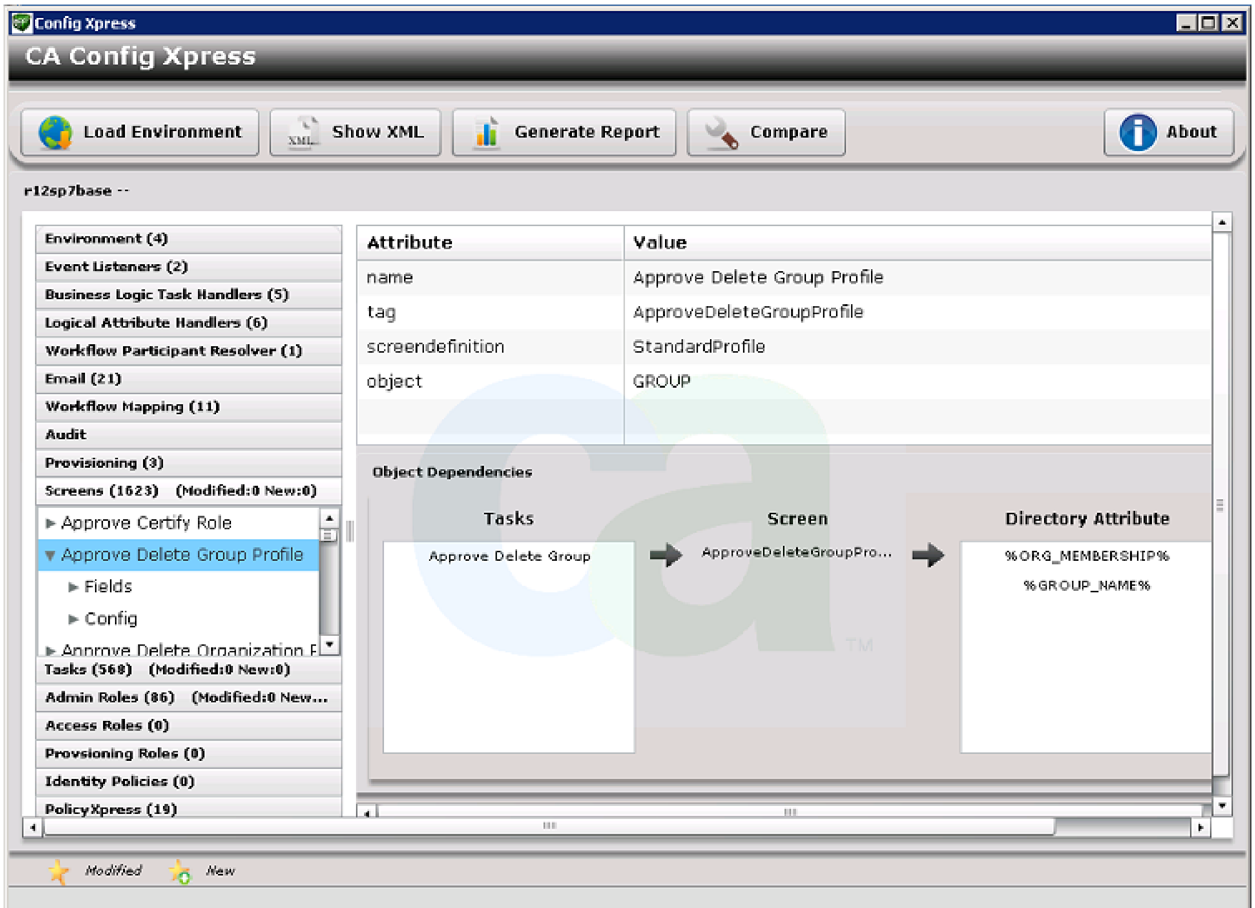
    c. Select Use HTTPS if your server is set up to allow HTTPS only.
    d. Select 12.5 SP7 if the version of the server is more recent than r12.5 SP6.
    e. Click Connect.
    f. Choose an environment from the *Choose Environment to load* list, then click Load.
3. To load an **environment file** that was exported from your CA Identity Manager environment:
    a. Export a CA Identity Manager environment.
    b. In Config Xpress, click the File System tab.
    c. Select the version, then browse to the environment file, then click Load.
4. To load a **baseline environment file** that was installed with Config Xpress:
    a. Click the Base Versions tab.
    b. Select the version that you require, then click Select.

Config Xpress analyzes the environment, then displays details of the environment.

You can now publish some or all of the environment as PDF (see page 206) or XML (see page 207). If you load a second environment, you can compare these environments and move components (see page 205) between them.

Example: Config Xpress after loading a baseline configuration file

This screenshot shows how Config Xpress displays dependent objects:



Move a Component from One Environment to Another

Without Config Xpress, the task of moving components between staging areas is complex and likely to fail.

When you use Config Xpress to move components, the tool also moves all required objects. For example, if you move a task that requires a screen, Config Xpress asks if you want to select the required components also. Config Xpress understands that the task uses this screen and should also be moved to the target environment.

If you want to move a component to a live environment, Config Xpress uploads it immediately. If you want to move the component to an environment file, save the component as an XML file and then import that file into the environment.

Follow these steps:

1. Load the environment that contains the component that you want to move.
2. Compare this environment with a second one:
   a. Click Compare.
   b. Load the target environment.

      Config Xpress displays a list of the differences between the two environments.
3. In the list of differences, find a component that you want to move. You can click the Name column to sort the list.
4. For each component, do the following steps:
   a. Select the item in the Action column.

      Config Xpress analyzes the component, which can take some time.
   b. If the component has any dependent components, the Add Modified Dependant Screens box appears. Click Yes or No to continue.

   When you have selected all of the components that you want to move, you are ready to move the updated components.
5. If you are moving the components to a live server, click Upload To.

   The components are moved immediately.
6. If you are moving the components to an environment file:
   a. Click Save.
   b. Enter a file name, then click Save again.

      Config Xpress saves all of the components that you selected in an XML file. You can now import this XML file into the actual target environment.

Publish a PDF Report

Config Xpress can generate a report that documents the current state of a CA Identity Manager environment. You can use this report to take a snapshot of a production environment. When you generate the report, you choose whether to include the complete configuration, or just the changes since installation.

This report is useful for future reference or as a part of a system recovery plan.

Follow these steps:

1. Load an environment into Config Xpress.
2. Click Generate Report.

   In the Generate PDF Report dialog, you can change the font size and you can enter text for the title or cover pages. You also choose whether to include all configuration items, or only new or modified items.

   Important! If you do not click the *Only include details of new or modified tasks, screens, roles* box, the report will contain the entire environment. The PDF file will be about 2000 pages long, and over 40 MB.
3. Click OK.
4. Enter a filename, then save the report. Saving may take a few minutes, or much longer if you chose to publish the whole environment.

   The report opens in the PDF reader.

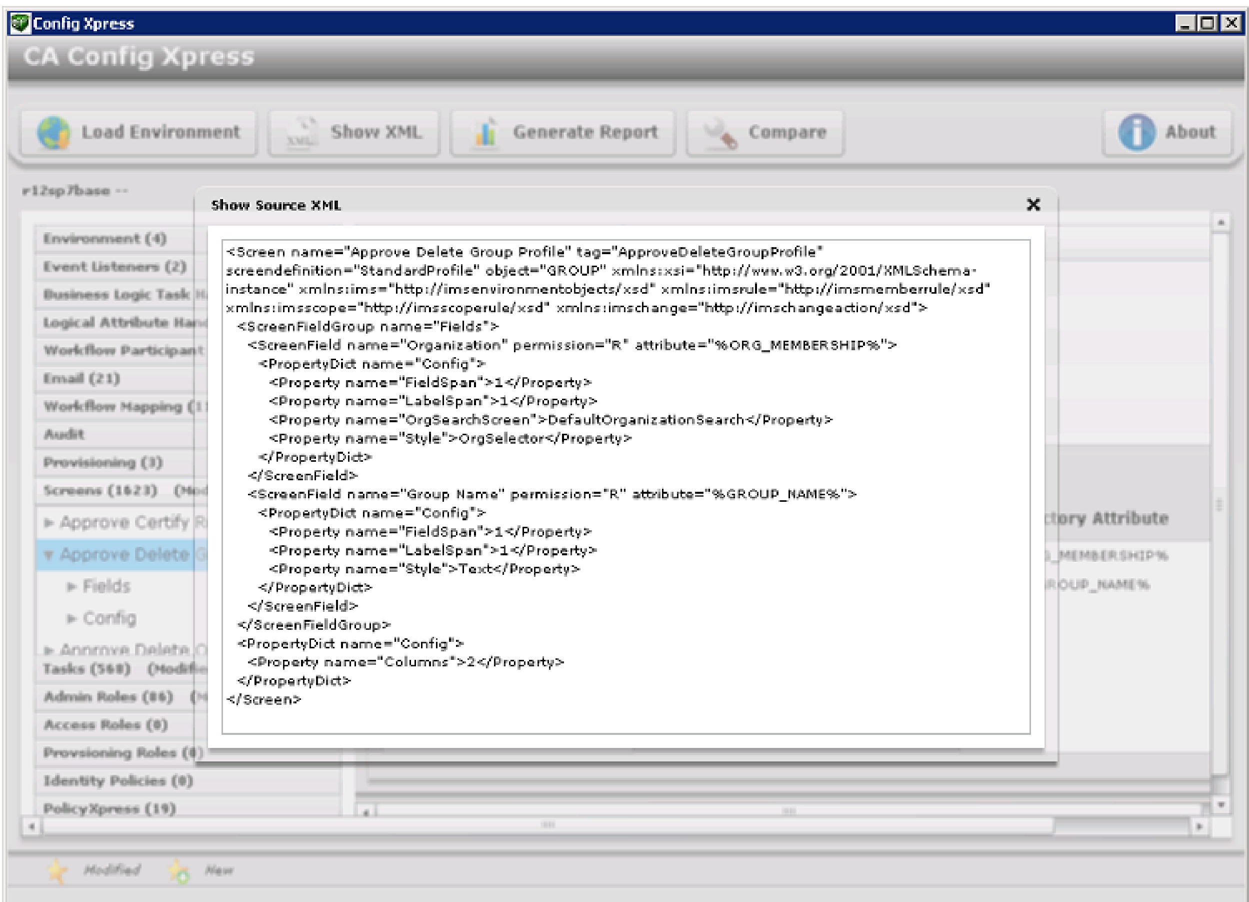
Display XML Configuration

Config Xpress can display the XML configuration for a particular component. You can study this XML file to help you understand a system.

Follow these steps:

1. Load an environment into Config Xpress.
2. Click on a component in the Config Xpress screen.
3. Click Show XML.

   The XML configuration appears:



Optimize Policy Rule Evaluation

Policy rules, which dynamically identify a set of users, are used in the evaluation of role member, admin, and owner policies, and identity policies. The evaluation of these rules can take significant time in large CA Identity Manager implementations.

Note: For more information about member, admin, owner, and identity policies, see the *Administration Guide*.

To reduce the evaluation time for rules that include user-attributes, you can enable the in-memory evaluation option. When the in-memory evaluation option is enabled, CA Identity Manager retrieves information about a user to be evaluated from the user store and stores a representation of that user in memory. CA Identity Manager uses the in-memory representation to compare attribute values against policy rules. This limits the number of calls CA Identity Manager makes directly to the user store.

You enable the in-memory evaluation option for an environment in the Management Console.

Follow these steps:

1. Open the Management Console.
2. Select Environments, *Environment Name*, Advanced Settings, Miscellaneous.

   The User Defined Properties page opens.
3. Enter the following text in the Property field:

   UseInMemoryEvaluation
4. Enter *one* of the following numbers in the Value field:

   0

   In-memory evaluation is disabled.

   1

   In-memory evaluation is enabled. When this option is specified, the attribute comparison is case-sensitive.

   3

   In-memory evaluation is enabled. When this option is specified, the attribute comparison is not case-sensitive.
5. Click Add.

   CA Identity Manager adds the new property to the list of existing properties for the environment.
6. Click Save.

Role and Task Settings

From the Role and Task Settings screen in the Management Console, you can import or export screen, tab, role and task settings in an XML file, called a Role Definitions file. CA Identity Manager provides predefined Role Definitions files that create screens, tabs, roles, and tasks for a set of functionality. For example, there is a Role Definitions file that supports Smart Provisioning, and other files that support endpoint management screens.

Additionally, you can use a Role Definitions file to apply the settings from one environment to multiple environments. Perform the following tasks:

- Configure screen, tab, task and role settings in one environment.
- Export these settings to an XML file.
- Import the XML file to the required environment.

Export Role and Task Settings

Perform the following procedure to export role and task settings.

Follow these steps:

1. In the Management Console, click Environments.

   A list of CA Identity Manager environments appears.

2. Click the name of the appropriate CA Identity Manager environment.

   The Properties screen for that environment appears.

3. Click Role and Task Settings, and click Export.
4. Click Open to view the file in a browser window or Save to save the settings in an XML file.

Import Role and Task Settings

Role and task settings are defined in XML files, named Role Definitions files. You can import predefined Role Definitions files to support specific sets of CA Identity Manager functionality (for example, Smart Provisioning) or import Role Definitions files from one environment to another.

Note: You can also import role definitions for custom connectors that are created with Connector Xpress. You create these role definitions files with the Role Definitions Generator. For more information, see the *Connector Xpress Guide*.

Perform the following procedure to import role and task settings.

Follow these steps:

1. In the Management Console, click Environments.

   A list of CA Identity Manager environments appears.

2. Click the name of the CA Identity Manager environment where you want to import the role and task settings.

   The Properties screen for that environment appears.

3. Click Role and Task Settings, and click Import.

4. Complete one of the following actions:

   - Select one or more Role Definitions files to create default roles and tasks for the environment.

     To select all available Role Definitions files, click Select/Deselect All.

   - Type the path and file name for the role definitions file to import or browse for the file. Then click Finish.

5. Click Finish.

   The status is displayed in the Role Configuration Output window.

6. Click Continue to exit.

How to Create Roles and Tasks for Dynamic Endpoints

Using Connector Xpress, you can configure dynamic connectors to allow provisioning and management of SQL databases and LDAP directories. For each dynamic connector, you can use the Role Definitions Generator to create task and screen definitions for account management screens that appear in the User Console.

After you run the Role Definitions Generator, you import the resulting Role Definitions file (see page 209) in the Management Console.

Note: For more information about the Role Definitions Generator, see the *Connector Xpress Guide*.

Modify the System Manager Account

A system manager is responsible for setting up and maintaining a CA Identity Manager environment. Typically, tasks of a system manager include:

- Creating and managing the initial environment
- Creating and modifying admin roles
- Creating and modifying other administrator accounts

You create a system manager account when you create a CA Identity Manager environment. If this account is "locked out," for example, if the system manager forgets password—you can re-create the account using the System Manager wizard.

The System Manager wizard guides you through the steps to assign a system management role to a user.

Note the following points before modifying the System Manager account:

- Assure that you are using an LDAP user store and you have configured a user container such as ou=People in the directory configuration file (directory.xml) for your CA Identity Manager directory. The selected users must exist in the same container where you configure the system manager. Selecting a user account that does not exist in the user container may cause failures.
- When the CA Identity Manager environment manages a user directory with a flat or flat user structure, the profile of the selected user must also include the organization. To ensure that the profile of a user is configured correctly, add the name of the organization of the user to the physical attribute corresponding to the %ORG_MEMBERSHIP% well-known attribute in the directory.xml file (see page 81). For example, when the physical attribute description is mapped to the %ORG_MEMBERSHIP% well-known attribute in the directory.xml file and the user belongs to the Employees organization, the profile of the user must contain the attribute/value pair description=Employees.

Follow these steps:

1. At the CA Identity Manager environments screen, click the name of the appropriate CA Identity Manager environment.

   The properties of that particular environment screen appear.

2. Click System Manager.

   The System Manager wizard appears.

3. Type the unique name for the user that has the System Manager role as follows:

   – For relational database users, type the unique identifier for the user or the value that is mapped to the %USER_ID% well-known attribute in the directory configuration file.

- For LDAP users, type the relative DN of the user. For example, if the DN of the user is uid=Admin1, ou=People, ou=Employees, ou=NeteAuto, type Admin1.

Note: Make sure that the System Manager is *not* the same user as the administrator of the user store.

4. Click Validate to display the full identifier of the user.
5. Click Next.
6. In the second page of the wizard, select a role to assign to the user as follows:
   - If you want to assign the System Manager role, do the following tasks:
     a. Select the radio button next to System Manager role.
     b. Click Finish.
   - If you want to assign a role other than the System Manager role, do the following tasks:
     a. Select a condition in the first list.
     b. Type a partial or complete role name or an asterisk (*) in the second list box. Click Search.
     c. Select the role to assign from the search results list.
     d. Click Finish.

   The System Manager Configuration Output screen displays status information.
7. Click Continue to close the System Manager wizard.

Access the Status of a CA Identity Manager Environment

CA Identity Manager includes a status page that you can use to verify the following status:

- The CA Identity Manager directory is loaded correctly.
- CA Identity Manager can connect to the user store.
- The CA Identity Manager Environment loads correctly.

Follow these steps:

1. Copy the status.jsp file from the following location:
   `C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\IAM Suite\Identity Manager\tools\samples\Admin\user_console.war`
2. Paste the status.jsp file in the following IAM application EAR location:
   `C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager/iam_im.ear/user_console.war`
3. Type the following URL in a browser:

   http://*hostname*/iam/im/status.jsp

   hostname

   Determines the fully qualified domain name of the server where CA Identity Manager is installed, for example, myserver.mycompany.com.

If the CA Identity Manager Environment starts correctly and all of the connections are running successfully, the status page resembles the following illustration:

Environment	Directory	Status
test1	Admin	OK
test2	NeteAuto	OK

The status page also indicates whether the environment is FIPS 140-2 compliant.

Troubleshooting CA Identity Manager Environments

The following table describes possible error messages and the troubleshooting process:

Message	Description	Troubleshooting
Not Loaded	The CA Identity Manager Directory which is associated with the environment was not loaded when CA Identity Manager started.	1. Verify that the user store is running. If CA Identity Manager integrates with SiteMinder, verify that SiteMinder can connect to the user store.

Message	Description	Troubleshooting
Not OK	CA Identity Manager cannot connect to the CA Identity Manager Directory.	In the Policy Server user interface, you can verify the connection by opening the properties page for the SiteMinder User Directory connection which is associated with the user store and click the View Contents button. If you can see the contents of the user store, SiteMinder can connect successfully. For more information about the Policy Server, see the CA *SiteMinder Web Access Manager Policy Server Configuration Guide*. 2. Restart CA Identity Manager and the Policy Server.
SM connection is not OK	CA Identity Manager cannot connect to the SiteMinder Policy Server (for implementations that include SiteMinder)	1. Verify the following conditions: ■ The Policy Server is running. ■ The Web agent is protecting resources. You can verify that the web agent is running correctly by accessing the Policy Server user interface. If you are prompted for credentials, the web agent is functioning correctly. 2. Restart CA Identity Manager and the Policy Server.
IMS is not available now	An error has occurred in CA Identity Manager.	Check the application server log for error details.
Windows 500 error message	The status page is not displayed when it is accessed while removing the connectivity with the LDAP user directory.	Set the internet browser option "Show friendly error message" to off to view the status page.

Chapter 7: Advanced Settings

The Advanced Settings window in the Management Console lets you do the following settings:

- Access screens for configuring advanced settings
- Import and export advanced settings as described in Import/Export Custom Settings (see page 227).

This section contains the following topics:

Auditing

Auditing logs maintain records of the operations performed in a CA Identity Manager environment. You can use the data in audit logs to monitor system activity.

CA Identity Manager audits *events*. An event is an operation that is generated by a CA Identity Manager task. A task can generate multiple events. For example, the CreateUser task may generate the CreateUserEvent and the AddToGroupEvent events.

By default, CA Identity Manager exports all event information to the audit database. To control the type and amount of event information that CA Identity Manager records, you can do the following tasks:

- Enable auditing for CA Identity Manager admin tasks.
- Enable auditing for some or all of the CA Identity Manager events generated by admin tasks.
- Record event information at specific states, for example, when an event completes or is canceled.
- Log information about attributes involved in an event. For example, you can log attributes that change during a ModifyUserEvent.
- Set the audit level for events and attributes.

Business Logic Task Handlers

A Business Logic Task Handler performs custom business logic before a CA Identity Manager task is submitted for processing. Typically, the custom business logic validates data. For example, a business logic task handler may check the membership limit of a group before CA Identity Manager adds a member to the group. When the group membership limit is reached, the business logic task handler displays a message informing the group administrator that the new member could not be added.

You can use the predefined business logic task handlers or can create custom handlers using the Business Logic Task Handler API.

Note: For information about creating custom business logic, see *Programming Guide for Java*.

The Business Logic Task Handlers screen contains a list of existing global business logic task handlers. The list includes predefined handlers shipped with CA Identity Manager and any custom handlers defined at your site. CA Identity Manager executes the handlers in the order in which they appear in this list.

Global business logic task handlers can be implemented only in Java.

Automatically Clear Password Fields on Reset User Password Task

You can configure CA Identity Manager to automatically clear password fields when a previously entered value violates a password policy, or when values in the Password and Confirm Password fields do not match.

Follow these steps:

1. Start the Management Console.
2. Select the environment that you want to manage, then click Advanced Settings.

   The Advanced Settings page appears.
3. Click Business Logic Task Handlers, BlthPasswordServices.

   The Business Logic Handler Properties page appears.
4. Create the following properties:

   ClearPwdIfInvalid=true

   PwdConfirmAttrName=|passwordConfirm|
5. Verify that ConfirmPasswordHandler settings are as follows:
   - Object type – User
   - Class – ConfirmPasswordHandler
   - ConfirmationAttributeName = |passwordConfirm|
   - OldPasswordAttributeName = |oldPassword|
   - passwordAttributeName = %PASSWORD%

   Users can now clear password fields in the Reset User Password task.

Event List

Admin tasks include *events*, actions that CA Identity Manager performs to complete the task. A task may include multiple events. For example, the Create User task may include events of creating the profile of a user, add the user to a group, and assign roles.

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

Use this page to view a list of the events that are available in CA Identity Manager.

Email Notifications

CA Identity Manager can send email notifications when a task or event completes, or when an event under workflow control reaches a specific state. For example, an email can inform an approver that an event requires approval.

To specify the content of email notifications, you can use predefined email templates or can customize the templates to suit your needs.

Using the Management Console, you can do the following tasks:

- Enable email notifications for a CA Identity Manager environment.
- Specify the template sets for creating email messages.
- Indicate the events and tasks for which email notifications are sent.

Event Listeners

A CA Identity Manager task is made up of one or more actions, named events that CA Identity Manager performs during the execution of the task. For example, the Create User task may include the following events:

- CreateUserEvent—Creates a user profile in an organization
- AddToGroupEvent—(Optional) Adds the user as a member of a group
- AssignAccessRole—(Optional) Assigns an access role to the user

An *event listener* "listens" for a specific event, and then performs custom business logic at a specific point in the lifecycle of an event. For example, after a new user is created in CA Identity Manager, an event listener may add the information of a user to a database of another application.

Note: For more information on configuring event listeners, see the *Programming Guide for Java*.

Identity Policies

An identity policy applies a set of business changes to users who meet certain rules or conditions. You can use identity policies to do the following tasks:

- Automate certain identity management tasks, such as assigning roles and group membership, allocating resources, or modifying user profile attributes.
- Enforce segregation of duties. For example, you can create an identity policy that prohibits members of the Check Signer role from having the Check Approver role.

- Enforce compliance. For example, you can audit users who have a certain title and can make more than $100,000.

You create and manage identity policy sets in the User Console. For more information on identity policies, see the *Administration Guide*.

Before you use identity policies, use the Management Console to do the following tasks:

- Enable identity policies for a CA Identity Manager environment.
- Set the recursion level (optional).

Logical Attributes Handlers

CA Identity Manager logical attributes let you display attributes of user store (named *physical attributes*) in a user-friendly format on task screens. CA Identity Manager administrators use task screens to perform functions in CA Identity Manager.

Logical attributes do not exist in a user store. Typically, they represent one or more physical attributes to simplify presentation. For example, the logical attribute *date* can represent the physical attributes *month*, *day*, and *year*.

Logical attributes are processed by the Logical attribute which are Java objects that are written using the Logical Attribute API. For example, when a task screen is displayed, a logical attribute handler might convert a physical attribute data from the user store to logical attribute data.

You can use predefined logical attributes and the logical attribute handlers included with CA Identity Manager or can create new ones using the Logical Attribute API.

Note: For more information, see the *Programming Guide for Java*.

Miscellaneous

User-defined properties that are defined on this screen apply to the entire CA Identity Manager environment. They are passed as name/value pairs to the init() method of every custom Java object that you create with the CA Identity Manager APIs. A custom object can use this data in any way that the object's business logic requires.

User-defined properties are also defined for a particular custom object. For example, assume that the user-defined properties are defined in the Properties screen for an event listener named MyListener. The object-specific user-defined properties and the environment-wide properties defined in the Miscellaneous screens are passed in a single call to MyListener.init().

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

To delete one or more user-defined properties, select the check box next to each name/value pair to delete, and click Delete.

Once the changes are made, click Save. Restart the application server for the changes to take effect.

Note: All miscellaneous properties are case-sensitive. Therefore, if you define a property named SelfRegistrationLogoutUrl and another property named selfregistrationlogouturl, both properties are added.

Notification Rules

A notification rule determines users to receive email notification. When a task completes or an event in a task reaches a certain state such as pending approval, approved, or rejected, the users receive an email notification according to the notification rule.

Note: For more information about the email notification feature, see the *Administration Guide*.

CA Identity Manager includes the following predefined notification rules:

ADMIN_ADAPTER

Sends an email message to the administrator who initiates the task

USER_ADAPTER

Sends an email message to the user affected by the task

USER_MANAGER

Sends an email to the manager of the user in the current context

To create custom notification rules, use the Notification Rule API.

Note: For more information about notification rules, see the *Programming Guide for Java*.

Organization Selectors

An organization selector is a custom logical attribute handler that determines where CA Identity Manager creates the profile of a self-registered user, which is based on information the user provides during registration. For example, the profile for users who provide a promotional code when they register may be added to a Promotional Users organization.

Provisioning

Use this screen when you are using CA Identity Manager with provisioning.

Note: A more detailed procedure, How to Configure an Environment for Provisioning (see page 182), provides step by step instructions.

The Provisioning Properties options are as follows:

Enabled

Specifies the use of two user stores, one for CA Identity Manager and a separate user store (called the Provisioning Directory) for provisioning accounts. If this option is disabled, only the CA Identity Manager user store is used.

Use Session Pool

Enables the use of a session pool.

Session Pool Initial Sessions

Defines the minimum number of sessions that are available in the pool at startup.

Default: 8

Session Pool Maximum Sessions

Defines the maximum number of sessions in the pool.

Default: 32

Enable Password Changes from Endpoint Accounts

Defines the setting for the Enable Password Synchronization Agent for each user in the Provisioning Server. This option allows password synchronization between CA Identity Manager users and associated endpoint accounts.

Enable Accumulation of Provisioning Role Membership Events

If enabled, this checkbox ensures that CA Identity Manager executes events that are related to the provisioning role membership in a specific order. All Add actions are combined into a single operation and sent to the Provisioning Server for processing. Once processing of the Add actions completes, CA Identity Manager combines the Remove actions into a single operation and sends that operation to the Provisioning Server. A single event, called AccumulatedProvisioningRoleEvent, is generated to execute the events in this order.

Note: For more information about the AccumulatedProvisioningRoleEvent, see the *Administration Guide*.

Organization for Creating Inbound Users

Defines the fully qualified path to the user store that CA Identity Manager uses. This field appears only when the user store includes an organization.

Inbound Administrator

Defines a CA Identity Manager administrator account that can execute tasks that are mapped to inbound mappings. These tasks are included in the Provisioning Synchronization Manager role. The administrator must be able to execute each task on any CA Identity Manager user.

Provisioning Directory

The Provisioning Directory is a repository for provisioning information including domain, global users, endpoint types, endpoints, accounts, and account templates. When you select it, other options appear for mapping the CA Identity Manager user store to the Provisioning Directory.

Enable Session Pooling

To improve performance, CA Identity Manager can preallocate a number of sessions for pooling when communicating with the Provisioning Server.

If the Session Pools option is disabled, CA Identity Manager creates and destroy sessions as needed.

For a new environment, Session Pools are enabled by default. For existing environments, you can enable Session Pools.

Follow these steps:

1. In the Management Console, choose Advanced Settings, Provisioning.
2. Select Use Session Pool.
3. Define the minimum number of sessions in the pool at startup.
4. Define the maximum number of sessions in the pool.
5. Click Save.
6. Restart the Application Server.

   The Session Pool is enabled per the defined settings.

Enable Password Synchronization

The Provisioning Server allows password synchronization between CA Identity Manager users and associated endpoint user accounts. In other words, when a user who has provisioning roles is created or modified in CA Identity Manager, the provisioning user gets set to allow password changes from endpoint accounts.

Note: When you enable this feature in the Management Console, *all* users in the environment gets set to allow password changes from endpoint accounts.

Follow these steps:

1. In the Management Console, choose Advanced Settings, Provisioning.
2. Check Enable Password Changes from Endpoint Accounts.
3. Click Save.
4. Restart the Application Server.

Attribute Mappings

Attribute mappings associate the user attributes in provisioning-related admin tasks, such as Provision Create User, with the corresponding attributes in the Provisioning Server. A single provisioning attribute can be mapped to multiple attributes in the CA Identity Manager user store.

Default mappings exist for the attributes in the default tasks, which are listed in the Inbound Mappings section. If you modify one of these admin tasks in a way to use different attributes, update the attribute mappings as needed.

Inbound Mappings

Inbound mappings map events, generated by the Provisioning Server, to an admin task. These mappings are preset and cannot be modified.

Outbound Mappings

Outbound Mappings associate events, which are generated by admin tasks, with events that are applied to the Provisioning Directory. Default mappings exist for the events that affect user attributes.

User Console

You access a CA Identity Manager Environment using the User Console, a web application that allows users to perform admin tasks. You define certain properties for the User Console that administrators use to access an Environment in the User Console page in the Management Console.

The User Console page includes the following fields:

General Properties

Define properties that apply to an Environment.

Show Recently Completed Tasks

Determines whether CA Identity Manager displays a status message when a task completes.

When this option is selected, users must click OK to clear the status message that CA Identity Manager displays.

To disable the message and prevent users from having to click OK when each status message appears, clear this option.

Show About Link

Determines whether an About link appears in the lower-right corner of the User Console. When this option is selected, CA Identity Manager users can click the About link to view version information for CA Identity Manager components.

Enable Language Switching

Determines whether CA Identity Manager includes a Choose Language drop-down list in the login screen and in the User Console. When this field is selected, CA Identity Manager users can change the language in the User Console by selecting a new language from the list.

Note: To display the Choose Language field, verify that you select the Enable Language Switching field, *and* configure CA Identity Manager to support multiple languages.

See the *User Console Design Guide* for more information.

Job Timeout

Determines the amount of time that CA Identity Manager waits after a task is submitted before displaying a status message.

When the task completes within the specified amount of time, CA Identity Manager displays the following message:

"Task Completed"

If the task takes longer to complete or is under workflow control, CA Identity Manager displays the following message:

"Task has been submitted for processing on the *current date"*

Note: Changes may not take effect immediately.

Theme Properties

Let you customize the icon and title of the User Console in an Environment. For example, you can add a company logo and the company name to User Console screens.

Theme properties include the following settings:

Icon (URI)

Defines the icon using a URI to an image available to the application server.

Example: http://myserver.mycompany.com/images/front/logo.gif

Icon Link (URI)

Defines the navigation link to the image using a URI.

Icon Title

Defines the tooltip which appears as a mouseover text on the icon.

Title

Specifies custom text, which is displayed next to the icon at the top of the User Console.

Note: If you defined a custom skin, you can specify an icon or title by referencing a properties file for the skin. For example, if the entry for the icon image in the properties file for a custom skin is image/logo.gif, you can enter that same string in the Icon field.

Login Properties

Specify the authentication method and location of the login page to which users are directed when they access an Environment.

Authentication Provider module class name

Specifies the class name of the authentication provider module.

Login Page

Specifies the page to which users are directed when they access an Environment.

Web Services

The CA Identity Manager Task Execution Web Service (TEWS) enables third-party client applications to submit CA Identity Manager tasks to CA Identity Manager for execution remotely.

The Web Services Properties screen lets you configure the TEWS for an environment. On this screen, you can do the following tasks:

- Enable TEWS for a CA Identity Manager environment.
- Generate task-specific Web Services Definition Language (WSDL) documents.
- Allow impersonation.
- Specify that the Admin password is required for authentication.
- Configure SiteMinder authentication.
- Configure SiteMinder to secure the web services URL, if CA Identity Manager integrates with SiteMinder
- Specify Web Security Services Username token authentication.
- Specify at least one of the three possible authentication types.

For information regarding issuing remote requests to CA Identity Manager through the Task Execution Web Service, see the *Programming Guide for Java*.

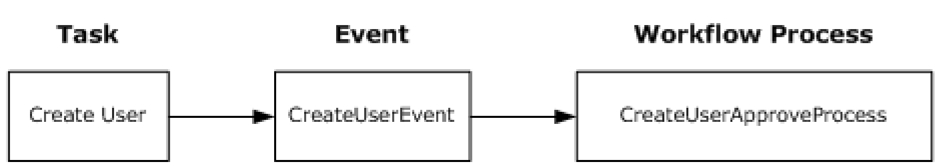
Workflow Properties

If enabled, the workflow feature controls the execution of a CA Identity Manager task that is associated with a workflow process.

A workflow process is a set of steps that are performed to accomplish a business objective, such as creating a user account. Typically, one of these steps involves approving or rejecting the task.

An admin task is associated with one or more events, which may trigger one or more workflow processes. After the workflow processes complete, CA Identity Manager performs or rejects the task which is based on the results of the workflow processes.

The following illustration shows the relationship between a CA Identity Manager task, an associated event, and a workflow process:



Workflow Properties

Use the check box to enable or disable workflow for the CA Identity Manager Environment.

Work Item Delegation

If enabled, work item delegation allows a participant (the delegator) to specify that another user (the delegate) to get permissions to approve tasks in the work list of the delegator. A participant can assign work items to another approver during periods when the delegator is "out of the office." Delegators retain full access to their work items during the delegation period.

Delegation uses the following well-known attribute:

%DELEGATORS%

This well-known attribute stores the names of users who are delegating to the user with the attribute, as well as the time when the delegation was created.

Note: For more information about work item delegation, see the *Administration Guide*.

Workflow Participant Resolvers

The activities in a workflow process, such as approving or rejecting a task are performed by *participants.*

You use the Workflow Participant Resolvers screen to map a custom participant resolver to a fully qualified participant resolver Java class.

A custom *participant resolver* is a Java object that determines participants of a workflow activity and returns a list to CA Identity Manager. CA Identity Manager then passes the list to the workflow engine.

Typically, you write a custom participant resolver only if none of the standard participant resolvers can provide the list of participants that an activity requires.

Note: For information about developing custom participant resolvers, see the *Programming Guide for Java*. For information about standard participant resolvers, see the *Administration Guide*.

Import/Export Custom Settings

From the Advanced Settings screen in the Management Console, you can apply advanced settings to multiple environments, as follows:

- Configure advanced settings in one environment.
- Export the advanced settings to an XML file.
- Import the XML file to the required environments.

Java Virtual Machine Out-of-memory Errors

Symptom:

I receive JVM out-of-memory errors during stress or high load periods that affect the functionality of the CA Identity Manager Server.

Solution:

We recommend that you set JVM debugging options so that you are alerted on out-of-memory conditions.

Note: For more information about setting JVM debugging options, see Debugging Options in Java HotSpot VM Options at http://www.oracle.com.

Chapter 8: Auditing

This section contains the following topics:

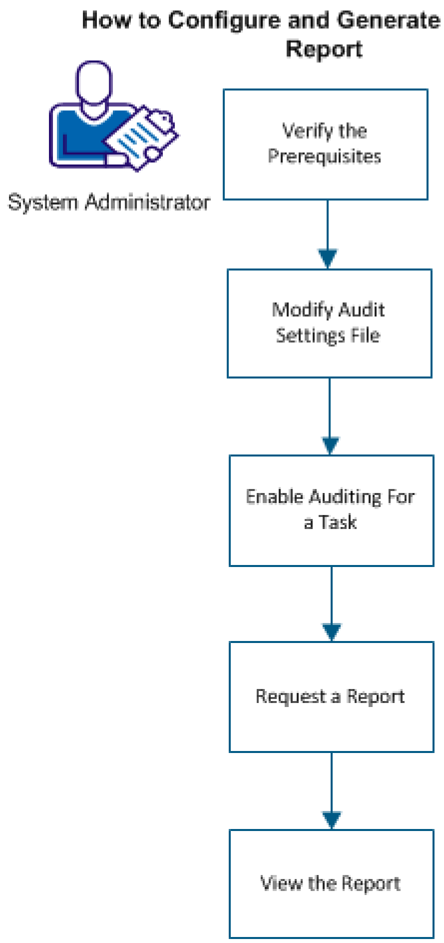
How to Configure and Generate Audit Data Report

Audit data provides a historical record of operations that occur in an environment. When you configure and enable auditing, the system records information about the tasks in an auditing database. The audit information can be used to generate reports. Some examples of audit data include the following points:

- System activity for a specified period of a time.
- User login and logout events while accessing a particular environment.
- The tasks that a specific user performs
- A list of objects that were modified during a specific period of a time.
- The user assigned roles
- The operations which are performed for a particular user account.

Audit data is generated for CA Identity Manager *events*. An event is an operation that is generated by a CA Identity Manager task. For example, the Create User task may include an AssignAccessRoleEvent event.

The following diagram describes how a system administrator configures auditing and generates a report on audit data:



As an administrator, complete the following steps:

1. Verify Prerequisites (see page 231)
2. Modify Audit Settings File (see page 231)
3. Enable Auditing For A Task (see page 236)
4. Request a Report (see page 237)
5. View the Report (see page 239)

Verify the Prerequisites

Verify that the following prerequisites are met before you configure audit settings:

- A separate database instance is created for storing data that is related to auditing. By default, the CA Identity Manager database schema file is located in the following location:
  - **Windows**: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\Identity Manager\tools\db
- Configure the report server connection to request and view the audit report.
- Add a connection object for the audit report. Perform the following steps:
  a. Log in to the user console with administrative privileges.
  b. Go to Roles and Tasks, Admin Tasks, and search for an audit report to modify.
  c. Enter the following connection name in the Connection Object for the Report field:
     `rptParamConn`

Modify Audit Settings File

Configure audit settings in the audit settings file to define the type of information that CA Identity Manager must audit. You can configure an audit settings file to perform the following tasks:

- Audit some or all of the admin tasks generated events.
- Record event information at specific states, such as when an event completes or is canceled.
- Log information about attributes that are involved in an event. For example, you can log attributes that change during a ModifyUserEvent event.

- Set the audit level for attribute logging.

The audit settings file is an XML file that you create by exporting audit settings. The file has the following schema:

```
<Audit enabled="" auditlevel="" datasource="">
  <AuditEvent name="" enabled="" auditlevel="">
    <AuditProfile objecttype="" auditlevel="">
      <AuditProfileAttribute name="" auditlevel="" />
    </AuditProfile>
    <EventState name="" severity=""/>
  </AuditEvent>
</Audit>
```

For more information about the Audit elements and schema, see the comments in the Audit Settings file.

The AuditProfileAttribute elements indicate the attributes that CA Identity Manager audits. The attributes apply to the object specified in the AuditProfile element.

Note: If there are no audit profile attributes specified, all the attributes for the object that are specified in the AuditProfile element are logged.

The following table shows the valid attributes for CA Identity Manager object types:

Valid Attributes for CA Identity Manager Object Types

Object Type	Valid Attributes
ACCESS ROLE	■ name—User-visible name for the role ■ description—An optional comment about the purpose of the role. ■ members—The users who can use the role. ■ administrators—The users who can assign role member or administrators. ■ owners—The users who can modify the role. ■ enabled—Indicates whether the role is enabled or not. ■ assignable—Indicates whether the role assignable by an administrator or not. ■ tasks—The access tasks that are associated with the role.

Valid Attributes for CA Identity Manager Object Types

Object Type	Valid Attributes
ACCESS TASK	■ name—User-visible name for the task ■ description—An optional comment about the purpose of the task ■ application—The application that is associated with the task. ■ tag—The unique identifier for the task ■ reserved1, reserved2, reserved3, reserved4—The values of the reserved fields for the task
ADMINISTRATIVE ROLE	■ name—User-visible name for the role ■ description—An optional comment about the purpose of the role ■ members—The users who can use the role. ■ administrators—The users who can assign role member or administrators. ■ owners—The users who can modify the role. ■ enabled—Indicates whether the role is enabled or not. ■ assignable—Indicates whether the role assignable by an administrator or not. ■ tasks—The tasks that are associated with the role.

Valid Attributes for CA Identity Manager Object Types

Object Type	Valid Attributes
ADMINISTRATIVE TASK	■ name—User-visible name for the task ■ description—An optional comment about the purpose of the task ■ tag—The unique identifier for the task ■ category—The category in the CA Identity Manager user interface where the task appears ■ primary_object—The object on which the task operates ■ action—The operation that is performed on the object. ■ hidden—Indicates whether the task does *not* appear in menus. ■ public—Indicates whether the task is available to users who have not logged in to CA Identity Manager. ■ auditing—Indicates whether the task enables the recording of auditing information. ■ external—Indicates whether the task is an external task. ■ url—The URL where CA Identity Manager redirects the user when an external task executes. ■ workflow—Indicates whether the CA Identity Manager events associated with the task trigger workflow ■ webservice—Indicates whether the task is one for which Web Services Description Language (WSDL) output can be generated from the CA Identity Manager Management Console.
GROUP	Any valid attribute that is defined for the GROUP object in the directory configuration file (directory.xml).
ORGANIZATION	Any valid attribute that is defined for the Organization object in the directory configuration file (directory.xml).
PARENTORG	

Valid Attributes for CA Identity Manager Object Types	
Object Type	**Valid Attributes**
RELATIONSHIP	■ %CONTAINER%—Unique identifier of the parent object. For example, if the RELATIONSHIP object describes role membership, the container would be the role. ■ %CONTAINER_NAME%—User-visible name of the parent group ■ %ITEM%—Unique identifier of the object that is contained in the parent object. For example, if the RELATIONSHIP object describes role membership, the items would be the role members. ■ %ITEM_NAME%—User-visible name for the nested group
USER	Any valid attribute that is defined for the USER object in the directory configuration file (directory.xml)
NONE	No attributes

Note: The following points apply to the preceding table:

- Enabled, assignable, auditable, workflow, hidden, webservice, and public are logged as true or false.
- When auditing tasks for roles, the user visible name is logged.
- The database stores member, administrator, and owner policies in compiled XML format. This format is different from the user interface where each policy appears as an expression.

Follow these steps:

1. Log in to the Management Console, select the environment, Advanced Settings, and click Auditing.
2. Click Export.

   The system exports the current audit settings to an audit settings XML file.

3. Modify the audit settings in the XML file that you exported in the previous step. Do the following tasks:
   a. Set the value for Audit enabled ="true" and provide the JNDI Name value of "iam_im_<auditdb>.xml" for the element Data source.
   b. Specify the following JNDI name:
      `java:/auditDbDataSource`

      Note: The datasource is located in the following location:
      `iam/im/jdbc/auditDbDataSource`
   c. Add, modify, or delete elements in the file.
   d. Modify the level of information that is recorded for each event.
4. Repeat steps 1 and 2. Click Import and upload the modified audit settings XML file.
5. Restart the environment.

The Audit settings file is now updated.

Enable Auditing For A Task

Enable auditing for the tasks for which you have configured auditing in the Audit Settings file.

Follow these steps:

1. Log in to the User Console with system administrator privileges.
2. Create or modify the task for which you want to enable auditing.
3. On the Profile tab, ensure that the Enable Auditing check box is selected.
4. Click Submit.

Auditing is now enabled to the task.

Request a Report

To view the report, request for a report to a user with report administration privileges. Select the appropriate report that tracks the audit data. If your report request requires an approval, the system sends you an email alert.

Before scheduling a report, perform the following steps:

1. Log in to the User Console with administrative privileges.
2. Go to Roles and Tasks, Modify Admin Task, and select an audit report to modify.
3. Select Tabs tab and click IAM ReportServerScheduler to edit.
4. Select the Enable Recurrence Option check box.
5. Click OK and Submit.

Follow these steps:

1. Log in to the User Console with report tasks user privileges.
2. Select Reports, Reporting Tasks, Request a Report.

   A list of reports appears.

3. Select an audit based report.

   A parameters screen appears.

4. Click Schedule Report, and select a schedule for your report.

   Now

   Specifies that the report runs immediately.

   Once

   Specifies that the report runs once, during a specific time period. Select the start date, end date, start time, and end time when you want to generate the report.

   (Audit Report Only) Hourly

   Specifies that the report is generated at the start time and subsequently every 'n' hours; 'n' denotes the interval between successive reports. Select the start date, end date, start time, end time, and the interval between successive reports.

   (Audit Report Only) Daily

   Specifies that the report is generated at the start time and subsequently every 'n' days; 'n' denotes the interval between successive reports. Select the start date, end date, start time, end time, and the interval between successive reports.

(Audit Report Only) Weekly

Specifies that the report is generated every week on the selected day from the start date. Select the start date, end date, start time, and end time when you want to generate the report.

(Audit Report Only) Monthly

Specifies that the report is generated monthly from the start date and subsequently every 'n' months. 'n' denotes the interval between successive reports. Select the start date, end date, start time, end time, and the interval between successive reports.

(Audit Report Only) Run the report on a Nth day of the month

Specifies that the report is generated on the specific day of the month you have mentioned. Select the start date, end date, start time, and end time when you want to generate the report.

(Audit Report Only) First Monday

Specifies that the report is generated every first Monday of the month. Select the start date, end date, start time, and end time when you want to generate the report.

(Audit Report Only) Last Day of the month

Specifies that the report is generated on the last day of the month. Select the start date, end date, start time, and end time when you want to generate the report.

(Audit Report Only) On Xth day of Nth week of every month

Specifies that the report is generated on a specific day in a specific week of every month. Select the start date, end date, start time, and end time when you want to generate the report. For example, you can generate a report on a Friday in the 3rd week of every month.

5. Click Submit.

   The report request is submitted. Depending on your environment configuration, the request runs immediately, or it runs after approval by an administrator.

   Typically, a system administrator or another user with report administration privileges must approve a report request before the system completes it. Approval is required because some reports can require a long time or significant system resources to run. If your report request requires approval, the system sends you an email alert.

   Note: Enable WorkFlow for the environment if approval is required.

View the Report

Depending on your environment configuration, a report will be available to view when an administrator approves the request for that report. If your report request is pending approval, the system sends you an email alert. The report that you want to view does not appear in the search list until it is approved.

Note: In order to view reports in CA Identity Manager using the View My Reports task, enable third-party session cookies in your browser.

Follow these steps:

1. In the User Console, go to Reports, Reporting Tasks, and click View My Reports.
2. Search for the generated report that you want to view.

   Both recurrence generated reports and on-demand report instances are displayed.

   Note: If the status of the report is Pending/Recurring, the report is not generated and may take time to complete.
3. Select the report that you want to view.
4. (Optional) Click Export this report (top left corner) to export the report to the following formats:
   - Crystal Report
   - PDF
   - Microsoft Excel (97-2003)
   - Microsoft Excel (97-2003) Data-Only
   - Microsoft Excel (97-2003) - Editable
   - Rich Text Format (RTF)
   - Seperated Values (CSV)
   - XML

Clean Up the Audit Database

The auditing database may eventually accumulate records that are no longer necessary. To remove these records, execute the following database procedure in the db\auditing directory:

`garbageCollectAuditing12` *`environment-ID MM/DD/YYYY`*

environment-ID

Defines the ID of the CA Identity Manager environment

MM/DD/YYYY

Defines the date before which the auditing records must be removed.

Chapter 9: Production Environments

This section provides step by step functional descriptions to migrate specific pieces of functionality. Ensure that it is used only when limited changes were made in the development environment and those changes are well-understood.

This section contains the following topics:

To migrate Admin roles and task definitions

You can customize CA Identity Manager roles and tasks to meet the specific needs of your company. The customization involves creating or modifying admin roles and tasks or by using a Create or Modify task for an admin role or task.

An alternative method, though *not recommended*, is to modify roles and tasks in the roledefinition.xml file. Use this method for very limited changes because of the risk of errors in editing.

This process will only migrate administrative role and task definitions. If the roles were bound to organizations, consider migrating the entire CA Identity Manager environment.

Important! If you changed role or task definitions in the production environment, those changes are lost when you import role or task definitions from a development environment. Importing role and task definitions overwrites existing role and task definitions with the same names.

To export Admin role and task definitions

If changes were made directly to the roledefinition.xml file, this file can be directly imported into the production environment. Otherwise, to export the role and task definitions:

1. If you have a Policy Server cluster, check that only one Policy Server is running.
2. Stop all but one CA Identity Manager node.
3. Log in to the Management Console.
4. Click CA Identity Manager environments.
5. Select the CA Identity Manager environment, from which to export the role and task definitions.
6. Click Roles, then click Export and supply a name for the file.
7. Follow the instructions in the next procedure to import this file.

To import Admin role and task definitions

Follow these steps:

1. Copy the file created in the preceding procedure to the production environment.
2. Log in to the Management Console in the production environment.
3. Click CA Identity Manager environments.
4. Select the appropriate CA Identity Manager environment.
5. Click Roles.
6. Click Import and specify the name of the XML file that export generates.
7. If these steps succeeded, start any extra Policy Servers and CA Identity Manager nodes that you stopped.

   Note: If there are still any changes to make in a CA Identity Manager environment, repeat the step 6.

To verify the role and task import

To verify that the roles and tasks were imported successfully, log in to CA Identity Manager as an administrator account that can use the following tasks:

- Modify Admin Role
- Modify Admin Task

Execute these tasks and verify that the roles and tasks reflect the newly imported role definitions.

To migrate CA Identity Manager skins

CA Identity Manager skins can be customized to give a specific look and feel to the application. If you have modified or created new skins for a set of users, use the following steps to migrate skins from the development to the production environment.

If you are modifying a skin, copy the modified files.

Follow these steps:

1. Copy new and modified files from the development to the production server such as image files, style sheets, properties files, and the console page (index.jsp).
2. If multiple skins are being used, configure SiteMinder response.

   Note: For more information on using multiple skins, see the *Configuration Guide*.

To verify the migration of skins, log in to the User Console and check that the skin appears correctly.

Update CA Identity Manager in a Production Environment

After migrating CA Identity Manager from development to production, you may need to perform incremental updates. To migrate new CA Identity Manager functionality from your development environment to your production environment, execute the following steps:

1. Migrate CA Identity Manager environments.
2. Copy the iam_im.ear.
3. Migrate workflow process definitions.

To migrate a CA Identity Manager environment

A CA Identity Manager Environment is created from the Management Console. CA Identity Manager environment includes a set of role and task definitions, workflow definitions, custom features that are created with CA Identity Manager APIs, and a CA Identity Manager Directory.

Follow these steps:

1. If CA Identity Manager integrates with SiteMinder and you have a Policy Server cluster, check that only one Policy Server is running.
2. Stop all but one CA Identity Manager node.
3. Export CA Identity Manager Environments from the Management Console in the development environment.

4. Import the exported environments in the Management Console in the production environment.

5. If CA Identity Manager integrates with SiteMinder, reprotect the CA Identity Manager realms in the Policy Server User Interface.

   The policy domain is not exported from the policy store when you export a CA Identity Manager Environment.

6. Restart the Policy Server and CA Identity Manager nodes that you stopped.

When migrating a CA Identity Manager Environment, the following activities occur:

- If the same object exists in both locations, the changes on the development server overwrite changes on the production server.
- If new objects are created on the development environment, they are added to the production server.
- If new objects are created on the production server, they are maintained.

To export a CA Identity Manager environment

To deploy a CA Identity Manager environment on a production system, you export the environment from a development or staging system and import that environment to the production system.

Note: When you import a previously exported environment, CA Identity Manager displays a log in a status window in the Management Console. To see validation and deployment information for each managed object and its attributes in this log, select the Enable Verbose Log Output field on the Environment Properties page *before* you export the environment. Make sure that selecting the Enable Verbose Log Output field can cause significant performance issues during the import.

Follow these steps:

1. Click Environments in the Management Console.

   The CA Identity Manager environments screen appears with a list of CA Identity Manager environments.

2. Select the environment that you want to export.

3. Click the Export button.

   A File Download screen appears.

4. Save the ZIP file to a location that is accessible to the production system.

5. Click Finish.

   The environment information is exported to a ZIP file that you can import into another environment.

To import a CA Identity Manager environment

After exporting a CA Identity Manager environment from a development system, you can import it into a production system.

Follow these steps:

1. Click Environments in the Management Console.

   The CA Identity Manager environments screen appears with a list of CA Identity Manager Environments.

2. Click the Import button.

   The Import Environment screen appears.

3. Browse for the ZIP file required to import an environment.
4. Click Finish.

The environment is imported into CA Identity Manager.

To verify the CA Identity Manager environment migration

To verify proper migration of the CA Identity Manager environment, confirm that the CA Identity Manager Environment appears in the Policy Server User Interface for the Policy Server in the production environment.

In the Policy Server User Interface, verify the following points:

- The CA Identity Manager user directory settings are accurate.
- The new CA Identity Manager domain exists
- The correct authentication schemes protect the CA Identity Manager realms.

Also upon logging in to the Management Console, verify that the CA Identity Manager Environment appears when you select the Environments.

Migrate the iam_im.ear for JBoss

Redeploy the iam_im.ear each time functionality is migrated from the development environment to the production environment. By migrating the entire EAR, you ensure that your production environment is identical to your development environment.

Follow these steps:

1. Copy the iam_im.ear from your development environment to a location accessible to your production environment.
2. In the copy of the iam_im.ear, edit the Policy Server connection information, so that it reflects the production environment.

   To achieve this change, copy the *jboss_home*/server/default/iam_im.ear/policyserver_rar/META-INF/ra.xml from your production environment into the iam_im.ear.
3. Replace the installed iam_im.ear with the copy of the iam_im.ear from your development environment from step 2 as follows:
   a. On the production server, delete the iam_im.ear:

      cluster_node_jboss_home\server\default\deploy\iam_im.ear
   b. Replace the deleted files with the edited copy of the iam_im.ear from the development environment.
4. Repeat these steps for each node in the cluster.

Migrate the iam_im.ear for WebLogic

Redeploy the iam_im.ear each time functionality is migrated from the development environment to the production environment. By migrating the entire EAR, you ensure that your production environment is identical to your development environment.

Follow these steps:

1. Preserve Policy Server connection information.

   Policy server connection information is stored in the ra.xml file in the policyserver_rar/WEB-INF directory. Copy this file to another location, so that it can be replaced in the iam_im.ear before redeploying it.
2. Copy the iam_im.ear to a location available to the WebLogic Admin Server.
3. Replace the Policy Server connection information.

   In the iam_im.ear, replace the policyserver_rar/WEB-INF/ra.xml file with the one preserved from Step 1.
4. Redeploy the iam_im.ear
   a. Log in to the WebLogic console.
   b. Go to Deployments, Application, Identity Manager.

On the Deploy Tab, select Deploy (Re-Deploy) Application.

Migrate the iam_im.ear for WebSphere

Follow these steps:

1. Copy the *imsInstall.jacl* script from *was_im_tools_dir**WebSphere-tools* to the *deployment_manager_dir* *bin* directory where:
   - *was_im_tools_dir* is the directory on the development system where the CA Identity Manager Tools for WebSphere are installed.
   - *deployment_manager_dir* is the location where the Deployment Manager is installed.
2. On the development system where you configured the CA Identity Manager application, copy *was_im_tools_dir* *WebSphere-tools**imsExport.bat* or *imsExport.sh* to *was_home**bin.*
3. On the command line, navigate to *was_home**bin.*
4. Make sure the WebSphere application server is running.
5. Export the deployed CA Identity Manager application as follows:

   For Windows, enter this command:

   ```
   imsExport.bat "path-to-exported-ear"
   ```

   where *path-to-exported-ear* is the full path and file name that the imsExport utility creates.

   For Windows systems, use forward slashes (/) instead of back slashes (\) when you specify the path to was_im.ear. For example:

   ```
   imsExport.bat "c:/program files/CA/CA Identity Manager/
   exported_ear/iam_im.ear"
   ```

   For UNIX, enter this command:

   ```
   ./wsadmin -f imsExport.jacl -conntype RMI -port 2809 path to exported ear
   ```

   where *path-to-exported-ear* is the full path including the file name of the exported EAR file.
6. Copy the exported EAR file from the location on the development system where you exported it to a location on the system where the Deployment Manager is installed.
7. Replace the *was_im_tools_dir*/WebSphere-ear/iam_im.ear/policyserver_rar/META-INF/ra.xml with the one from the production environment.

   The ra.xml file contains the Policy Server connection information.

8. On the system where the Deployment Manager is installed, deploy the Identity Manager EAR:

   a. From the command line, navigate to:

      deployment_manager_dir \bin.

   b. Make sure that the WebSphere application server is running.

   c. Run the imsInstall.jacl script as follows:

   Note: The imsInstall.jacl script may take several minutes to execute.

   Windows:

      wsadmin -f imsInstall.jacl *"path-to-copied-ear" cluster_name*

      where *path-to-copied-ear* is full-path including the file name for the Identity Manager EAR that you copied to the Deployment Manager system.

      For example:

      wsadmin -f imsInstall.jacl "c:\Program Files\CA\Identity Manager\WebSphere-tools\was_im.ear" im_cluster

   UNIX:

      ./wsadmin -f imsInstall.jacl *path-to-copied-ear cluster_name*

      where *path-to-copied-ear* is full-path including the file name for the Identity Manager EAR that you copied to the Deployment Manager system.

      For example:

      ./wsadmin -f imsInstall.jacl /opt/CA/Identity Manager/WebSphere-tools/was_im.ear im_cluster

9. If CA Identity Manager integrates with SiteMinder, verify the following points:

   - The SiteMinder agents can connect to your policy store.
   - The Policy Server can connect to the user store.
   - The CA Identity Manager domains have been created.

Migrate Workflow Process Definitions

If you used workflow in the development environment, export the workflow definitions and import them into the production environment. Then, configure workflow in each of the server nodes.

Export process definitions

On the development environment system, you export the workflow process definitions.

Follow these steps:

1. Make sure that the application server is running.
2. Go to *admin_tools*\Workpoint\bin\ and run Archive.bat (for Windows) or Archive.sh (for UNIX) as follows:
   a. In the Import dialog, select the root object.
   b. Click Add.
   c. Specify the name of the file to generate.
   d. Click Export.
   e. Click Go.

   admin_tools refers to the Administrative Tools, which are installed by default in one of the following locations:

   - **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
   - **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools
3. Follow the instructions in the next section, To import process definitions (see page 249).

Import process definitions

On the production environment system, import the workflow process definitions.

Follow these steps:

1. Restart the application server.
2. Optionally, create a backup copy of your current definitions by exporting the definitions using the preceding procedure.

3. Go to *admin_tools*\Workpoint\bin\ and run the Archive script as follows:

    a. In the Import dialog, select all items to import.

    b. When you are prompted about using the new or old format, retain the old format.

       The new format does not support CA Identity Manager.

    c. Supply the name of the file that the export generates.

    d. Click Go.

    admin_tools refers to the Administrative Tools, which are installed by default in one of the following locations:

    - **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
    - **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools

Chapter 10: CA Identity Manager Logs

This section contains the following topics:

How to Track Problems in CA Identity Manager

CA Identity Manager includes the following methods for recording status and tracking issues:

View Submitted Tasks task

Displays the status of all events and tasks in a CA Identity Manager environment. Administrators use this task in the User Console.

View Submitted Tasks provides the following types of information:

- The list of events and tasks that occur in the environment.
- The list of attributes that are associated with an event.
- Successful and failed events
- Events that are in a pending or stalled state.
- Rejected events, including the reason for rejection
- Account synchronization status
- Identity policy synchronization status
- Provisioning information (when provisioning is enabled).

Application Server logs

Displays information about all of the components in a CA Identity Manager installation, and provides details about all operations in CA Identity Manager.

The location and type of log file depends on which of the following types of application servers you are using:

- WebLogic—CA Identity Manager information is written to standard out. By default, standard out is the console window in which the server instance is running.
- JBoss—CA Identity Manager information is written to the console window where the server instance is running, and to *jboss_home*\server\log\server.log
- WebSphere—CA Identity Manager information is written to the console window where the server instance is running, and to *was_home*\AppServer\logs*server_name*\SystemOut

See the documentation for your application server for more information.

Directory Server log file

Contains information about activity that occurs in the user directory.

The type of information that is recorded and the location of the log file depend on the type of directory server that you are using. See the server documentation of directory for more information.

Policy Server log file

Displays the following information when CA Identity Manager integrates with SiteMinder:

- SiteMinder connection issues
- SiteMinder authentication issues
- Information about CA Identity Manager managed objects in the SiteMinder policy store.
- Password policy evaluation

For information about configuring SiteMinder logs, see the *CA SiteMinder Web Access Manager Policy Server Administration Guide*.

Policy Server Profiler

If CA Identity Manager integrates with SiteMinder, allows you to trace internal Policy Server diagnostics and processing functions, including functions that are related to CA Identity Manager.

For more information, see How to Trace Components and Data Fields (see page 253).

Web Agent log files

If CA Identity Manager integrates with SiteMinder, the web agents write information to the following two logs:

- Error log file—contains program and operational-level errors, for example, the web agent not being able to communicate with Policy Server.
- Trace log file—contains warning and informational messages, such as trace messages and flow state messages. It also includes data such as header details and cookie variables.

Note: For more information about Web Agent log files, see the *CA SiteMinder Web Access Manager Web Agent Configuration Guide*.

How to Trace Components and Data Fields

When CA Identity Manager integrates with SiteMinder, you can use the SiteMinder Policy Server Profiler to trace components and data fields in the CA Identity Manager extensions for the Policy Server. The Profiler lets you configure filters for the tracing output so that only specific values for a component or data field are captured.

Note: For instructions on using the Policy Server Profiler, see the *CA SiteMinder Web Access Manager Policy Server Administration Guide*.

You can enable tracing for the following components:

Function_Begin_End

Provides low-level trace statements when certain methods in the CA Identity Manager extensions for the Policy Server are executed.

IM_Error

Traces runtime errors in the CA Identity Manager extensions for the SiteMinder Policy Server.

IM_Info

Provides general tracing information for the CA Identity Manager extensions.

IM_Internal

Traces general information about internal CA Identity Manager operations.

IM_MetaData

Provides tracing information when CA Identity Manager processes the directory metadata.

IM_RDB_Sql

Provides tracing information for relational databases.

IM_LDAP_Provider

Provides tracing information for LDAP directories.

IM_RuleParser

Tracks the process of parsing and evaluating member, owner, and admin policies, which are defined in an XML file that gets interpreted at runtime.

IM_RuleEvaluation

Traces the evaluation of member, admin, owner, and scope rules.

IM_MemberPolicy

Traces the evaluation of member policies, including membership and scope.

IM_AdminPolicy

Traces the evaluation of admin policies.

IM_OwnerPolicy

Traces the evaluation of owner policies.

IM_RoleMembership

Traces information relating to role membership, such as the list of roles a user has and the list of members in a certain role.

IM_RoleAdmins

Traces information relating to role administration, such as the list of roles a user can administer and the list of administrators for a certain role.

IM_RoleOwners

Traces information relating to role ownership, such as the list of roles a user owns and the list of owners for a certain role.

IM_PolicyServerRules

Traces evaluation of member rules, such as RoleMember, RoleAdmin, RoleOwner that Policy Server has resolved, and scope rules, such as All and AccessTaskFilter rules for AccessTasks

IM_LLSDK_Command

Traces communication between the internal CA Identity Manager SDK and the policy server. Technical Support uses this trace component.

IM_LLSDK_Message

Traces messages are explicitly sent by Java code to the Policy Server from the internal CA Identity Manager SDK. Technical Support uses this trace component.

IM_IdentityPolicy

Traces the evaluation and application of Identity Policies.

IM_PasswordPolicy

Traces the evaluation of password policies.

IM_Version

Provides information about the CA Identity Manager version.

IM_CertificationPolicy

Traces the evaluation of certification policies.

IM_InMemoryEval

Traces the processing of CA Identity Manager policies, including member, admin, owner, and Identity Policies. Technical Support uses this trace component.

IM_InMemoryEvalDetail

Provides additional detail about the processing of CA Identity Manager policies, including member, admin, owner, and Identity Policies. Technical Support uses this trace component.

The data fields for which you can configure tracing are listed in the *CA SiteMinder Web Access Manager Policy Server Administration Guide*.

Chapter 11: CA Identity Manager Protection

This section contains the following topics:

User Console Security

The User Console is the user interface that lets administrators manage objects like users, groups, and organizations in a CA Identity Manager environment. These objects are assigned with a set of associated roles and tasks. When an administrator logs in to the User Console, tasks that are related to the administrator are displayed in that environment.

By default, CA Identity Manager protects access to the User Console with native authentication. CA Identity Manager administrators enter a valid username and password to log in to a CA Identity Manager environment. CA Identity Manager authenticates the name and password against the user store that CA Identity Manager manages.

If CA Identity Manager integrates with SiteMinder, CA Identity Manager *automatically* uses SiteMinder basic authentication to protect the environment. No additional configuration is required to use basic authentication. You can configure advanced authentication methods using the SiteMinder Administrative User Interface.

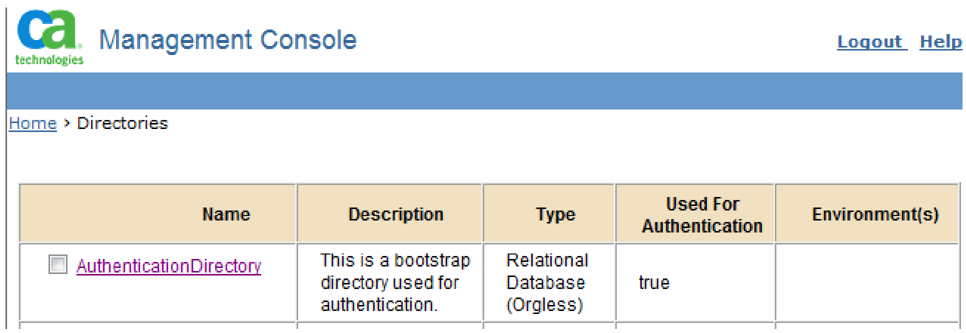
Note: For more information, see the *CA SiteMinder Web Access Manager Policy Server Configuration Guide.*

Management Console Security

The Management Console enables administrators to create and manage CA Identity Manager directories and environments. Administrators can also use the Management Console to configure custom functionality for an environment.

The CA Identity Manager installation includes an option to secure the Management Console. This option is selected by default. During the installation, you specify credentials that CA Identity Manager uses to authenticate an administrator who can access the Management Console. CA Identity Manager creates a user with the credentials you supply in a bootstrap directory named AuthenticationDirectory. You can view this directory in the Management Console.

Note: You cannot use native security to protect the Management Console when CA Identity Manager integrates with CA SiteMinder.



Add Additional Management Console Administrators

By default, a Management Console that is protected by native CA Identity Manager security has one administrator account, which is created in a new CA Identity Manager directory during installation.

To add additional administrators, you specify a CA Identity Manager directory that contains users who need access to the Management Console. Using an existing directory enables you to grant Management Console access to users in your organization, without having to create new accounts.

You can only specify one directory for authentication. You cannot delete a directory while it is configured for authentication.

Follow these steps:

1. Log in to the Management Console with the user credentials you supplied during installation.
2. Open Directories, and click the directory that contains users that require access to the Management Console.

3. Click Update Authentication.
4. Select the Used for Authentication option.
5. Enter the login name for the first user and click Add.
6. Continue adding users that require access to the Management Console until all users have been added. Then, click Save.

   The users that you specified can now use their user name and password to access the Management Console.

Disable Native Security for the Management Console

If you enabled native security for the Management Console, and now want to use a different application to protect it, disable the native security before implementing another security method.

Follow these steps:

1. Disable native security for the Management Console in the web.xml file as follows:
   a. Open *CA Identity Manager_installation*\iam_im.ear\management_console.war\WEB-INF\web.xml in a text editor.
   b. Set the value of the Enable parameter for ManagementConsoleAuthFilter to false as follows:
   ```
   <filter>
   <filter-name>ManagementConsoleAuthFilter</filter-name>
   <filter-class>com.netegrity.ims.manage.filter.ManagementCon
   soleAuthFilter</filter-class>
   <init-param>
   <param-name>Enable</param-name>
   <param-value>false</param-value>
   </init-param>
   </filter>
   ```
   c. Save the web.xml file.
2. Restart the CA Identity Manager server.

   The Management Console is no longer protected by native security.

Use SiteMinder to Secure the Management Console

To protect the Management Console initially, you can create a SiteMinder policy.

A SiteMinder policy identifies a resource that you want to protect, such as the Management Console, and grants a set of users access to that resource.

Follow these steps:

1. **(CA Identity Manager 12.6 or higher only)** Disable native security (see page 259) for the Management Console.
2. Log in to one of the following interfaces as an administrator with Domain privileges:
   - For CA SiteMinder r12 or higher, log in to the Administrative UI.
   - For CA SiteMinder 6.0 SP*x*, log in to the Policy Server User Interface.

   Note: For information on using these interfaces, see the documentation for the version of SiteMinder that you are using.
3. Locate the policy domain for the appropriate CA Identity Manager Environment.

   This domain is created automatically when CA Identity Manager integrates with SiteMinder. The domain name has the following format:

   *Identity Manager-environment*Domain

   In this format, *Identity Manager-environment s*pecifies the name of the environment you are modifying. For example, when the name is *employees*, the domain name is *employees*Domain.
4. Create a realm with the following resource filter:

   /iam/immanage/
5. Create a rule for the realm. Specify an asterisk (*) as the filter to protect all pages in the Management Console.
6. Create new a policy and associate it with the rule you created in the previous step.

   Be sure to associate users who can access the Management Console with the policy.
7. Restart the application server.

Protect an Existing Environment After Upgrade

After you upgrade to CA Identity Manager 12.6 or above, you can protect the Management Console using native security.

Note: You cannot use native CA Identity Manager security to protect the Management Console when CA Identity Manager integrates with CA SiteMinder.

Follow these steps:

1. Enable native security for the Management Console in the web.xml file as follows:
    a. Open *CA Identity Manager_installation*\iam_im.ear\management_console.war\WEB-INF\web.xml in a text editor.
    b. Set the value of the Enable parameter for ManagementConsoleAuthFilter to true as follows:
    ```
    <filter>
    <filter-name>ManagementConsoleAuthFilter</filter-name>
    <filter-class>com.netegrity.ims.manage.filter.ManagementConsoleAuthFilter</filter-class>
    <init-param>
    <param-name>Enable</param-name>
    <param-value>true</param-value>
    </init-param>
    </filter>
    ```
    c. Save the web.xml file.
2. Create the IM_AUTH_USER table in the CA Identity Manager object store.

    The IM_AUTH_USER table stores information about Management Console administrators.

    a. Navigate to CA\Identity Manager\IAM Suite\Identity Manager\tools\db\objectstore
    b. Run one of the following scripts against the object store:
        - sql_objectstore.sql
        - oracle_objectstore.sql

    Note: For information about running a script against an existing database, see the vendor documentation for that database.

3. Use the password tool to encrypt the user password.

   The password tool is installed with the CA Identity Manager tools in the following location:

Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\PasswordTool

UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/tools/PasswordTool

   PasswordTool

   Run the password tool using the following command:
   `pwdtools -JSAFE -p` *`anypassword`*

   The JSAFE option encrypts a plain text value using the PBE algorithm.

1. Insert the bootstrap user information into the IM_AUTH_USER table. Specify values for all columns in the IM_AUTH_USER table.

   For example:

   USER_NAME: admin1

   PASSWORD: *anypassword*

   DISABLED: 0

   ID:1

2. Restart the CA Identity Manager server.

   The Management Console is protected by native security.

Protection from CSRF Attacks

CA Identity Manager is enhanced to improve the resistance to Cross-Site Request Forgery (CSRF) attacks. By default, the enhancement is disabled in CA Identity Manager.

To enable the enhancement:

1. Open the web.xml file located in the following location:

   application-server/iam_im.ear/user_console.war/WEB-INF

2. Find the <context-param> element with <param-name> csrf-prevention-on.
3. Set the <param-value> to true.
4. Restart the application server.

Chapter 12: CA SiteMinder Integration

This section contains the following topics:

SiteMinder and CA Identity Manager

When CA Identity Manager integrates with CA SiteMinder, CA SiteMinder can add the following functionality to a CA Identity Manager environment:

Advanced Authentication

CA Identity Manager includes native authentication for CA Identity Manager Environments by default. CA Identity Manager administrators enter a valid username and password to log in to a CA Identity Manager Environment. CA Identity Manager authenticates the name and password against the user store that CA Identity Manager manages.

When CA Identity Manager integrates with CA SiteMinder, CA Identity Manager uses CA SiteMinder basic authentication to protect the Environment. When you create a CA Identity Manager Environment, a policy domain and an authentication scheme are created in CA SiteMinder to protect that Environment.

When CA Identity Manager integrates with CA SiteMinder, you can also use SiteMinder authentication to protect the Management Console.

Access Roles and Tasks

Access roles enable CA Identity Manager administrators to assign privileges in applications that CA SiteMinder protects. Access roles represent a single action that a user can perform in a business application, such as generating a purchase order in a finance application.

Directory Mapping

An administrator can possibly need to manage users whose profiles exist in a different user store from the one that is used for authenticating the administrator. When logging in to the CA Identity Manager Environment, the administrator is authenticated using one directory and a different directory to authorize the administrator to manage users.

When CA Identity Manager integrates with CA SiteMinder, you can configure a CA Identity Manager Environment to use different directories for authentication and authorization.

Skins for Different Sets of Users

A skin changes the look of the User Console. When CA Identity Manager integrates with CA SiteMinder, you can enable different sets of users to see different skins. To accomplish this change, you use a SiteMinder response to associate a skin with a set of users. The response is paired with a rule in a policy, which is associated with a set of users. When the rule fires, it triggers the response to pass information about the skin to CA Identity Manager, to build the User Console.

Note: For more information, see the *User Console Design Guide*.

Locale Preferences for a Localized Environment

When CA Identity Manager integrates with CA SiteMinder, you can define locale preference to a user using an imlanguage HTTP header. In the SiteMinder Policy Server, you set this header within a SiteMinder response and specify a user attribute as value of the header. This imlanguage header acts as the highest priority locale preference for a user.

Note: For more information, see the *User Console Design Guide*.

More Information:

Collect User Credentials Using a Custom Authentication Scheme (see page 316)

How Resources are Protected

Advanced authentication requires you to use a SiteMinder Policy Server in your implementation. The application server hosting the CA Identity Manager Server is on a different operating environment from Web Server. To provide forwarding services, the Web Server requires:

- An application server vendor provided plug-in.
- A SiteMinder agent to protect the CA Identity Manager resources, such as the User Console, Self Registration, and the Forgotten Password feature.

The Web Agent controls the access of users who request CA Identity Manager resources. Once the users are authenticated and authorized, the Web Agent allows the Web Server to process the requests.

When the Web Server receives the request, the application server plug-in forwards it to the application server hosting the CA Identity Manager Server.

The Web Agent protects CA Identity Manager resources that are exposed to users and administrators.

Overview of SiteMinder and CA Identity Manager Integration

When the policy administrator and the identity administrator work together to integrate SiteMinder into an existing CA Identity Manager installation, the CA Identity Manager architecture expands to include the following components:

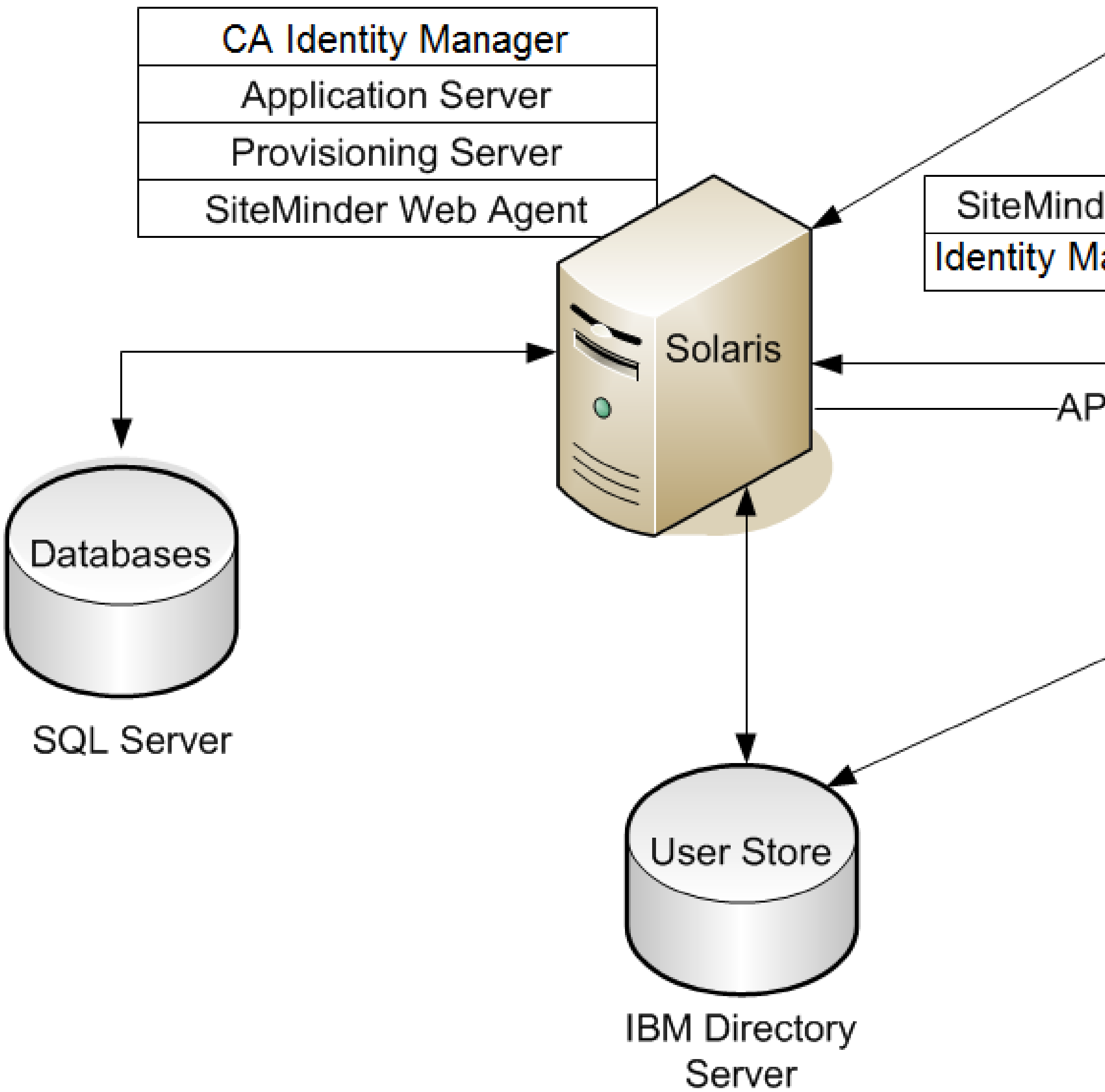
SiteMinder Web Agent

Protects the CA Identity Manager Server. The Web Agent is installed on the system with the CA Identity Manager Server.

SiteMinder Policy Server

Provides advanced authentication and authorization for CA Identity Manager.

The following figure is an example of a CA Identity Manager installation with a SiteMinder Policy Server and Web Agent:

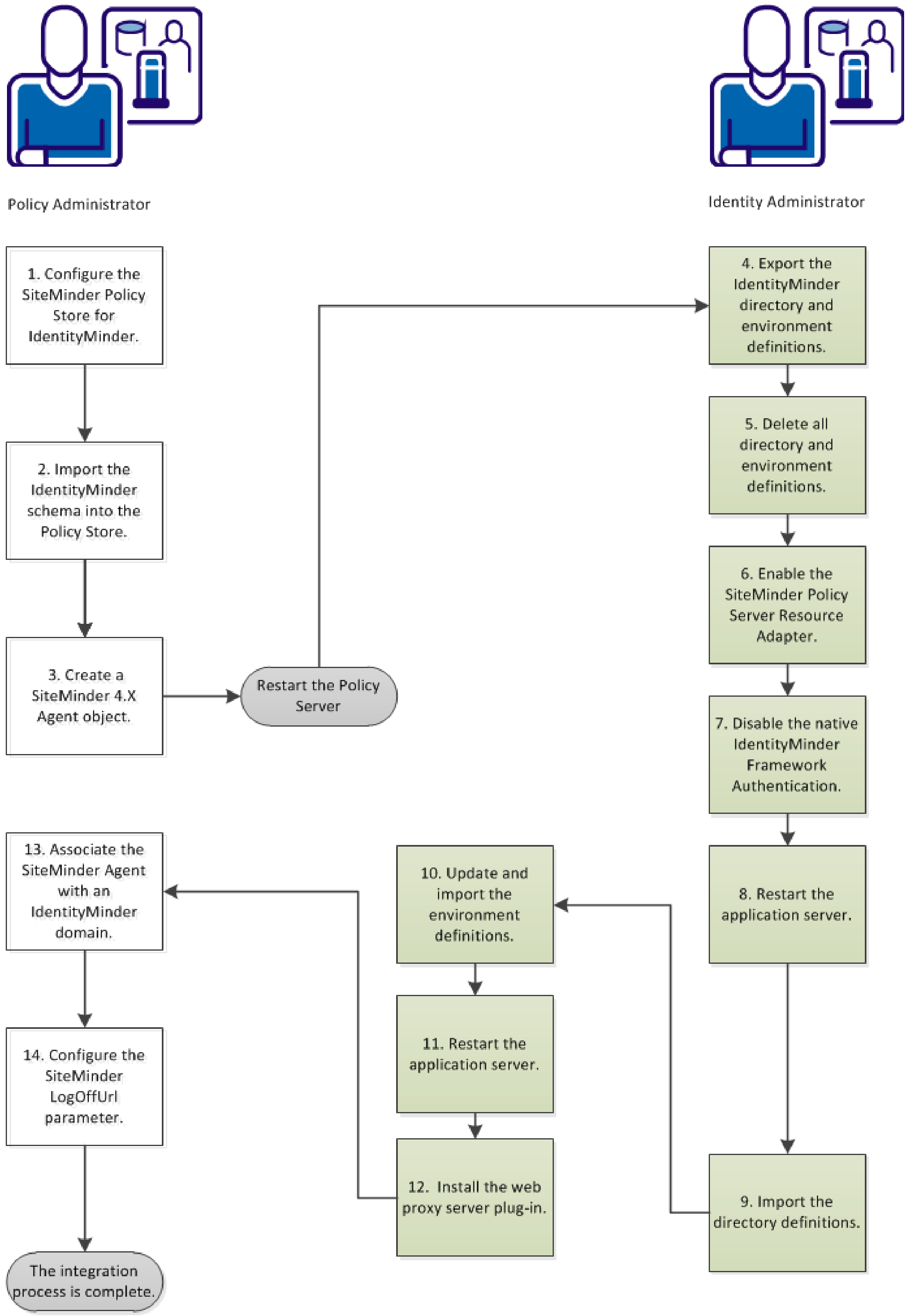

CA Identity Manager
Application Server
Provisioning Server
SiteMinder Web Agent
Solaris
SiteMind
Identity Ma
AP
Databases
SQL Server
User Store
IBM Directory
Server

Note: The components are installed on different platforms as examples. However, you can choose other platforms. The CA Identity Manager databases are on Microsoft SQL Server and the user store is on the IBM directory Server. The SiteMinder Policy Store is on AD LDS on Windows.

Completing this process requires two roles: the CA Identity Manager identity administrator and the SiteMinder policy administrator. In some organizations, one person fills both roles. When two people are involved, close collaboration is required to complete the procedures in this scenario. The policy administrator begins and ends this process; the identity administrator does all the steps in the middle.

Important! For CA Identity Manager installations starting with Release12.5 SP7, the Java Cryptography Extension Unlimited Strength Jurisdiction Policy Files (JCE libraries) are required. Download these libraries from the Oracle Web site. Load them into the following folder: <Java_path>\<jdk_version>\jre\lib\security\.

The following diagram illustrates the complete process of integrating SiteMinder into CA Identity Manager:


Policy Administrator
Identity Administrator
1. Configure the SiteMinder Policy Store for IdentityMinder.
2. Import the IdentityMinder schema into the Policy Store.
3. Create a SiteMinder 4.X Agent object.
Restart the Policy Server
4. Export the IdentityMinder directory and environment definitions.
5. Delete all directory and environment definitions.
6. Enable the SiteMinder Policy Server Resource Adapter.
7. Disable the native IdentityMinder Framework Authentication.
8. Restart the application server.
9. Import the directory definitions.
10. Update and import the environment definitions.
11. Restart the application server.
12. Install the web proxy server plug-in.
13. Associate the SiteMinder Agent with an IdentityMinder domain.
14. Configure the SiteMinder LogOffUrl parameter.
The integration process is complete.

Follow these steps:

1. Configure the SiteMinder Policy Store for CA Identity Manager. (see page 273)
2. Import the CA Identity Manager Schema into the Policy Store. (see page 278)
3. Create a SiteMinder 4.X agent object. (see page 279)
4. Export the CA Identity Manager directories and environments. (see page 280)
5. Delete all directory and environment definitions. (see page 280)
6. Enable the SiteMinder Policy Server Resource Adapter. (see page 281)
7. Disable the native CA Identity Manager Framework Authentication Filter. (see page 282)
8. Restart the application server. (see page 282)
9. Configure a data source for SiteMinder. (see page 283)
10. Import the directory definitions. (see page 283)
11. Update and import environment definitions. (see page 284)
12. Restart the application server. (see page 282)
13. Install the web proxy server plug-in. (see page 284)
14. Associate the SiteMinder Agent with an CA Identity Manager domain. (see page 302)
15. Configure SiteMinder LogOffUrl Parameter. (see page 303)

Configure the SiteMinder Policy Store for CA Identity Manager

As a policy administrator you use the CA Identity Manager Administrative Tools to access the SQL scripts or LDAP schema text to add the IMS schema to the policy store. The identity administrator will have installed these tools in the Admin Tools folder. Follow *one* of the following procedures to configure the policy store:

Configure a Relational Database (see page 273)

Configure Sun Java Systems Directory Server or IBM Directory Server (see page 274)

Configure Microsoft Active Directory (see page 274)

Configure Microsoft ADAM (see page 275)

Configure CA Directory Server (see page 276)

Configure Novell eDirectory Server (see page 277)

Configure Oracle Internet Directory (OID) (see page 278)

Configure a Relational Database

After configuration, you can use your relational database as a SiteMinder policy store.

Follow these steps:

1. Configure the database as a supported SiteMinder policy store.

   Note: For configuration instructions, see the *SiteMinder Policy Server Installation Guide.*

2. Run the appropriate script for your database:

   - **SQL:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\policystore-schemas\MicrosoftSQLServer\ims8_mssql_ps.sql
   - **Oracle:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools/policystore-schemas/OracleRDBMS/ims8_oracle_ps.sql

   The preceding paths are default installation locations. The location for your installation may be different.

Configure Sun Java Systems Directory Server or IBM Directory Server

To configure a Java or IBM directory server you apply the appropriate schema file.

Follow these steps:

1. Configure the directory as a supported SiteMinder policy store.

   Note: For configuration instructions, see the *CA SiteMinder Policy Server Installation Guide.*

2. Add the appropriate LDIF schema file to the directory. The default Windows location for the LDIF files is C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\policystore-schemas.

   Adding the following schema files for your directory:

   - **IBM Directory Server:**

     IBMDirectoryServer\V3.identityminder8

   - **Sun Java Systems Directory Server (iPlanet):**

     SunJavaSystemDirectoryServer\sundirectory_ims8.ldif

Configure Microsoft Active Directory

To configure a Microsoft Active Directory policy store, you apply the activedirectory_ims8.ldif script.

Follow these steps:

1. Configure the directory as a supported SiteMinder policy store.

   Note: For configuration instructions, see the *CA SiteMinder Policy Server Installation Guide.*

2. Modify the activedirectory_ims8.ldif schema file as follows:

   a. In a text editor, open the activedirectory_ims8.ldif file. The default Windows location is:

      C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\policystore-schemas\MicrosoftActiveDirectory

   b. Replace all instances of {root} with the root organization for the directory.

      The root organization must match the root organization that you specified when you configured the policy store in the Policy Server Management Console.

For example, if the root is dc=myorg,dc=com, replace
dn: CN=imdomainid6,CN=Schema,CN=Configuration,{root} with dn:
CN=imdomainid6,CN=Schema,CN=Configuration,dc=myorg,dc=com

c. Save the file.

3. Add the schema file as described in the documentation for your directory.

Configure Microsoft ADAM

To configure a Microsoft ADAM policy store, you apply the adam_ims8.ldif script.

Follow these steps:

1. Configure the directory as a supported SiteMinder policy store.

   Note: For configuration instructions, see the *CA SiteMinder Policy Server Installation Guide.*

   Make note of the CN value (the guid).

2. Modify the adam_ims8.ldif schema file as follows:

   a. Open the adam_ims8.ldif\.ldif file in a text editor. The default Windows location is:

      C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\policystore-schemas\MicrosoftActiveDirectory

   b. Replace every cn={guid} reference with the string you found when you configured the SiteMinder policy store in Step 1 of this procedure.

      For example, if the guid string is CN={39BC711D-7F27-4311-B6C0-68FDEE2917B8}, then replace every cn={guid} reference with CN={39BC711D-7F27-4311-B6C0-68FDEE2917B8}.

   c. Save the file.

3. Add the schema file as described in the documentation for your directory.

Configure CA Directory Server

To configure a CA Directory server you create a custom schema file. In the steps that follow, *dxserver_home* is the directory where CA Directory is installed. The default source location for this file on Windows is C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\policystore-schemas\eTrustDirectory.

Follow these steps:

1. Configure the directory as a supported SiteMinder policy store.

   Note: For configuration instructions, see the *CA SiteMinder Policy Server Installation Guide.*

2. Copy etrust_ims8.dxc to *dxserver_home*\config\schema.

3. Create a custom schema configuration file as follows:

   a. Copy the *dxserver_home*\config\schema\default.dxg to *dxserver_home*\config\schema*company_name*-schema.dxg.

   b. Edit the *dxserver_home*\config\schema*company_name*-schema.dxg file by adding the following lines to the bottom of the file:

   ```
   # Identity Manager Schema
   source "etrust_ims8.dxc";
   ```

4. Edit the *dxserver_home*\bin\schema.txt file by adding the contents of etrust_ims_schema.txt to the end of the file. The default source location for this file on Windows is C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\policystore-schemas\eTrustDirectory.

5. Create a custom limits configuration file as follows:

   a. Copy the *dxserver_home*\config\limits\default.dxc to *dxserver_home*\config\limits*company_name*-limits.dxc.

   b. Increase the default size limit to 5000 in the *dxserver_home*\config\limits*company_name*-limits.dxc file as follows:

   ```
   set max-op-size=5000
   ```

   Note: Upgrading CA Directory overwrites the limits.dxc file. Therefore, make sure that you reset max-op-size to 5000 after the upgrade is completed.

6. Edit the *dxserver_home*\config\servers*dsa_name*.dxi as follows:

   ```
   # schema
   source "company_name-schema.dxg";

   #service limits
   source "company_name-limits.dxc";
   ```

   where *dsa_name* is the name of the DSA using the customized configuration files.

7. Run the dxsyntax utility.

8. Stop and restart the DSA as the dsa user to make the schema changes take effect, as follows:

   ```
   dxserver stop dsa_name
   dxserver start dsa_name
   ```

Configure Novell eDirectory Server

To configure a Novell eDirectory Server policy store, you apply the novell_ims8.ldif script.

Follow these steps:

1. Configure the directory as a supported SiteMinder policy store.

   Note: For configuration instructions, see the *CA SiteMinder Policy Server Installation Guide.*

2. Find the Distinguished Name (DN) of the NCPServer for your Novell eDirectory Server by entering the following information in a command window on the system where the Policy Server is installed:

   ```
   ldapsearch -h hostname -p port -b container -s sub
   -D admin_login -w password objectClass=ncpServer dn
   ```

   For example:

   ```
   ldapsearch -h 192.168.1.47 -p 389 -b "o=nwqa47container" -s sub -D
   "cn=admin,o=nwqa47container" -w password objectclass=ncpServer dn
   ```

3. Open the novell_ims8.ldif file.

4. Replace every NCPServer variable with the value you found in Step 2.

   The default location for novell_ims8.ldif on Windows is:

   C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\policystore-schemas\NovelleDirectory

   For example, if the DN value is cn=servername,o=servercontainer, you would replace every instance of *NCPServer* with cn=servername,o=servercontainer.

5. Update the eDirectory Server with the novell_ims8.ldif file.

   See the Novell eDirectory documentation for instructions.

Configure Oracle Internet Directory (OID)

To configure an Oracle Internet Directory you update the oracleoid ldif file.

Follow these steps:

1. Configure the directory as a supported SiteMinder policy store.

   Note: For configuration instructions, see the *CA SiteMinder Policy Server Installation Guide.*

2. Update the Oracle Internet Directory Server with the oracleoid_ims8.ldif file. The default installation location for this file on Window is:

   install_path\policystore-schemas\OracleOID\

   See the Oracle Internet Directory documentation for instructions.

Verify the Policy Store

To verify the policy store, confirm the following points:

- Your Policy Server log does not contain a section of warnings that begins with the following code:
  ```
  *** IMS NO SCHEMA BEGIN
  ```
  This warning appears only if you have installed the Extensions for the SiteMinder Policy Server, but you have not extended the policy store schema.

- The CA Identity Manager objects exist in the policy store database or directory. The CA Identity Manager objects begin with an ims prefix.

Import the CA Identity Manager Schema into the Policy Store

The policy administrator imports the CA Identity Manager schema into the policy store. This task lets CA Identity Manager create, update, and delete policy objects. Examples include directory objects, domains, realms, rules, policies, and the policy objects that enable access roles and tasks.

Follow these steps:

1. On the SiteMinder Policy Server, shut down the Policy Server service.
2. Run the CA Identity Manager installer for the version that you are using.
3. When asked which components to install, select the Extensions for SiteMinder (if SiteMinder is installed locally).
4. Verify that the Policy Server service is restarted before continuing.

Create a SiteMinder 4.x Agent Object

The policy administrator creates a SiteMinder 4.x Web Agent. This task enables communication between SiteMinder and CA Identity Manager. The identity administrator references this agent during the CA Identity Manager configuration.

Follow these steps:

1. Log on to the SiteMinder Administrative UI.

   The relevant tabs for your administrator privileges appear.

2. Click Infrastructure, Agents, Agent, Create Agent.

   The Create Agent dialog appears.

3. Select Create a new object of type Agent, and then click OK.

   The Create Agent dialog appears.

4. Enter a name and an optional description.

   Note: Use a name that you can easily associate with the corresponding SharePoint Connection Wizard.

5. Select SiteMinder.
6. Select Web Agent from the drop-down list.
7. Enable 4.x functionality with the following steps:

   a. Select the Supports 4.x agents check box.

      The trust settings fields appear.

   b. Add the trust settings by completing the following fields:

      IP Address

      Specifies the IP Address of the Policy Server.

      Shared Secret

      Specifies a password that is associated with the 4.x Agent object. The SharePoint Connection Wizard also requires this password.

      Confirm Secret

      Confirms a password that is associated with the 4.x Agent object. The SharePoint Connection Wizard also requires confirmation of this password.

8. Click Submit.

   The Create Agent Object task is submitted for processing and the confirmation message appears.

Export the CA Identity Manager Directories and Environments

The integration process removes all of the current environment and directory definitions. To help ensure that this information is maintained, the identity administrator exports the environments using the CA Identity Manager Management Console. After you complete the integration, these definitions restore the directories and environments.

Follow these steps:

1. Open the CA Identity Manager Management Console.
2. Click Directories.
3. Click the first directory in the list and click Export.
4. Save and archive the directory xml file.
5. Repeat this process for the remaining directories.
6. Click Home, and then click Environments.
7. Select the first environment.
8. Click Export.
9. Repeat this process for the remaining environments.

   Note: This process can take a few minutes for each environment.

Delete All Directory and Environment Definitions

To prepare for SiteMinder to protect CA Identity Manager, the identity administrator deletes the directory and environment definitions using the CA Identity Manager Management Console.

Follow these steps:

1. Open the CA Identity Manager Management Console.
2. Click Environments.
3. Select the first environment
4. Click Delete.
5. Repeat this process for each of your remaining environments.

   Note: Delete your environments before deleting your directories because the environments reference the directories.

6. Navigate back to the Directories section.
7. Select all of the directories listed.
8. Click Delete.

Enable the SiteMinder Policy Server Resource Adapter

The identity administrator enables the SiteMinder Policy Server Resource Adapter. The purpose of the adapter is to validate the SMSESSION cookie. After validation, SiteMinder creates the user context.

Follow these steps:

1. Navigate to the \policysever.rar\META-INF folder located within the iam_im.ear on the application server that is running CA Identity Manager.
2. Open the ra.xml file in an editor.
3. Search for the Enabled config-property, and then change the config-property-value to true as shown in the following example:

```
        <config-property-name>validateSmHeadersWithPS</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>true</config-property-value>
    </config-property>
    <config-property>
        <config-property-name>Enabled</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>true</config-property-value>
    </config-property>
    <!-- Set FIPS Mode to true if SiteMinder is in FIPS Only Mode -->
    <config-property>
        <config-property-name>FIPSMode</config-property-name>
```

4. Locate the ConnectionURL property and provide the hostname of the SiteMinder Policy Server. Use a fully qualified domain name (FQDN).
5. Locate the UserName property and specify the account to use for communication with SiteMinder. SiteMinder is the default value for this account.
6. Locate the AdminSecret property. Provide the encrypted password. Copy the password from the directory.xml file that you exported and paste it into ra.xml. If you are not sure that you have a common password, encrypt your password using the CA Identity Manager Password Tool.
7. Paste the encrypted password into the ra.xml file.
8. Specify the 4.x agent name that the policy administrator created during the SiteMinder configuration.
9. Specify the encrypted password. Use the Password Tool to encrypt the password if necessary.
10. Save the changes to the ra.xml file.

The SiteMinder Policy Server Resource Adapter is enabled.

More information:

Modify a SiteMinder Password or Shared Secret (see page 334)

Disable the Native CA Identity Manager Framework Authentication Filter

With the SiteMinder adapter in place, the Framework Authentication Filter is no longer needed. The identity administrator can disable the filter.

Follow these steps:

1. Locate and edit the web.xml file in the \user_console.war\WEB-INF folder under the iam_im.ear.
2. Locate the FrameworkAuthFilter and switch the value of the Enable init-param to false.

   If you are using CA Identity Manager r12.5 SP7 or later, verify that the Java Cryptographic Extension Unlimited Strength Jurisdiction Policy Files (JCE) are downloaded into \<Java_path>\<jdk_version>\jre\lib\security in the CA Identity Manager environment. These files enable CA Identity Manager to connect to SiteMinder.

   If the JCE libraries are installed, you see the following messages during CA Identity Manager application startup:

   ```
   2012-07-06 11:23:56,079 WARN  [ims.default] (main) * Startup Step 2 : Attempting
   to start PolicyServerService
   2012-07-06 11:23:56,081 WARN [ims.default] (main) Unlimited Strength Java Crypto
   Extensions enabled: TRUE
   ```

   Otherwise, the value is false for the "Unlimited Strength Java Crypto Extensions enabled" entry. CA Identity Manager fails to connect to the Policy Server.

Restart the Application Server

The restart refreshes the application server with the changes. The identity administrator validates that the switch was successful and that a proper connection to the SiteMinder Policy Server exists.

Follow these steps:

1. Use the services panel to restart CA Identity Manager when your application server is running as a service.
2. Refer to the server.log to validate the connection

Configure a Data Source for SiteMinder

If your CA Identity Manager environment uses a relational database for its identity store, the identity administrator is required to complete an additional process on the SiteMinder Policy Server. SiteMinder requires a local data source to communicate with the database.

Follow these steps:

1. For Windows servers, open the ODBC Data Source Administrator console that is found under Administrative Tools.
2. Click the System DSN tab.
3. Click Add and select the corresponding SiteMinder driver for your database.
4. Provide the needed information to reference the relational database user store.
5. Test the connectivity before you continue.

Import the Directory Definitions

To prepare for importing the environments, the identity administrator imports the directories that the environments reference. Importing the directory definition in CA Identity Manager also adds the directory information to the SiteMinder policy store.

Follow these steps:

1. Ensure CA Identity Manager is running and connected to SiteMinder,
2. Navigate to the CA Identity Manager Management Console.
3. Click Directories and then click Create or Update from XML.
4. Select your directory configuration file (directory.xml). This file is the one that you exported in Export the CA Identity Manager Directories and Environments (see page 280).
5. Click Next.
6. Click Finish and review the load output. Verify that the directory is present in CA Identity Manager and SiteMinder.
7. Repeat these steps for the Provisioning Store and any remaining directories.
8. Log in to the SiteMinder Administrative UI to validate the creation of the user directories.

Update and Import Environment Definitions

The identity administrator imports the updated environments back into CA Identity Manager.

Follow these steps:

1. Unlike the directory exports, the environment export is in the form of a zip file. Drag a copy of the *name*.xml file out of the zipfile.
2. Copy the *name*.xml file. Insert a reference to the protecting agent (not the SM 4.x agent) at the end of the ImsEnvironment element, before the enclosing /> bracket: *agent="idmadmin"*
3. Save and paste the file back into the zip file.
4. Open the CA Identity Manager Management Console and click Environments and then Import.
5. Enter the name of the updated environment zip file.
6. Click Finish and review the import output.
7. Repeat this process for all your remaining environments.
8. Restart the Application Server.

Install the Web Proxy Server Plug-in

Based on which application is installed, the identity administrator installs one of the following plug-ins that the web server uses to forward requests to the application server:

- WebSphere (see page 285)
- JBoss (see page 291)
- WebLogic (see page 295)

Install the Proxy Plug-In on WebSphere

The web server on which you installed the web agent forwards request requests to the application server that hosts the CA Identity Manager server. The vendor-provided web server proxy plug-in provides this service.

Use the procedures that are applicable for your deployment:

1. Configure the IBM HTTP Server (see page 285) (All web servers)
2. Configure the Proxy Plug-In (see page 286) (All web servers)
3. One of the following:
   - Complete the Configuration on IIS (see page 289)
   - Complete the Configuration on iPlanet or Apache (see page 291)

Configure the IBM HTTP Server

For all web servers, you install the proxy plug-in and use the configurewebserver command.

Follow these steps:

1. Install the proxy plug-in from the WebSphere Launch Pad.
2. Add the web server to the WebSphere cell by running the configurewebserver1.bat command as follows:
   a. Edit *websphere_home*\Plugins\bin\configurewebserver1.bat/,sh in a text editor.
   b. Add a user name and password to after wsadmin.bat/.sh as follows:
   `wsadmin.bat -user wsadmin -password` *`password`* `-f configureWebserverDefinition.jacl`
   c. Run configurewebserver1.bat/.sh.

   Note: See the IBM WebSphere documentation for more information about the configurewebserver command.
3. Continue with the procedure to Configure the Proxy Plug-In (see page 286).

Configure the Proxy Plug-In

For all web servers, you update the plug-in using the GenPluginCfg command of WebSphere:

Follow these steps:

1. Log in to the system where WebSphere is installed.
2. From the command line, navigate to *websphere_home*\bin, where *websphere_home* is the installed location of WebSphere.

   For example:

   - **Windows:**

     C:\Program Files\WebSphere\AppServer\profile\AppSrv01\bin

   - **UNIX:**

     /*home_dir*/WebSphere/AppServer/profile/AppSrv01/bin

3. Run the GenPluginCfg.bat or GenPluginCfg.sh command.

   Running this command generates a plugin-cfg.xml file in the following location:

   websphere_home\AppServer\profiles\AppSrv01\config\cells

4. Continue with one of the following procedures:

   - Complete the Configuration on IIS (see page 289)
   - Complete the Configuration on iPlanet or Apache (see page 291)

Complete the Configuration on IIS (7.x)

Before starting this procedure, verify that you are using a Version 6.1.0.9, or later, of the web server plug-in. Earlier versions of the plug-in do not support the Windows Server 2008 operating system.

Follow these steps:

1. Install IIS Version 7.x with the IIS Version 6.0 Management Compatibility components. IIS Version 6.0 Management Compatibility components are not installed by default.
2. Complete the following steps to bring up the Server Manager window on Windows Server 2008:
    1. Click Start, Administrative Tools, Server Managers.
    2. Click Action, Add Roles, and then click Next.
    3. Select the Web Server (IIS) role on the Select Server Roles page, and then click Next.
    4. Click Add Feature, Next, when a prompt for the Windows Process Activation Service feature displays
    5. Click Next on the IIS introduction page.
3. When the Role Services window displays, verify that the following options are selected in addition to the default options that are already selected.
    - Internet Information Services: Management Tools
    - IIS Version 6.0 Management Compatibility: IIS Version 6.0 Management Console, IIS Version 6.0 Scripting Tools, IIS Version 6.0 WMI Compatibility, and IIS Metabase compatibility
    - Application Development: ISAPI Extensions, ISAPI Filters
4. Click Next to enable the selected options, and then click Install on the next window to perform the installation.
5. Click Close on the Installation Results window when the installation finishes.
6. Open the Command Prompt and go to :\Program Files\IBM\WebSphere\AppServer\profiles\Dmgr01\bin.
7. Run this command: GenPluginCfg.bat.

    The plugin-cfg.xml file will be generated at this location: C:\Program Files\IBM\WebSphere\AppServer\profiles\Dmgr01\config\cells.
8. Create a directory under c:\, for example, c:\plugin.
9. Copy the plugin-cfg.xml file to the c:\plugin directory.
10. Copy iisWASPlugin_http.dll file to the c:\plugin directory.

11. Select Start , All Programs, Administrative Tools, Internet Information Services (IIS) Manager on a Windows Server 2008 operating system. This action starts the IIS application and creates a new virtual directory for the Web site instance. These instructions assume that you are using the Default Web Site.
12. Expand the tree on the left until you see Default Web Site.
13. Right-click Default Web Site,Add Virtual Directory to create the directory with a default installation.
14. Enter setPlugins in the Alias field on the Virtual Directory Alias window of the Virtual Directory Creation Wizard.
15. Browse to the c:\plugin directory in the Physical Path field of the Web Site Content Directory window of the wizard, and then click OK.
16. Click the Test Settings button. If the settings test fails, you can change the permissions of the physical directory. Alternatively, select Connect As, and let IIS connect as a Windows user account that has authority to files in that physical path.
17. Click OK to add the setPlugins virtual directory to your web site.
18. Select the setPlugins virtual directory that you just created in the navigations tree.
19. Double-click Handler Mappings, and then click Edit Feature Permissions on the Actions panel.
20. Select Script and Execute, if they are not already selected.
21. Click OK.
22. Return to the IIS Manager window, and expand the Web Sites folder in the left-hand navigation tree of that window.
23. Select Default Web Site in the navigation tree.
24. Complete the following steps on the Default Web Site Properties panel to add the ISAPI filter:
    1. Double-click the ISAPI Filters tab.
    2. Click to open the Add/Edit Filter Properties dialog.
    3. Enter iisWASPlugin in the Filter name field.
    4. Click Browse to select the plug-in file located in the c:\plugin\iisWASPlugin_http.dll directory.
    5. Click OK to close the Add/Edit Filter Properties dialog.
25. Select the top level server node in the navigation tree.
26. Double-click ISAPI and CGI Restrictions on the Features panel.

    To determine the value to specify for the ISAPI or CGI Path property, browse to, and then select the same plug-in file that you selected in the previous step. For example:c:\plugin\iisWASPlugin_http.dll.

27. Click Add on the Actions panel.
28. Enter WASPlugin in the Description field, select Allow extension path to execute, and then click OK to close the ISAPI and CGI Restrictions dialog window.
29. Create the new file plugin-cfg.loc in location c:\plugin. Set the value in the plugin-cfg.loc file to the location of the configuration file. The default location is C:\plugin\plugin-cfg.xml.

Update the Web Agent

After configuring IIS 7.x, make the following changes to the web agent:

1. Click Application pools and change the Default App Pool to Classic mode.
2. Click Submit.
3. Make sure the agent is higher in the ISAPI Filters priority list than the plug-in for the application server used by CA Identity Manager.
4. Restart IIS Version 7.x and your WebSphere Application Server profile.

Complete the Configuration on IIS

After you have configured the IBM HTTP server and the proxy plug-in, you make sure the proxy plugin-cfg.xml is in the right location and perform steps to configure an additional plugin file.

Follow these steps:

1. Copy the plugin-cfg.xml as follows:
   a. Log in to the system where the web agent is installed.
   b. Create a folder with no spaces under the C: drive. For example: C:\plugin.
   c. Copy the plugin-cfg.xml file to the C:\plugin folder.
2. Create a file called plugin-cfg.loc in the C:\plugin folder and add the following line into the file:

   C:\plugin\plugin-cfg.xml

3. Download the Websphere Plugin installer from www.ibm.com to the system where WebSphere is installed.

4. Go to the location of the WebSphere Plugin installer.

5. Generate the iisWASPlugin_http.dll file by using this command:

   ```
   install is:javahome "c:\IBM\WebSphere\AppServer\Java
   ```

   Respond to the questions presented based on your configuration.

   When the wizard ends, the iisWASPlugin_http.dll file is saved in the C:\IBM\WebSphere\Plugs\bin folder. Look for a 32-bit or 64-bit subfolder.

6. Copy the iisWASPlugin_http.dll file to the C:\plugin folder on the system with the web agent.

7. Create a virtual directory as follows:

   a. Open the IIS Manager.

   b. Right-click Default web sites.

   c. Click New virtual directory and supply these values:

      Alias: sePlugins (it is case-sensitive.)

      Path: c:\plugin

      Permission: Read + Execute (ISAPI or CGI)

8. Add an ISAPI filter as follows:

   a. Right-click Default Web Site.

   b. Click properties.

   c. Click Add on the ISAPI filter tab.

   d. Supply these values:

      Filter name: sePlugins

      Executable: c:\plugin\ iisWASPlugin_http.dll

9. Create a web service extension as follows:

   a. In IIS6 Manager, expand the computer name.

   b. Create a Web Service Extension and set it to allowed.

      Extension name: WASPlugin

      Path: C:\plugin\ iisWASPlugin_http.dll

   c. Right click each Web Service Extension to change it to Allowed Status.

10. Restart the IIS Web server.

    In the master WWW service, ensure that the WebSphere plug-in (sePlugin) appears after the SiteMinder Web Agent plug-in and that the WebSphere plug-in started successfully.

Complete the Configuration on iPlanet or Apache

After you have configured the IBM HTTP server and the proxy plug-in, you make sure the proxy plugin-cfg.xml is in the right location and restart the web server.

Follow these steps:

1. Copy the plugin-cfg.xml from the system where you installed the proxy plug-in to the following location:

   websphere_home\AppServer\profiles*server_name*\config\cells*websphere_cell*\nodes\webserver1_node\servers\webserver1\

2. Ensure that the WebSphere plug-in (libns41_http.so) is loaded after the SiteMinder Web Agent plug-in (NSAPIWebAgent.so) on all iPlanet Web Servers

3. Check the order of plug-ins in *iplanet_home*/https-*instance*/config/magnus.conf for IPlanet 6.0 Web Servers.

4. Copy the following lines from *iplanet_home*/https-*instance*/config/magnus.conf to *iplanet_home*/https-*instance*/config/obj.conf (IPlanet 5.x Web Serverse):

   Init fn="load-modules" funcs="as_init,as_handler,as_term" shlib="/export/WebSphere/AppServer/bin/libns41_http.so"

   Init fn="as_init" bootstrap.properties="/export/WebSphere/AppServer/config/cells/plugin-cfg.xml"

   Add the following code after AuthTrans fn="SiteMinderAgent" in the obj.conf file:

   Service fn="as_handler"

5. Be sure that the SiteMinder Web Agent plug-in (mod2_sm.so) is loaded before the WebSphere plug-in (mod_ibm_app_server_http.so) on Apache Web Servers. This command is in the Dynamic Shared Object (DSO) Support section of *apache_home*/config/httpd.conf,

6. Restart the web server.

Install the Proxy Plug-In for JBoss

After the SiteMinder web agent authenticates and authorizes a request for a CA Identity Manager resource, the web server forwards the request to the application server that hosts the CA Identity Manager server. To forward these requests, install and configure a JK Connector on the system where the SiteMinder Web Agent installed. See the following Jakarta Project web site for more information about the JK Connector:

http://community.jboss.org/wiki/usingmodjk12withjboss

The CA Identity Manager Administrative Tools include sample configuration files that you can use to configure the JK Connector. For instructions, see the readme.txt file in the directory noted in the following table:

Platform	Location
IIS Web server on a Windows system	C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\samples\ConnectorConfiguration\windows\IIS_JBoss*
Sun Java System Web server on a Solaris system	/opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools/samples/ConnectorConfiguration/solaris/Iplanet_JBoss*
Apache Web server on a Solaris system	/opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools/samples/ConnectorConfiguration/solaris/apache_JBoss*

Install and Configure a JBoss Application Plug-in (IIS 7.x)

This procedure describes the configuration of the JBoss Apache Plug-in starting with IIS 7.0

Follow these steps:

1. Deploy and Update ISAPI filters on the File System.

   Deploy the ISAPI folder to the root of the C drive.

2. Edit the jakarta.reg file located in the unzipped folder.

   If you placed the ISAPI folder on the root of C:\, do not change this file. If you placed it in a different folder, specify that folder on lines 9, 11,and 12.

3. Save your changes and then double-click to update your registry.
4. Edit the workers.properties file by specifying the location of your JBoss application server. The port and type do not need to change.
5. Install IIS7 or IIS7.5 on Windows 2008.
6. Open the System Manager and verify that IIS ISAPI filter and ISAPI Extension are installed.
7. Start inetmgr in Run window.
8. Select the m/c name and double click ISAPI and CGI Restriction.
9. Click the Add button on the right-side panel.
10. Add ISAPI or CGI Restrictions window appears.
11. Select isapi_redirect.dll and enter the description as ISAPI.

12. Select Allow Extension path to execute.
13. Click OK in the Add ISAPI or CGI Restrictions window.
14. Expand the Sites in the Connection section, select the Default WebSite, and right click on Add Virtual Directory.
15. Enter the alias as "jakarta" and enter the location of the isap_redirect.dll file(c:\ajp) in the physical path.
16. Click the Test Settings button:
    - If Authentication and Authorization passed, click OK.
    - If Authorization fails, click the Connect As button.
17. Select the Specific user and provide the Admin User Name and password.
18. Click again the Test Settings button. This time the authorization passes.
19. Click the Default Web Site at Left and double click the ISAPI filter.
20. Click the Add button on the right-side panel.
21. Enter the name and provide the location of the isapi_redirect.dll file.
22. Click OK.
23. Expand the Default WebSite and click on the jakarta virtual directory.
24. Double click Handler Mapping.
25. Select the ISAPI-dll and click Edit Feature Permission.
26. Verify that all permissions (Read, Script, Execute) are selected.
27. Click OK.

Update the Web Agent

After configuring IIS 7.x, make the following changes to the web agent:

1. Click Application pools and change the Default App Pool to Classic mode.
2. Click Submit.
3. Make sure the agent is higher in the ISAPI Filters priority list than the plug-in for the application server used by CA Identity Manager.

The JBoss plug-in is configured.

Install and Configure a JBoss Application Plug-in (IIS 6.0)

This integration assumes that SiteMinder authenticates and authorizes a user before reaching CA Identity Manager. A user is required to have an SMSESSION cookie before reaching CA Identity Manager. Use an application plug-in (proxy redirection) protected by a SiteMinder Web Agent. Through this configuration, a user is authenticated by SiteMinder and then redirected to CA Identity Manager after an SMSESSION cookie has been created.

This procedure is for the deployment and configuration of the JBoss Apache Plug-in for IIS 6.0:

Follow these steps:

1. Deploy and Update ISAPI filter on the File System.

   Be sure to deploy the ISAPI folder to the root of the C drive.

2. Edit the jakarta.reg file located in the unzipped folder.

   If you placed the ISAPI folder on the root of C:\, do not change this file. If you place it in a different folder, specify that folder on lines 9, 11,and 12.

3. Save your changes and then double-click to update your registry.
4. Edit the workers.properties file by specifying the location of your JBoss application server. The port and type do not need to change.
5. Deploy the ISAPI filter on IIS.
6. Open the Internet Information Services Manager from Administrative Tools.
7. Expand the levels until the Default Web Site is visible. Right-click and select New, Virtual Directory.
8. Enter *jakarta* as the alias.
9. Reference the path where you installed the ISAPI plug-in.
10. Select Read, Run scripts (such as ASP), and Execute (such as ISAPI applications or CGI).
11. Click Next to continue and finish the wizard.
12. Right-click the Default Web Site and select properties, select the ISAPI Filters tab, and click Add.
13. Enter *jakarta* for the filter name and then click browse to select the isapi_redirect.dll. Then click OK twice.
14. For IIS 6.0, enable this filter under the Web Service Extensions.
15. Select the Web Service Extensions folder. Click the blue link on the left to Add a new Web service extension.

16. Supply Jakarta-Tomcat for the name. Click Add and browse for the same dll as above. Click OK Click the Set extension status to Allowed and then click OK.

17. Restart the IIS Server.

With the proxy now in place, you can access CA Identity Manager through IIS. For example, here are the links for accessing CA Identity Manager before and after the proxy configuration:

Before

http://identitymgr.forwardinc.ca:8080/idmmange
http://identitymgr.forwardinc.ca:8080/idmmange

After

http://smserver.forwardinc/idmmanage http://smserver.forwardinc/idmmanage

Note: A forward slash "/" can be needed at the end of this URL for the proxy to work. Reference the proxy logs if you are not forwarded to the Management Console.

Install the Proxy Plug-In on WebLogic

Once the Web Agent authenticates and authorizes a request for a CA Identity Manager resource, the Web Server forwards the request to the application server that hosts the CA Identity Manager Server.

1. Install the WebLogic proxy plug-in for your Web Server as described in the WebLogic documentation.

   Note: For IIS users, when you install the proxy plug-in, be sure to configure proxying by file extension and by path. When you configure proxying by file extension, add an application mapping in the App Mapping tab with the following properties:
   Executable: IISProxy.dll
   Extension: .wlforward

2. Configure the proxy plug-in for CA Identity Manager as described in one of the following sections:

   - IIS Proxy Plug-in (see page 298)
   - iPlanet Proxy Plug-in (see page 299)
   - Apache Proxy Plug-in (see page 302)

Configure the Proxy Plug-in for IIS (7.x)

The following procedure walks through the deployment and configuration of the WebLogic proxy plug-in for IIS 7.x.

Note: These instructions are for 32-bit operating environments. The same instructions apply for 64-bit operating environments. The location of the installation .dll file is different:

- %WL_HOME%server\plugin\win\32\
- %WL_HOME%server\plugin\win\64\

Follow these steps:

1. Install Web agent and configure it on IIS7.
2. Create a folder with the name ‘plugin’ in ‘C’ drive.
3. Copy the following files to the plugin folder:
   - Iisforward.dll
   - Iisproxy.dll
   - iisproxy.ini

   You can find these files at \\lodimmaple.ca.com\RegressionHarness\thirdparty\weblogic\Weblogic_Proxy_Files_IIS7.
4. Install the Application Development and Management Tools role services on IIS7.
5. Open Inet Manager and select the Default Web site.
6. Click Handler Mappings.
7. Double click Static File and modify the Request path to *.*.
8. Click the Request Restrictions button.
9. On the Mappings tab select Invoke handler only if the request is mapped to a File or folder.
10. In Handler Mappings dialog click Add Script Map… on the right-hand side menu options. Enter the following values:
    - Request path :   *
    - Executable: iisProxy.dll
    - Name: proxy
11. Click the Request Restrictions button.
12. Clear Invoke handler only if the request is mapped to.
13. Click Yes to the prompt about allowing this IASPI extension.

14. Click the Root node (Machine name) of the IIS Manager tree and click ISAPI and CGI Restrictions.
15. Click Add in Actions pane and enter the following values:
    - ISAPI or CGI Path: C:\plugin\ iisproxy.dll.
    - Description: Weblogic
    - Select Allow extension path to execute.
16. Click the Root node (Machine name) of the IIS Manager tree and click on the "ISAPI and CGI Restrictions". Select the option "Weblogic" and click on "Edit Feature Settings" at the right side pane.
17. Select Allow unspecified ISAPI modules and Allow unspecified CGI modules.
18. Do the same for Webagent.
19. In Features View, on the 'Default Web Site', double-click Handler Mappings.
20. On the Handler Mappings page, in the Actions pane, click Add Script Map and the following values:
    - Request path : .jsp
    - Executable : iisproxy.dll
    - Name : JSP
21. Click Request restrictions.
22. On the Mapping tab select Invoke handler only if request is mapped to File.
23. Click OK.
24. Click Add Script Map and the following values:
    - Request path : .do
    - Executable : C:\plugin\iisproxy.dll
25. Click Request restrictions. The settings are the same .jsp.
26. Click OK.
27. Click Add Script Map and enter the following values:
    - Request path : .wlforward
    - Executable : C:\plugin\iisproxy.dll
28. Click Request restrictions. The settings are the same as for .jsp.
29. Click Default Web Site and double click ISAPI Filters.
30. Click View Order List on the right-side pane.

31. Place the SiteMinder Agent executable in second place in the list. After this entry, only the Weblogic executable is in the list.

    Note: If SiteMinder Agent executable appears after the Weblogic executable, then move the SiteMinder Agent using MOVE UP action.

32. Click Application pools and change the Default App Pool to Classic mode.

The WebLogic plug-in is configured.

(WL)Configure the IIS 6.0 Proxy Plug-in

This procedure applies to configurations of the WebLogic proxy plug-in for IIS 6.0.x:

Follow these steps:

1. Create a folder on the system where the web agent is installed. For example: c:\weblogic_proxy.
2. Log in to the system where the CA Identity Manager server is running.
3. Go to this folder: *Weblogic_Home*\wlserver_11\server\plugin
4. Copy the following files to the weblogic proxy folder created in step 1.
   - iisforward.dll
   - iisproxy.dll
5. Create a file named iisproxy.ini in the same folder and include the following content:

```
# This file contains initialization name/value pairs
# for the IIS/WebLogic plug-in.
WebLogicHost=host-name
WebLogicPort=7001
ConnectTimeoutSecs=20
ConnectRetrySecs=2
WlForwardPath=/castylesr5.1.1,/iam,/im , /ca/Odata/
WLLogFile= c:\weblogic_proxy \proxy.log
DebugConfigInfo=ON
```

   Replace *host-name* with the actual host name.

6. Start IIS Manager.
7. Expand Web Sites.
8. Right-click Default Web Site.
9. Select Properties.

10. Add a filter as follows:
    a. Click ISAPI Filters.
    b. Click Add and complete the dialog as follows:

       For Filter Name: WebLogic

       For Executable: Path of the iisforward.dll
11. Provide the location of the iisproxy.dll file as follows:
    a. Click Home Directory.
    b. Click Configuration.
    c. Click Add.
    d. Enter the path of the iisproxy.dll file.
    e. Enter .jsp in the Extension field.
    f. Clear the Verify that file exists option.
12. Repeat step 11 for the .do and .wlforward extensions.
13. Add a web service extension for wlforward (in all lower case) pointing to the location of iisforward.dll.

    Set the extension status to Allowed.
14. Right click each Web Service Extension to change it to Allowed Status.
15. Restart the IIS web server.

Configure the iPlanet Proxy Plug-in

To configure the plug-in, modify the following iPlanet configuration files:

- magnus.conf
- obj.conf

The iPlanet configuration files have strict rules about the placement of text. To avoid problems, note the following points:

- Eliminate extraneous leading and trailing white space. Extra white space can cause your iPlanet server to fail.
- If you must enter more characters than you can fit on one line, place a backslash (\) at the end of that line and continue typing on the following line. The backslash directly appends the end of the first line to the beginning of the following line. If a space is necessary between the words that end the first line and begin the second line, be certain to use one space either at the end of the first line (before the backslash) or at the beginning of the second line.
- Do not split attributes across multiple lines.

The iPlanet configuration files for your iPlanet instance are found in the following location:

iplanet_home/https-*instance_name*/config/

where *iplanet_home* is the root directory of the iPlanet installation and *instance_name* is your particular server configuration.

Follow these steps:

1. From the *weblogic_home*/server/lib directory, copy the libproxy.so file that corresponds to your version of your iPlanet Web Server to the file system where you installed iPlanet.
2. In a text editor, modify the iPlanet magnus.conf file.

   To instruct iPlanet to load the libproxy.so file as an iPlanet module, add the following lines to the beginning of the magnus.conf file:

   ```
   Init fn="load-modules" funcs="wl_proxy,wl_init"\
   shlib=path in file system from step 1/libproxy.so
   Init fn="wl_init"
   ```

   For example:

   ```
   Init fn="load-modules" funcs="wl_proxy,wl_init"\
   shlib=/usr/local/netscape/plugins/libproxy.so
   Init fn="wl_init"
   ```

   The function load-modules tags the shared library for loading when iPlanet starts up. The values wl_proxy and wl_init identify the functions that the plug-in executes.

3. In a text editor, modify the iPlanet obj.conf file as follows:

   a. After the last line that begins with the following text:

      NameTrans fn=....

      Add the following Service directive to the Object name="default" section:

      ```
      Service method="(GET|HEAD|POST|PUT)" type=text/jsp fn="wl-proxy"
      ```

      Note: You may add this directive in a line following existing Service directives.

   b. Add the following code to the end of the file:

      ```
      <Object name="idm" ppath="*/iam/*">">
      Service fn="wl-proxy" WebLogicHost="hostname" WebLogicPort="portnumber"
      PathTrim="/weblogic"
      </Object>
      <Object name="weblogic1" ppath="*/console*">
      Service fn="wl-proxy" WebLogicHost="hostname" WebLogicPort="portnumber"
      PathTrim="/weblogic"
      </Object>
      ```

      where *hostname* is the server name and domain of the system where you installed WebLogic and *portnumber* is the WebLogic port (default is 7001).

      You may have more than one Object entry.

      For example:

      ```
      <Object name="idm" ppath="*/iam/*">">
      Service fn="wl-proxy" WebLogicHost="MyServer.MyCompany.com"
      WebLogicPort="7001" PathTrim="/weblogic"
      <Object name="weblogic1" ppath="*/console*">
      Service fn="wl-proxy" WebLogicHost="MyServer.MyCompany.com"
      WebLogicPort="7001" PathTrim="/weblogic"
      </Object>
      ```

4. Save your iPlanet configuration file.

5. Restart your Web Server instance.

IM_12.8--Configure the Apache Proxy Plug-in

Configuring the Apache Proxy Plug-in requires editing the http.conf file.

Follow these steps:

1. Stop the Apache web server after installing a Web Agent on Solaris and copy the mod_wl_20.so file from the following location:

   weblogic_home/server/lib/solaris

   to

   apache_home/modules

2. Edit the http.conf file (located in *apache_home*/conf) and make the following changes:

   a. Under the load module section, add the following code:
   ```
   LoadModule weblogic_module     modules/mod_wl_20.so
   ```

   b. Edit the server name with the name of the Apache server system.

   c. Add an If block at the end of the file as follows:
   ```
   <IfModule mod_weblogic.c>
     WebLogicHost weblogic_server.com
     WebLogicPort 7001
     MatchExpression /iam
     MatchExpression /castylesr5.1.1
     MatchExpression /ca/Odata
   </IfModule>
   ```

3. Save the http.conf file.

4. Restart the Apache web server.

Associate the SiteMinder Agent with an CA Identity Manager Domain

The policy administrator performs this task after completing the CA Identity Manager tasks. While you load your environments into CA Identity Manager, reference the 4.X agent. SiteMinder uses that agent when creating the Domain/Realm on the SiteMinder Policy Server. This agent validates SMSESSION cookies. Update the Domain/Realm and reference the fully functioning agent that is on the web server is used to access CA Identity Manager. This web server acts as the access point to CA Identity Manager and creates SMSESSION cookies.

Follow these steps:

1. Log in to the SiteMinder Administrative UI.

2. Navigate to Policies, Domains.

3. Modify the domain for your environment.

4. On the Realms tab, edit the first listed realm: XXX_ims_realm.

5. Search and select the agent on your proxy.

   Note: If you do not have a proxy agent (web server agent), create one. Verify that you have a web server and proxy in place to front CA Identity Manager.

6. Click OK twice and then repeat this process for the Public realm XXX_pub_realm.

7. After you update both realms, click Submit.

8. Wait for the agent to refresh, or restart the web server where the proxy agent is located.

Configure SiteMinder LogOffUrI Parameter

After you add SiteMinder to the environment, the logoff in CA Identity Manager does not really do anything. To reenable this functionality, update the Agent Configuration Object (ACO) for the agent on the proxy.

Follow these steps:

1. Log in to the SiteMinder Administrative UI. Click the Infrastructure tab, Agents, Expand Agent Configuration and then click Modify Agent Configuration.

2. Locate your ACO. Locate the #LogoffUri parameter. Click the play button (arrow pointing to the right) to the left of that parameter.

3. Remove the pound sign (#) from the name in the Value field and enter /idm/logout.jsp.

4. Click OK and then Submit to update the agent configuration object.

   The next time that the agent retrieves its configuration from the policy server, the new setting is propagated.

Troubleshooting

The following topics describe common errors that you can occur. Where at all possible a resolution has been paired with the error to assist you with your integration.

Missing Windows DLL

Symptom:

Missing Windows DLL (MSVCP71.dll)

We observed that after the SiteMinder connection was enabled, JBoss threw a java error complaining about a missing DLL (MSVCP71.dll).

Note: This error may not appear if JBoss is running as a service. If at all possible, test your configuration without running JBoss as a service.

Solution:

Follow these steps:

1. Locate MSVCP71.dll on the SiteMinder Policy Server, if it is running on Windows.
2. Copy this DLL (MSVCP71.dll) into the \Windows\system32 folder.
3. After you place this file in the correct location, register it with the OS.
4. From a command window, run the regsvr32 command. As long as the file is loaded you should be ok.
5. Restart the application server.

Incorrect SiteMinder Policy Server Location

Symptom:

Incorrect SiteMinder Policy Server Location.

Solution:

An incorrect location is referenced in ra.xml the error, "Cannot connect to policy server: xxx" appears.

Follow these steps:

1. Verify the hostname provided in ra.xml.

```
</config-property>
<config-property>
    <config-property-name>ConnectionURL</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value>smserver.forwardinc.ca,44441,44442,44443</config-property-value>
</config-property>
<config-property>
    <config-property-name>UserName</config-property-name>
```

2. In the ConnectionURL property, specify your SiteMinder Policy Server hostname. Use a FQN (Fully Qualified Name).

Incorrect Admin Name

Symptom:

Incorrect Admin Name

Solution:

An incorrect admin is referenced in ra.xml, the error "Unknown administrator" appears.

Follow these steps:

1. Check the UserName property in ra.xml.

```
    <config-property-value>smserver.forwardinc.ca,44441,44442,44443</co
</config-property>
<config-property>
    <config-property-name>UserName</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value>SiteMinder</config-property-value>
</config-property>
<!--The property 'password' has been removed.  'AdminSecret' is used in
This is due to the fact that we have added algorithm name padding in th
the algorithm name (for ex, PBES) with its own handlers.  This crashes
```

2. In the UserName property, specify the account used to communicate with CA SiteMinder. For example, use the SiteMinder account (default value).

Incorrect Admin Secret

Symptom:

Incorrect Admin Secret

Solution:

An incorrect admin secret is used in ra.xml, the error, "Cannot connect to the policy server: Invalid credentials" appears.

Follow these steps:

1. Check the AdminSecret property in ra.xml.

```
-->
<config-property>
    <config-property-name>AdminSecret</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value>{PBES}:xfx8/9xxmHD0B3Raw9VZJA==</config-property-value>
</config-property>
<config-property>
    <config-property-name>AgentName</config-property-name>
```

2. In the AdminSecret property, specify the encrypted password for the username referenced in the UserName property.

More information:

Modify a SiteMinder Password or Shared Secret (see page 334)

Incorrect Agent Name

Symptom:

Incorrect Agent Name

Solution:

An incorrect agent name is used in ra.xml, the error, "Cannot connect to the policy server: Failed to init Agent API: -1" appears.

Follow these steps:

1. Check the AgentName property in ra.xml.

```
</config-property>
<config-property>
    <config-property-name>AgentName</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value>idmagent</config-property-value>
</config-property>
<config-property>
```

2. Specify the 4.X agent name that you created during the 3rd step of the SiteMinder configurations.

Incorrect Agent Secret

Symptom:

Incorrect Agent Secret

Solution:

An incorrect agent secret is used in ra.xml, the error "Cannot connect to the policy server: Failed to init Agent API: -1" appears with a preceding crypto handler error.

Follow these steps:

1. Check the AgentSecret property in ra.xml.

```
<config-property>
    <config-property-name>AgentSecret</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value>{PBES}:xfx8/9xxmHD0B3Raw9VZJA==</config-property-value>
</config-property>
<config-property>
```

2. Specify the encrypted password that was used when creating that agent.

More information:

Modify a SiteMinder Password or Shared Secret (see page 334)

No User Context in CA Identity Manager

Symptom:

No User Context in CA Identity Manager.

If a user tries to access CA Identity Manager without a SMSESSION cookie, CA Identity Manager cannot authenticate the user. In this case, you can expect to see can emptyCA Identity Manager UI.

If you have Workflow enabled for your environment, expect to see a failure much like this.

```
Exception during page display:

java.lang.IllegalArgumentException
        at com.netegrity.webapp.bean.WorkList.(WorkList.java:84)
        at com.netegrity.webapp.bean.WorkList.(WorkList.java:70)
        at com.netegrity.webapp.bean.WorkList.getConsoleWorkListFromRequest(WorkList.java:109)
        at com.netegrity.taglib.skin.TagUtilLocal.getWorkItems(TagUtilLocal.java:660)
        at com.netegrity.taglib.skin.TagUtilLocal.hasWorkItems(TagUtilLocal.java:846)
        at com.netegrity.taglib.skin.IfWorkItemsTag.doStartTag(IfWorkItemsTag.java:73)
        at idm_jsp.app.ca12.home_jsp._jspService(Unknown Source)
        at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
        at org.apache.catalina.core.ApplicationDispatcher.invoke(ApplicationDispatcher.java:654)
        at org.apache.catalina.core.ApplicationDispatcher.doInclude(ApplicationDispatcher.java:557)
        at org.apache.catalina.core.ApplicationDispatcher.include(ApplicationDispatcher.java:481)
        at org.apache.jasper.runtime.JspRuntimeLibrary.include(JspRuntimeLibrary.java:968)
        at idm_jsp.app.ca12.index_jsp._jspx_meth_skin_ifhomepage_0(Unknown Source)
        at idm_jsp.app.ca12.index_jsp._jspService(Unknown Source)
        at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
        at org.apache.catalina.core.ApplicationDispatcher.invoke(ApplicationDispatcher.java:654)
        at org.apache.catalina.core.ApplicationDispatcher.processRequest(ApplicationDispatcher.java:445)
        at org.apache.catalina.core.ApplicationDispatcher.doForward(ApplicationDispatcher.java:379)
        at org.apache.catalina.core.ApplicationDispatcher.forward(ApplicationDispatcher.java:292)
        at com.netegrity.webapp.filter.ConsolePageFilter.doFilter(ConsolePageFilter.java:521)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
        at com.netegrity.webapp.page.jsf.FacesFilter.doFilter2(FacesFilter.java:180)
```

Solution:

A few things can cause this, but it is usually one of the following:

- You have directly accessed CA Identity Manager.
- The SiteMinder agent at the proxy is disabled (that is, nothing is protected – The SMSESSION Cookie is not being created).
- The SiteMinder Domain for the CA Identity Manager environment is misconfigured .

The first two causes are pretty straight forward. Make sure that you route through the web server with the fully functional web agent enabled. If however you are going through the web server and the agent is enabled, then you need to modify the Domain.

Follow these steps:

1. Log in to the SiteMinder Administrative UI.
2. Locate your CA Identity Manager Domain and click through the layers to modify it. Click the Realm Tab and then the first realm in the list.
3. The default location of the forward slash is under the realm. Delete it.
4. Click into the Rule under this Realm.

   The default effective resource for the rule is an asterisk "*".
5. Add the forward slash "/" in front of the asterisk.

   You have moved the forward slash from the realm to the rule. The protection is the same, but SiteMinder treats it differently.

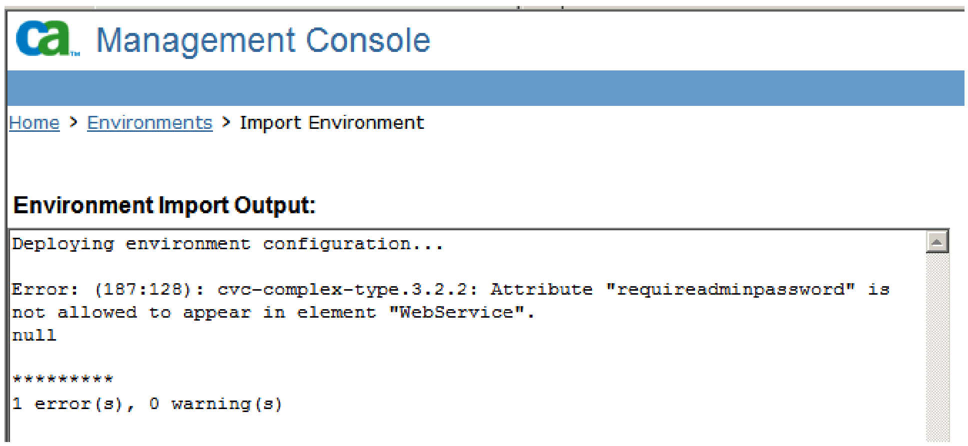
   You can successfully log in to CA Identity Manager through SiteMinder. To validate proper protection, review your SiteMinder agent logs.

Error Loading Environments

Symptom:

When importing an environment back into CA Identity Manager after integrating with SiteMinder, an error appears about attribute "requireadminpassword" and the element "WebService".

Note: This issue can also occur when SiteMinder is not part of the deployment.



Solution:

This error allows partial deployment of the environment. The partial deployment can create empty elements in the CA Identity Manager object store. Correct one of the environment XMLs and reimport.

Follow these steps:

1. Locate the archived ZIP file, and explore it.
2. Create a copy of the XXX_environment_settings.xml.
3. Edit this file and locate the "WebService" element.
4. Delete the tag "requireadminpassword="false."

   Note: Remove the tag *and* the value. Do not remove only the value.
5. Save your changes and place the file back into the ZIP file.
6. Reimport the archived environment zip file.

   You do not have to delete the environment that was created from the failed attempt. Reimporting a corrected file fixes the errors from the failed attempt.

Cannot Create a CA Identity Manager Directory or Environment

Symptom:

Cannot create an CA Identity Manager directory or environment, when SiteMinder integration is enabled.

Solution:

This issue can be caused by a missing entry in the registry.

Verify that the following registry setting exists on the SiteMinder Policy Server machine:

- Solaris or Linux:

  Verify that the following entry exists in sm.registry:
  ImsInstalled=8.0; REG_SZ

- Windows:

  Verify that setting "ImsInstalled=8.0; REG_SZ" exists in the following location:
  HKLM\SOFTWARE\Netegrity\SiteMinder\CurrentVersion

  Note: If the registry path \Netegrity\SiteMinder\CurrentVersion does not exist, create it manually.

If you change the registry, be sure to restart the Policy Server for the changes to take effect.

Important! Before you modify the registry, perform a full system backup.

User Cannot Log In

Symptom:

A new user cannot log into an environment with a clear text password.

Solution:

Verify that the following data classification is not included in the password attribute definition in the directory configuration file (directory.xml):

<DataClassification name="AttributeLevelEncrypt"/>

In environments that include the following components, enabling attribute level encryption prevents users from logging in:

- CA SiteMinder, and
- A relational database

How to Configure CA Identity Manager Agent Settings

When CA Identity Manager integrates with SiteMinder, CA Identity Manager uses a built-in CA Identity Manager agent to communicate with the SiteMinder Policy Server. To tune the performance, configure the following connection settings for the CA Identity Manager agent.

1. Complete one of the following steps:
   - If CA Identity Manager is running on a WebLogic or WebSphere application server, edit the resource adapter in the policyserver_rar connector descriptor in the console of application server.
   - If CA Identity Manager is running on a JBoss application server, open policyserver-service.xml from <JBoss_home>\server\default\deploy\iam_im.ear\policyserver_rar\META-INF.
2. Configure the settings as follows:

   ConnectionMax

   Sets the maximum number of connections to the policy server, for example, 20.

   ConnectionMin

   Sets the minimum number of connections to the policy server, for example, 2.

   ConnectionStep

   Sets the number of additional connections to open when all the agent connections are in use.

   ConnectionTimeout

   Specifies the amount of time in seconds that the agent is required to wait to connect to SiteMinder before timing out.
3. Restart the application server.

Configure SiteMinder High Availability

If you have created a SiteMinder Policy Server cluster, you can configure an application server cluster to use it for load balancing and failover.

Follow these steps:

1. Edit the ra.xml file in this location:
   ```
   WebSphere:
   WAS_PROFILE/config/cells/CELL_NAME/applications/iam_im.ear/deployments/IdentityMinder/policyserver_rar/META-INF
   Jboss:jboss_home/server/all/deploy/iam_im.ear/policyserver_rar/META-INF
   WebLogic: wl_domain/applications/iam_im.ear/policyserver_rar/META-INF
   ```
2. Modify these items, which are explained in the sections that follow:
   - Connection settings for the Policy Server
   - The number of Policy Servers
   - The selection of load balancing or failover for the cluster.
3. Repeat these steps for each CA Identity Manager server in the cluster.
4. Restart the application server for changes to take effect.

Note: When you are creating a CA Identity Manager directory or an environment or modifying directory or environment settings, set SiteMinder Failover and FailoverServers to false. Otherwise, the directory object could be created but not replicated in time to be used. For example, you create a directory in Server 1. Then, you create an attribute using the object ID of that directory on Server 2, but the second directory does not exist yet. You receive an Object not Found error.

Modify Policy Server Connection Settings

The Policy Server connection information must reflect the primary server for the production environment. This information consists of the ConnectionURL, the user name and password for the SiteMinder Admin account, and the name and shared secret for the Agent.

In the following example, the editable values appear in CAPITAL LETTERS.

```
<config-property>
        <config-property-name>ConnectionURL</config-property-name>
   <config-property-type>java.lang.String</config-property-type>
        <config-property-value>DEVELOPMENT.SEVERCOMPANY.COM,VALUE,VALUE,VALUE</config-
                property-value>
</config-property>
```

```
<config-property>
        <config-property-name>UserName</config-property-name>
   <config-property-type>java.lang.String</config-property-type>
        <config-property-value>SITEMINDER-ADMIN-NAME</config-property-
                value>
</config-property>

<config-property>
        <config-property-name>AdminSecret</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>ENCRYPTED-PASSWORD</config-
                property-value>
</config-property>
<config-property>
        <config-property-name>AgentName</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>DEVELOPMENT-AGENT-NAME</config-property-
                value>
</config-property>

<config-property>
        <config-property-name>AgentSecret</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>ENCRYPTED-AGENT-SECRET</config-
                property-value>
</config-property>
```

Note: For the values that require encrypted text, use the CA Identity Manager password tool. For more information, see the *Configuration Guide*.

Add More Policy Servers

To add more Policy Servers to the CA Identity Manager installation instance, edit the FailoverServers entry in the ra.xml file.

Note: Include the primary Policy Server and all failover servers in the FailoverServers entry.

For each Policy Server, enter an IP address and port numbers for authentication, authorization, and accounting services. Use a semi-colon to separate entries as shown here:

```
<config-property>
        <config-property-name>FailoverServers</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>
                172.123.123.123,44441,44442,44443;172.123.123.124,33331,
                33332,33333
        </config-property-value>
</config-property>
```

Select Load Balancing or Fail Over

The default behavior of CA Identity Manager is to use round-robin load balancing using the servers that are identified by the ConnectionURL and FailoverServers. Load balancing occurs if you leave FailOver set to false.

To select the failover, set FailOver to true:

```
<config-property>
        <config-property-name>FailOver</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>true</config-property-value>
</config-property>
```

Removing SiteMinder from an Existing CA Identity Manager Deployment

This section provides detailed instructions for removing CA SiteMinder from an existing CA Identity Manager environment.

Follow these steps:

Important! Password history information will not be accessible after the migration.

1. Stop the application server.
2. Disable the Policy Server in the ra.xml file located in \iam_im.ear\policyserver.rar\META-INF by setting the Enabled config-property value to false.
3. Edit the web.xml file located in \iam_im.ear\User_console.war/WEB-INF and set the FrameworkAuthFilter property to Enabled = true.

   Note: For WebSphere, the web.xml file is located in *WebSphere_home*/AppServer/profiles/*Profile_name*/config/cells/*Cell_name*/applications/iam_im.ear/deployments/IdentityMinder/user_console.war/WEB-INF.

4. Start the application server.
5. (WebSphere only) Update the policyServer object in the Administrative Console with same values as in the ra.xml file.

SiteMinder Operations

The following sections discuss how to modify SiteMinder features, including policy domains and authentication schemes, to support CA Identity Manager:

Collect User Credentials Using a Custom Authentication Scheme **(see page 316)**

Changes the method that CA Identity Manager uses to collect credentials for users who try to access a CA Identity Manager Environment.

Configure Access Roles **(see page 317)**

Provides access to functions in an application.

Configure the LogOff URL **(see page 331)**

Prevents unauthorized access to a CA Identity Manager Environment by enforcing a full logout.

Update an Alias in SiteMinder Realms **(see page 332)**

Updates the realms that protect a CA Identity Manager Environment when you change the alias of the Environment.

SiteMinder Passwords **(see page 334)**

Lets you change the password for the administrator account that CA Identity Manager uses to communicate with SiteMinder, and the shared secret for the SiteMinder agent that protects a CA Identity Manager Environment.

Configure CA Identity Manager Agent Settings **(see page 311)**

Tunes the performance of the CA Identity Manager agent that communicates with the SiteMinder Policy Server.

Use Different Directories for Authentication and Authorization **(see page 335)**

Enables administrators who have profiles in one directory to manage users in a different directory.

Improve the Performance of LDAP Directory Operations **(see page 337)**

Increases the throughput of CA Identity Manager requests to the user store by configuring SiteMinder to open multiple connections to the same directory.

Collect User Credentials Using a Custom Authentication Scheme

SiteMinder uses an authentication scheme to collect user credentials and determine identity of a user at login time. Once a user is identified, CA Identity Manager generates a personalized User Console that is based on the privileges of the user.

You can implement any SiteMinder authentication scheme to protect a CA Identity Manager Environment.

For example, you can implement an HTML Forms Authentication Scheme, which collects credentials in an HTML form. Using an HTML form lets you create a login page that may include branding elements, such as a company logo, and links to the self-registration and forgotten password pages.

Note: For information about authentication schemes, see the *CA SiteMinder Policy Server Configuration Guide*.

Follow these steps:

1. Log in to one of the following interfaces:
   - For CA SiteMinder Web Access Manager r12 or higher, log in to the Administrative UI.
   - For CA eTrust SiteMinder 6.0 SP5, log in to the Policy Server User Interface.

   Note: For information about using these interfaces, see the documentation for the version of SiteMinder that you are using.
2. Create an authentication scheme as described in the *CA SiteMinder Policy Server Configuration Guide*.
3. Modify the realm that protects the appropriate CA Identity Manager Environment to use the authentication scheme you created in Step 1.

   The realm name has the following format:

   *Identity Manager-environment*_ims_realm

   Note: If you configured support for public tasks, you see an additional realm, *Identity Manager-environment*_pub_realm. This realm uses an anonymous authentication scheme to enable unknown users to use the self-registration and forgotten password features without supplying credentials. Do not modify the authentication schemes for these realms.

IM_12.6.2--Import Data Definitions into the Policy Store

You can control the access of a user to application functions using SiteMinder policies. The Policy Server installation includes the required data definitions to allow this control. You import the IdmSmObjects.xdd file from this location:

siteminder_home\xps\dd

siteminder_home is the Policy Server installation path.

Use the following command from the Policy Server:

```
C:\Program Files (x86)\CA\siteminder\xps\dd>XPSDDInstall
IdmSmObjects.xdd
```

How to Configure Access Roles

Access roles enable centralized management of user privileges in external applications that SiteMinder has secured. CA Identity Manager administrators can create and assign roles in the CA Identity Manager User Console that determine access to users to applications outside of CA Identity Manager. For example, a Role Administrator may create roles in the User Console that control access to a finance application and grant the ability to assign the roles to the Help Desk administrator. The Help Desk administrator can assign or revoke that role through the User Console.

Access roles are enabled through integration with SiteMinder. SiteMinder associates roles with policies to determine which users can access a protected resource and to deliver user-specific roles and task information to protected resources.

Access roles require configuration in CA Identity Manager and SiteMinder. Two administrators are involved:

- The CA Identity Manager administrator creates access roles and tasks in CA Identity Manager. The default System Manager and Access Role Manager roles include these tasks.
- The SiteMinder administrator manages System and Domain objects in CA SiteMinder. The SiteMinder administrator must have System scope.

  Note: For more information, see the *Policy Server Configuration Guide*.

The following procedure outlines the steps to create an access role. Review these steps *before* configuring access roles for use with SiteMinder.

1. A CA Identity Manager administrator completes the following tasks:
   a. Enables access roles and tasks for use with SiteMinder.
   b. Creates access tasks.
   c. Creates an access role.
   d. Communicates role and task information to the SiteMinder administrator for the purpose of creating SiteMinder role-based access control policies.
2. A SiteMinder administrator creates a role-based access control policy by completing the following steps:
   a. Assigning a user directory that is associated with one or more CA Identity Manager environments to a Policy Domain.
   b. Associating one or more CA Identity Manager environments with the Policy Domain in step 1.
   c. Creating realms and rules in the Policy Domain (if they do not exist). The realms and rules must correspond to the resources to which the access roles grants access.
   d. Creating policies and binding them to roles from the CA Identity Manager environment.
   e. (optional) Specifying responses which deliver entitlement information to the protected resources.

   Note: For detailed instructions on these steps, see the *Policy Server Configuration Guide*.

More information:

Enable Access Roles for Use with SiteMinder (see page 318)

Enable Access Roles for Use with SiteMinder

To use access roles with CA SiteMinder, CA Identity Manager mirrors all objects in the CA Identity Manager object store that are related to those access roles in the SiteMinder policy store. To enable access roles for use with SiteMinder, configure a property in the CA Identity Manager Management Console.

Follow these steps:

1. Open the Management Console.
2. Select Environment, *Your Environment*, Advanced Settings, Miscellaneous.

3. Add a property by providing the following information:
   - In the Property field, enter the following text:

     EnableSMRBAC
   - In the Value field, enter the following text:

     true
4. Click Add. Then, click Save.

   A message appears indicating that the environment to restart.
5. Click Restart Environment.

   CA Identity Manager now supports access roles and tasks for use with CA SiteMinder.

Once you enable access roles for use with CA SiteMinder, note the following points:

- If you used access roles in CA Identity Manager r8x, perform an additional migration step to manage those access roles in the current version of CA Identity Manager. For more information, see the *Upgrade Guide.*
- To disable support for access roles in SiteMinder, delete the CA Identity Manager access role and task objects from the SiteMinder policy store. Then, remove the “EnableSMRBAC” Property from the Miscellaneous Properties list and restart the Environment.

Add Access Task to Admin Role

By default, the Access tasks tasks does not appear under Roles and Tasks tab, you need to add the Access tasks to Admin role of the logged in user.

Follow these steps:

1. Log in to a CA Identity Manager account with a role that includes a task for creating access roles.
2. Click Roles and Tasks, Modify Admin Role.
3. Select the Admin role of logged in user.
4. Click Tasks tab, Filter by Category field, Select Roles and Tasks from the drop down.
5. Select Create Access Task from Add Task dropdown.
6. Click Submit.

Create an Access Task

An access task is a single action that a user can perform in a business application, such as generating a purchase order in a finance application. Users can perform that action when they are assigned an access role that includes the access task.

Important! To create an Access tasks, you need to add the Access tasks (see page 319) to Admin role of the logged in user.

Follow these steps:

1. Select Roles and Tasks, Access Tasks, Create Access Task.
2. Select one of the following options:
   - Create an access task
   - Create a copy of an access task.
3. Complete these fields:

   Name

   A unique name that you can assign to the task, such as Generate Purchase Order.

   Tag

   A unique tag for the task. The tag must start with a letter or an underscore character containing letters, numbers, or underscores only.

   Description

   An optional note about the purpose of the task.

   Application ID

   An identifier for an application such as the application name associated with the task. The application ID cannot contain any spaces or nonalphanumeric characters.

   Make note of this ID; you need it when you enable the role in SiteMinder.
4. To complete the access task, click Submit.

How to Create an Access Role

An access role contains access tasks, which provide access to functions in an application. For example, a role may contain tasks that enable role members to place an order in a purchasing application and update quantities in an inventory control application.

You complete the following steps to create an access role:

1. Begin access role creation. (see page 321)
2. Define basic properties for the access role in the Profile tab. (see page 321)
3. Select access tasks for the role. (see page 322)
4. Define member policies for the role. (see page 323)
5. Define admin policies for the role. (see page 323)
6. Define owner rules for the role. (see page 324)

Begin Access Role Creation

Follow these steps:

1. Log in to a CA Identity Manager account with a role that includes a task for creating access roles.
2. Click Access Roles, Create Access Role.

   Choose the option to create a role or a copy of role. If you select Copy, search for the role.
3. Continue with next section, Define the Profile of an Access Role.

Define the Profile of an Access Role

Follow these steps:

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

   Note: You can specify custom attributes on the Profile tab that specify additional information about access roles. You can use this additional information to facilitate role searches in environments that include a significant number of roles.
2. Select Enabled if you are ready to make the role available for use as soon as you create it.
3. Continue with the next section, Define Member Policies for an Access Role.

Select Access Tasks for the Role

On the Tasks tab:

1. Select the tasks to include in this role. First, select the applications, then the task. You can include tasks from different applications:

   Note: If another role has the tasks you need, click Copy Tasks from another role. You can edit the list that appears.

   In creating a role or task, you see icons for adding, editing, and removing items:

   Go forward or select the current item to view or edit it.

   If JavaScript is disabled, press the forward button to select from a drop-down list.

   Go back or undo a previous selection.

   Insert an element, such as a task or rule.

   Delete the current task or, in a rule, the expression that follows.

   Move up the current item in the list.

   Move the current item down in the list.

2. Continue with the next section, Define Admin Policies for an Access Role.

Define Member Policies for an Access Role

A member policy defines a member rule and scope rules for a role. You can define several member policies for one role. For each policy, users who meet the condition in the member rule has the scope for using the role that is defined in the policy.

Follow these steps:

1. Select the Members tab.
2. Select Add to define the member policies.
3. (Optional) On the Member Policy page, optionally define a member rule for who must be able to use this role.

   Defining a member rule automatically assigns the role to users who match the criteria in the member policy.

   Note: Define member policies that use only directory attributes, for example: title=Manager. If you define member policies referencing to those objects not stored in the user directory such as admin roles, SiteMinder cannot be able to resolve the reference.

4. Verify that the Member Policy appears on the Members tab.

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

5. On the Members tab, enable the Administrators can add and remove members of this role check box.

   Once you enable this feature, you define the Add Action and Remove Action. These actions define what happens when a user is added or removed as a member of the role.

Define Admin Policies for an Access Role

An admin policy defines admin rules, scope rules, and administrator privileges for a role. You can define several admin policies for a role. Each policy indicates that if an administrator meets the condition in the admin rule has the scope and administrator privileges that is defined for the policy.

Follow these steps:

1. Select the Administrators tab for the access role.
2. If you want to make the Manage Administrators option available, enable the Administrators can add and remove administrators of this role check box.

   Once you enable this feature, define the actions for when a user is added or removed as an administrator of the role.

3. On the Administrators tab, add admin policies that include admin and scope rules and administrator privileges. Each policy needs at least one privilege (Manage Members or Manage Administrators).

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

   Note: Define admin policies that use only directory attributes, for example: title=Manager. If you define member policies referencing to those objects not stored in the user directory such as admin roles, SiteMinder cannot be able to resolve the reference.

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

5. Continue with the next section, Define Owner Rules for an Access Role.

Define Owner Rules for an Access Role

An owner rule defines who can modify a role. You can define several owner rules for a role.

Follow these steps:

1. Select the Owners tab for the access role.

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

   Note: Define owner rules that use only directory attributes, for example: title=Manager.If you define owner rules referencing to those objects not stored in the user directory such as admin roles, SiteMinder cannot be able to resolve the reference.

3. Click Submit.

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

Access Roles in SiteMinder

To configure roles-based access control to protected resources, a SiteMinder administrator associates a CA Identity Manager Environment with a Policy Domain in the Policy Server User Interface. The administrator creates a policy to protect an application and associates a role or roles with that policy. Users who have an associated role can access the protected application.

A SiteMinder administrator binds roles to security policies that define how users interact with resources. Policies link with the following objects:

- Users and User Groups

  Identify a set of policy affected users.

- Roles

  Identify users who have been assigned a set of privileges in CA Identity Manager.

- Rules

  Identify a resource and the actions that are allowed or denied for the resource. The resource is typically a URL, application, or script.

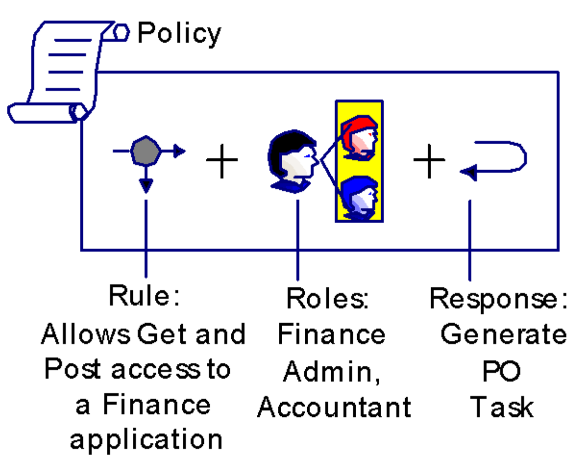
- Responses

  Determine a reaction to a rule. When a rule fires, responses are returned to a SiteMinder Agent.

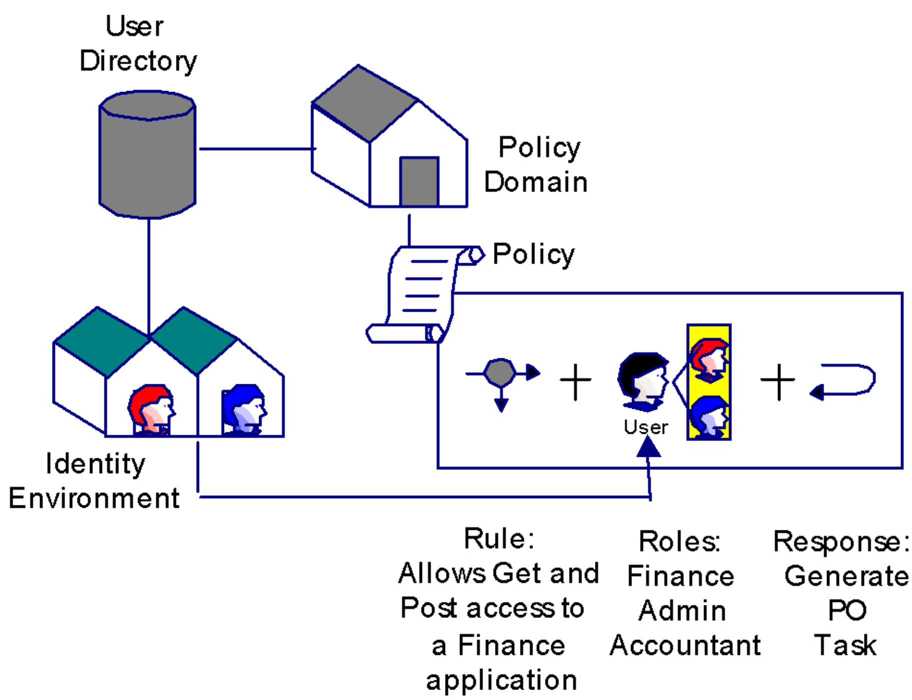
  CA Identity Manager uses SiteMinder responses to deliver specific task and role information to a protected resource.

You can bind SiteMinder policies to users, or to roles, or to users *and* roles. Assume that a user or role member attempts to access a protected resource. SiteMinder uses information in the policy to determine whether to grant access, and to trigger responses.

The following figure illustrates the relationship of policy objects in a role-based policy.



SiteMinder policies are created in policy domains, which logically tie user directories to protected resources. The following figure illustrates the relationship of policy objects in a role-based policy.



To supply user entitlements to a protected application, the SiteMinder administrator pairs a rule with the policy of an application with a response. The response contains a SiteMinder-generated response attribute that retrieves entitlement information from CA Identity Manager.

When SiteMinder authorizes a role member for a protected resource, the following events take place:

1. The rule of a policy executes in SiteMinder, triggering the paired response.
2. The Policy Server obtains entitlement information from CA Identity Manager to include in a response.
3. The Policy Server passes the response attribute to the Web Agent.
4. The Web Agent makes the entitlement information available to the application as an HTTP header variable or a cookie.

SiteMinder-Generated Response Attributes

CA Identity Manager passes entitlement information to applications through SiteMinder Web Agent responses. These responses contain HTTP header variables in response attributes, which the application can use to determine access privileges of a user. Responses are included in SiteMinder policies, which determine how users interact with a protected resource.

SiteMinder administrators can configure a response that includes two types of response attributes to pass information to an application:

- SM_USER_APPLICATION_ROLES[:*application id*]--Returns a list of roles that are assigned to a user.
- SM_USER_APPLICATION_TASKS[:*application id*]--Returns a list of tasks a user can perform based on roles that are assigned.

The application ID limits the requested set of roles and tasks to a specific application. For example, if you create the following response attribute:

SM_USER_APPLICATION_ROLES:Finance_application

SiteMinder returns the roles that have tasks in the Finance application to the Web Agent, which then passes the information to the Finance application.

Note: The *application id* you supply must match an *application id* you supplied when you used Create Access Task in CA Identity Manager. If the task is not yet created, you can choose any name for the application ID but it cannot contain any spaces or nonalphanumeric characters.

You can specify multiple application IDs in a comma-delimited list to return the set of roles and tasks from multiple applications in a single response attribute. For example, to return the list of roles that a user has in the Finance and Purchasing applications specify in the following way:

SM_USER_APPLICATION_ROLES:Finance, Purchasing

How to Enable Access Roles in SiteMinder

The following steps assume that SiteMinder already protects the application to which the access role grants access. For instance, assume that you are creating an access role for an application that SiteMinder does not protect yet. In such case, see one of the following guides in the SiteMinder bookshelf:

- For SiteMinder 6.0 SP5, see the *Policy Design Guide.*
- For SiteMinder 12.0 SP2, see the *Policy Server Configuration Guide.*

Note: To configure access roles in SiteMinder, use the Policy Server User Interface, an applet–based application, instead of the SiteMinder Administrative UI. In SiteMinder 12, this applet is named the SiteMinder Federation Security Services Administrative UI (FSS Administrative UI). You can install the FSS Administrative UI using the Policy Server installer.

To enable access roles in SiteMinder, you complete the following high-level steps:

1. In the Policy Server User Interface, associate a user directory and a CA Identity Manager environment with a Policy Domain (see page 328).
2. In the Policy Domain, create realms and rules (if they do not exist) corresponding to the resources to which the access role grants access.

   Note: For information about creating realms and rules, see one of the following guides in the SiteMinder bookshelf:

   - For SiteMinder 6.0 SP5, see the *Policy Design Guide.*
   - For SiteMinder 12.0 SP2, see the *Policy Server Configuration Guide.*
3. Create a response (see page 329) to pass entitlement information to the resource.
4. Create a policy and associate it with the following objects:
   - The access role (see page 330)
   - The realms and rules you created in step 2.
   - The responses created in step 3.

   Note: For information on creating policies, see the *Policy Design Guide* (for SiteMinder 6.0 SP5) or the *Policy Server Configuration Guide* (for SiteMinder 12.0 SP2).

Add CA Identity Manager Environments to a Policy Domain

To enable SiteMinder to support access roles, you associate a CA Identity Manager environment with a user directory and a policy domain in SiteMinder.

Note: Add the user store that is associated with the CA Identity Manager environment to the policy domain *before* you can add the CA Identity Manager environment to the policy domain.

Follow these steps:

1. In the Policy Domain dialog in the Policy Server User Interface, add the user store that is associated with the CA Identity Manager environment with a policy domain as follows:
   a. Select the User Directories tab.
   b. From the drop-down list box at the bottom of the tab, select the user directory to include in the policy domain.
   c. Click the Add button.

      The Policy Server User Interface adds the directory to the list displayed in the User Directories tab.
   d. Click Apply.
2. Add the CA Identity Manager environment to the policy domain as follows:
   a. Select the CA Identity Manager Environments tab.
   b. Select the CA Identity Manager Environment that you want to associate with the policy domain from the drop-down list, at the bottom of the tab.
   c. Click Add.

      The Policy Server User Interface adds your selection to the list of CA Identity Manager environments at the top of the tab.
3. Click OK to save your selections and close the dialog.

   The CA Identity Manager environments that you selected are available when you create policies.

Create a SiteMinder Response

Follow these steps:

1. Log in to the Policy Server User Interface.
2. Depending on your administrative privileges, do one of the following tasks:
   - If you have the Manage System and Domain Objects privilege:
     a. In the Object pane, click the Domains tab.
     b. Select the policy domain to which you want to add a response.
   - If you have the Manage Domain Objects privilege, select the policy domain to add a response in the Object pane.
3. From the menu bar, select Edit, <domain name>, Create Response.

   The SiteMinder Response dialog opens (see Response Dialog).
4. Enter a name and description for the new response.
5. In the Agent Type group box, select the SiteMinder radio button.

6. Select the Web Agent option from the drop-down list in the Agent Type group box and click Apply to save your changes.
7. Click Create.

   The SiteMinder Response Attribute Editor dialog opens.
8. From the Attribute drop-down list, select the WebAgent-HTTP-Header-Variable response attribute.
9. In the Attribute Setup tab, select the User Attribute radio button.
10. In the Variable field, enter the name of the variable that passes to the application.

    For example, if you specify the variable TASKS, the following header is returned to the application:

    HTTP_TASKS
11. In the Attribute Name field, specify the response attribute as follows:
    - SM_USER_APPLICATION_ROLES[:*application id1, application_id2, ...application_idn*]--Returns a list of roles that are assigned to a user.
    - SM_USER_APPLICATION_TASKS[:*application id1, application_id2, ...application_idn*]

    SiteMinder-Generated Response Attributes (see page 327) provides more information.
12. Click OK to save your changes and return to the SiteMinder Administration window.

Add Roles to a SiteMinder Policy

When a user is assigned to an appropriate access role accessing a protected resource, the SiteMinder Policy Server verifies access roles assignment to the user. Upon verification, it fires the rules included in the policy to check whether the user is allowed to access the resource or not.

Follow these steps:

1. In the SiteMinder Policy dialog, click the Users tab.

   The Users tab contains tabs for each user directory and CA Identity Manager environment included in the policy domain.
2. Select the CA Identity Manager Environment that contains the roles you want to add to the policy.
3. Click the Add/Remove button.

   The SiteMinder Policy CA Identity Manager Role dialog opens.
4. To add roles to the policy, select an entry from the Available Members list and move it to the Current Members list.
5. Click OK to save your changes and return to the SiteMinder Policy dialog.

Exclude Roles in a Policy

In addition to using access roles to grant access to applications, you can also use access roles to prevent members of access roles from accessing an application. To prevent access role members from accessing an application, you exclude the roles from SiteMinder policies. When a user who has been assigned the excluded access role in CA Identity Manager tries to access a protected resource, the Policy Server verifies exclusion of the CA Identity Manager role to the assigned user. Upon verification, it blocks access to the resource.

Follow these steps:

1. In the SiteMinder Policy dialog, click the Users tab.

   The Users tab contains tabs for each user directory and CA Identity Manager Environment included in the policy domain.

2. Click the CA Identity Manager Environment that contains the roles you want to exclude from your policy.

3. Click the Add/Remove button.

   The SiteMinder Policy CA Identity Manager Role dialog opens.

4. To add roles to the policy, select an entry from the Available Members list and click on the Left Arrow button, which points to the Current Members list.

   The opposite procedure removes roles from the Current Members list.

5. In the Current Members list, select the roles to exclude, and click the Exclude button that is located under the list.

   A red circle with a slash appears to the left of the excluded roles.

6. Click OK to save your changes and return to the SiteMinder Policy dialog.

Configure the LogOff URI

To protect a CA Identity Manager environment, configure the SiteMinder Web Agent that protects the environment to terminate a user session after the user logs off CA Identity Manager.

The Web Agent terminates a user session by deleting the SiteMinder session and authentication cookies from the Web browser and instructing the Policy Server to remove any session information.

To terminate the SiteMinder session, configure logout functionality in the LogOffURI field in the Agent Configuration Object for the SiteMinder agent that protects the CA Identity Manager environment.

Notes:

- A SiteMinder agent has one LogOff URI. All applications protected by the agent use the same logout page.
- When you configure custom logout pages in the Management Console as described in Configure Custom Logout Pages, CA Identity Manager sends the logout request to the custom logout page *and* the LogOff URI. However, CA Identity Manager displays only the custom logout page to the user.

Follow these steps:

1. Log in to one of the following interfaces:
   - For CA SiteMinder r12 or higher, log in to the Administrative UI.
   - For CA eTrust SiteMinder 6.0 SP5, log in to the Policy Server User Interface.

   Note: For information on using these interfaces, see the documentation for the version of SiteMinder that you are using.
2. Modify the #LogOffUri property in the Agent Configuration Object for the agent that protects the CA Identity Manager environment as follows:
   - Remove the pound sign (#)
   - In the Value field, specify the following URI:

     /iam/im/logout.jsp

   Note: You select an Agent Configuration Object when you install the Web Agent. For more information, see the *CA SiteMinder Web Access Manager Policy Server Installation Guide*.
3. Save the changes.
4. Restart the Web server.

Aliases in SiteMinder Realms

An *alias* is a unique string that is added to the URL for accessing a CA Identity Manager environment. For example, when the alias of an environment is *employees*, the URL for accessing that environment is as follows:

`http://`*`myserver.mycompany.org`*`/iam/im/employees`

myserver.mycompany.org

Defines the fully qualified domain name of the server where CA Identity Manager is installed.

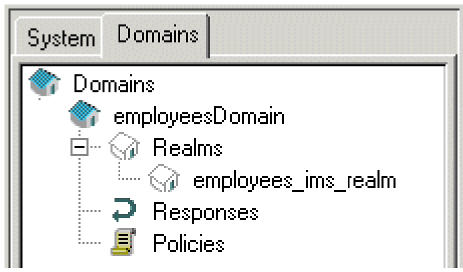
You specify at least one alias when you create a CA Identity Manager environment in the Management Console. (You may also specify a public alias.).

SiteMinder uses the environment name to name the objects that protect the environment. For example, when you specify the name *employees*, SiteMinder creates objects named *employeesobject_type*.

object_type

Defines the SiteMinder object, such as employees_ims_realm.

The following illustration shows two of the objects that SiteMinder creates:



Update an Alias in SiteMinder Realms

If you modify the protected or public alias in the Management Console, CA Identity Manager tries to update the alias names in the Policy Server. If CA Identity Manager cannot update the names, you can manually update them in one of the following interfaces:

- For CA SiteMinder Web Access Manager r12 or higher, use the Administrative UI.
- For CA eTrust SiteMinder 6.0 SP5, use the Policy Server User Interface.

Follow these steps:

1. Locate the realms for the CA Identity Manager Environment.

   These realms are created automatically (along with other required SiteMinder objects) when CA Identity Manager integrates with SiteMinder.

   The realms use the following naming convention:

   - *Identity Manager-environment*_ims_realm—Protects the User Console.
   - *Identity Manager-environment*_pub_realm—Enables support for public tasks, such as self-registration and forgotten password tasks. This realm appears only if you have configured a public alias.

     Note: If you are using the Policy Server User Interface to modify the realm, locate the policy domain (*Identity Manager-environment*Domain) for the CA Identity Manager environment first. The realms are located under the domain.

2. Modify the resource for the realm as follows:

   /iam/im/*new_alias*

   Do not remove the /iam/im/ that precedes the alias in the resource filter.

3. Save the changes.

Note: Modify CA Identity Manager Properties provide instructions on changing the alias in the Management Console.

Modify a SiteMinder Password or Shared Secret

When you install the CA Identity Manager Extensions to the Policy Server, supply the password for the SiteMinder administrator account that CA Identity Manageruses to communicate with the Policy Server.

You can change the password; however, the password must be encrypted. To encrypt a password, use the password tool which is provided with CA Identity Manager.

Note: Make sure the JAVA_HOME variable is defined for your environment before you change the SiteMinder password.

Follow these steps:

1. Encrypt the password as follows:

   a. From the command line, navigate to *admin_tools*\PasswordTool, where *admin_tools* is the installed location of the Administrative Tools, as in the following examples:

      - **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools\PasswordTool

      - **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager//tools/PasswordTool

   b. Type the following command:

      pwdtools *new_password*

      In this command, *new_password* is the password to encrypt.

      Note: For information about options for the pwdtools utility, enter the following command:

      pwdtools help

   c. Copy the encrypted password.

2. Complete the relevant step as follows:

   - If CA Identity Manager is running on a WebLogic application server, do the following tasks:

     a. In the WebLogic console, edit the WebLogic resource adapter in the policyserver_rar connector descriptor.

     b. Add the encrypted password as the value of the Password property.

   - If CA Identity Manager is running on a JBoss application server, do the following tasks:

     a. Open ra.xml from *JBoss_home*\server\default\deploy\iam_im.ear\policyserver_rar\META-INF.

     b. Add the encrypted password as the value of the Password config-property.

   - If CA Identity Manager is running on a WebSphere application server, complete the following tasks:

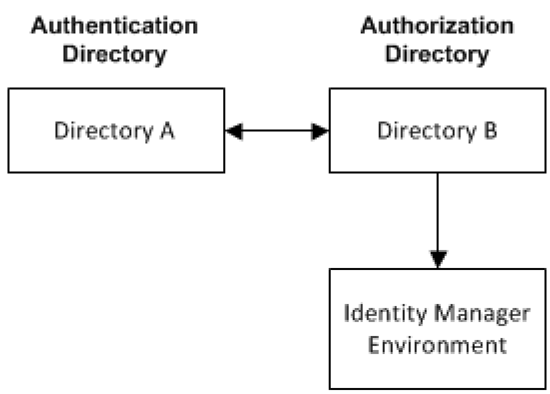
     a. In the WebSphere console, open ra.xml.

     b. Add the encrypted password as the value of the Password config-property.

3. Restart the application server.

Configure a CA Identity Manager Environment to Use Different Directories for Authentication and Authorization

An administrator may need to manage users whose profiles exist in a different user store from the one that is used for authenticating the administrator. In other words, when logging in to the CA Identity Manager Environment, the administrator must be authenticated using one directory and authorized to manage users in a second directory, as shown in the following illustration:



Follow these steps:

1. Log in to one of the following interfaces:
   - For CA SiteMinder Web Access Manager r12 or higher, log in to the Administrative UI.
   - For CA eTrust SiteMinder 6.0 SP5, log in to the Policy Server User Interface.

   Note: For information on using these interfaces, see the documentation for the version of SiteMinder that you are using.
2. Create two user directories.

   One directory references the authentication data (administrator profiles); the other directory references the authorization data (user profiles).
3. In the Management Console, create a CA Identity Manager Environment.

   Select the authorization directory as the CA Identity Manager directory.
4. In the interface for the version of SiteMinder used, add the authentication directory to the domain for the CA Identity Manager Environment that you created in the previous step.

   The domain and other objects that are required for SiteMinder are created automatically when you create an Environment and SiteMinder integrates with CA Identity Manager.

   The domain uses the following naming convention:

   *Identity Manager-environment*Domain
5. Make sure that this directory appears first in the list of directories that are associated with the domain.
6. Locate the *Identity Manager-environment*_ims_realm.
7. Map the authorization directory to the authentication directory in the Advanced section of the realm definition.
8. Locate the following *Identity Manager-environment*response_ims response.
9. Add response attributes to the responses as follows:

Field	Value
Attribute	Web-Agent-HTTP-Header-Variable
Attribute Kind	user attribute
Variable Name	sm_userdn
Attribute Name	SM_USERNAME

10. Save the changes.

    CA Identity Manager now uses different directories for authentication and authorization.

How to Improve the Performance of LDAP Directory Operations

Directory operations may take longer to process because all CA Identity Manager requests for the LDAP user directory are routed through a fixed set of connections.

To increase the throughput of CA Identity Manager requests to the user directory, configure SiteMinder to open multiple connections to the same directory. To do this procedure, add the LDAP server multiple times in the LDAP Directory Failover and Load Balancing Setup dialog box in the Policy Server user interface.

The number of times to enter the LDAP server (and the number of connections to create) depends on the load on CA Identity Manager.

Appendix A: FIPS 140-2 Compliance

This section contains the following topics:

FIPS Overview

The Federal Information Processing Standards (FIPS) 140-2 publication is a security standard for the cryptographic libraries and algorithms a product must use for encryption. FIPS 140-2 encryption affects the communication of all sensitive data between components of CA products and between CA products and third-party products. FIPS 140-2 specifies the requirements for using cryptographic algorithms within a security system protecting sensitive, unclassified data.

CA Identity Manager uses the Advanced Encryption Standard (AES) adapted by the US government. CA Identity Manager incorporates the RSA Crypto-J v3.5 and Crypto-C ME v2.0 cryptographic libraries, which have been validated as meeting the FIPS 140-2 Security Requirements for Cryptographic Modules.

Communications

FIPS encryption covers all data communications between CA Identity Manager and the following components:

- CA Identity Manager Server
- Provisioning Server
- Provisioning Manager and Clients
- C++ Connector Servers
- C++ Connector Server endpoints (if supported by endpoint)
- CA IAM Connector Servers (CA IAM CS)
- CA IAM CS endpoints (if supported by endpoint)
- Connector Xpress (if supported by endpoint)
- Windows Password Synchronization Agents
- Java Identity and Access Management (JIAM)

Installation

The CA Identity Manager installer allows you to configure CA Identity Manager to comply with FIPS 140-2.

All components in a CA Identity Manager environment must be FIPS 140-2 enabled for CA Identity Manager to support FIPS 140-2. You need a FIPS encryption key to enable FIPS 140-2 during installation. To generate a FIPS encryption key, run the Password Tool (pwdtools.bat/pwdtools.sh) from where you unpacked the install package. The Password Tool is available in the following locations:

- Windows: *package root*\PasswordTool\bin\pwdtools.bat
- UNIX: *package root*/PasswordTool/bin/pwdtools.sh

Note: The Password Tool is also installed in the following location:

```
C:\Program Files\CA\Identity Manager\IAM Suite\Identity
Manager\PasswordTool\pwdtools.bat
```

Important! Use the same FIPS 140-2 encryption key in all installations, and be sure to safeguard the Password Tool generated key file.

Connecting to SiteMinder

When connecting to CA SiteMinder during CA Identity Manager installation, be aware that FIPS mode and product version configurations are supported only as listed in the following table:

CA Identity Manager r12	SiteMinder	SiteMinder Version
FIPS-only mode	FIPS-only mode	r12
FIPS-only mode	FIPS-compatible mode	r12
Non-FIPS mode	FIPS-compatible mode	r12
Non-FIPS mode	Non-FIPS mode	r6

Key File Storage

CA Identity Manager uses the file system for FIPS encryption key storage. The CA Identity Manager administrator is responsible for protecting files from unauthorized access. The CA Identity Manager administrator can protect the files by setting the directory access permissions for specific group or user types, such as the user who is authorized to run CA Identity Manager.

The following table lists the location of the FIPS key files for each CA Identity Manager component.

Component	Installed Location
CA Identity Manager Server	*iam_im.ear*\config\com\netegrity\config\keys\FIPSkey.dat *iam_im.ear* is the installed location of CA Identity Manager on the application server.
Provisioning Server	*Provisioning Server install*\data\tls\keymgmt\imps_datakey
C++ Connector Server	*Provisioning Server install*\data\tls\keymgmt\imps_datakey
Password Synchronization Agent	*Provisioning Server install*\data\tls\keymgmt\imps_datakey

The Password Tool

The FIPS-compliant password tool utility, pwdtools.bat (or pwdtools.sh), can generate the encryption key during CA Identity Manager installation, from the command line.

Edit the pwdtools.bat/pwdtools.sh file before using the password tool and set the JAVA_HOME variable as required.

Important! CA Identity Manager does not support data migration or reencryption. Therefore, ensure that the encryption keys are not changed after installation.

This command has the following syntax:

```
pwdtools -{FIPSKEY|JSAFE|FIPS|RC2} -p plain text [-k <key file location>] [-f <encrypting parameters file>]
```

JSAFE

Encrypt a plain text value using the PBE algorithm.

Example:

`pwdtools -JSAFE –p` *`mypassword`*

Note: In the earlier versions, the password for the bootstrap administrator is stored in clear text. If you are upgrading or migrating to CA Identity Manager r12.6 SP1 or above, you need to manually encrypt the clear text password. Ensure that the JSAFE option is specified when using the tool and follow these steps:

1. After upgrading or migrating to CA Identity Manager r12.6 SP1 and above, go to the CA Identity Manager object store datatase and search for the following table:
   IM_AUTH_USER
2. Encrypt the clear text password using the password tool with JSAFE.
3. Replace the clear text with the encrypted password in the table.

FIPSKEY

For the installer, create a FIPS key file. You generate the key before installing CA Identity Manager.

Example:

`pwdtools -FIPSKEY –k C:\`*`keypath`*`\FIPSkey.dat`

Where *keypath* is the full path to the location where you want to store the FIPS key.

The password tool creates the FIPS key in the location specified. During installation, you provide the location of the FIPS key file to the installer.

Note: Be sure to secure the key by setting the directory access permissions for specific group or user types, such as the user who is authorized to run CA Identity Manager.

FIPS

Encrypt a plain text value using a FIPS key file. FIPS uses the existing FIPS key file.

Example:

`pwdtools -FIPS –p firewall -k C:\`*`keypath`*`\FIPSkey.dat`

Where *keypath* is the full path to the FIPS key directory.

Note: Use the same FIPS key file that you specified during installation.

RC2

Encrypt a plain text value using the RC2 algorithm.

Important! CA Identity Manager uses the FIPS key file to verify whether the application is to start in FIPS mode or in non-FIPS mode. Therefore, ensure that the key file is named FIPSKey.dat with the following application server deployment path:

```
iam_im.ear\config\com\netegrity\config\keys\FIPSkey.dat
```

where iam_im.ear is in the application server deployment directory, for example:

`jboss_home``\server\default\deploy`

FIPS Mode Detection

To determine whether CA Identity Manager is operating in FIPS mode or in non-FIPS mode, use the CA Identity Manager Environment status page.

To view the status page, enter the following URL in a browser:

`http://`*`server_name`*`/iam/im/status.jsp`

server_name

Determines the fully qualified domain name of the server where CA Identity Manager is installed, for example, myserver.mycompany.com. In this example, the complete URL is:

http://myserver.mycompany.com/iam/im/status.jsp

The FIPS status is displayed at the bottom of the page.

Note: You can also check if CA Identity Manager is operating in FIPS mode by locating the following key file:

```
/config/com/netegrity/config/keys/FIPSkey.dat
```

If this file exists, CA Identity Manager is operating in FIPS mode.

The password tool utility, pwdtools.bat (or pwdtools.sh) creates the FIPSkey.dat key file during CA Identity Manager installation.

Encrypted Text Formats

The algorithm name is added to the encrypted text as a prefix and it informs CA Identity Manager which algorithm was used for encryption.

In FIPS mode, the prefix is {AES}. For example, if you encrypt the text "password", the encrypted text is similar to the following example:

{AES}:eolQCTq1CGPyg6qe++0asg==

In non-FIPS mode (or JSAFE mode), depending on the algorithm, the prefix (algorithm tag) is {PBES} or {RC2}. For example, if you encrypt the text "password", the encrypted text is similar to this:

{PBES}:gSex2/BhDGzEKWvFmzca4w==

You can create dynamic keys using the Secret Keys task under System. If you define dynamic keys, the Key ID is inserted between an algorithm tag and tag delimiter (‘:’). The absence of a Key ID in the encrypted data indicates that hard-coded key was used for encryption. This can be used for backward compatibility or if no dynamic keys are defined for the given algorithm.

Encrypted Information

The following CA Identity Manager information is encrypted:

- Passwords in the datasource configuration for Jboss
- Forgotten password recovery information
- Provisioning Server callback secret

- Workflow session information
- Policy Server connection information

CA Identity Manager uses JSafe libraries to encrypt and decrypt data. To ensure that the libraries are not tampered with, CA Identity Manager uses CryptoJ self-test code during startup.

After the self-tests run, CryptoJ messages, which report the status of the test, appear in the application server log. The log contains one of the following messages:

```
[ims.default] * CryptoJ was initialized properly.
```

```
[ims.default] !!! CryptoJ was not initialized properly. !!!
```

FIPS Mode Logging

The following CA Identity Manager components indicate in log files whether FIPS mode is enabled:

- CA Identity Manager Server
- Provisioning Server
- C++ Connector Server
- CA IAM CS
- Provisioning Manager
- Password Synchronization Agent

In all cases, the log entry indicating that FIPS mode is enabled ends with the following string:

FIPS 140-2 MODE: ON

Appendix B: Replacing CA Identity Manager Certificates with SHA-2 Signed SSL Certificates

SHA-2 SSL certificate hashing is a cryptographic algorithm developed by the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA). SHA2 certificates are more secure than all previous algorithms. In CA Identity Manager, you can configure SHA-2 signed SSL certificates in place of certificates that are signed with the SHA-1 hash function.

Note: For more information about configuring SSL certificates, see the *Installation Guide*.

The following table shows the path location on the CA Identity Manager server where you can place the SHA-2 signed certificates:

Certificates	Install Location	Description
Provisioning Server Certificate	[Provisioning Server install dir]/data/tls/server/eta2_servercert.pem [Provisioning Server install dir]/data/tls/server/eta2_serverkey.pem *cs_install*/ccs/data/tls/server/eta2_servercert.pem *cs_install*/ccs/data/tls/server/eta2_serverkey.pem *cs_install*/jcs/conf/eta2_server.p12	Used by the Provisioning Server in .pem format and by CA IAM CS in .p12 format (including signed cert, private key and root CA cert). **Note**: Import the eta2_server.p12 into *cs_install*/jcs/conf/ssl.keystore under the alias eta2_server and remove the existing entry. The ssl.keystore password is the password of the connector server that is supplied during the install.

Certificates	Install Location	Description
Provisioning Client Certificate	[Provisioning Server install dir]/data/tls/client/eta2_clientcert.pem [Provisioning Server install dir]/data/tls/client/eta2_clientkey.pem [Provisioning Manager install dir]/data/tls/client/eta2_clientcert.pem [Provisioning Manager install dir]/data/tls/client/eta2_clientkey.pem *cs_install*/ccs/data/tls/client/eta2_clientcert.pem *cs_install*/ccs/data/tls/client/eta2_clientkey.pem *cs_install*/jcs/conf/eta2_client.p12	Used by the Provisioning Server in .pem format and by CA IAM CS in .p12 format (including signed cert, private key and root CA cert).
Provisioning Directory Trusted Certificate	*cadir_install*/config/ssld/impd_trusted.pem	Used by CA Directory in .pem format. It must contain certificate content in the following structure: -----BEGIN CERTIFICATE----- Cert contents -----END CERTIFICATE-----
Provisioning Directory Personality Certificate	*cadir_install*/config/ssld/personalities/impd-co.pem *cadir_install*/config/ssld/personalities/impd-inc.pem *cadir_install*/config/ssld/personalities/impd-main.pem *cadir_install*/config/ssld/personalities/impd-notify.pem *cadir_install*/config/ssld/personalities/impd-router.pem	Used by CA Directory in .pem format.

Certificates	Install Location	Description
Root CA Certificate	[Provisioning Server install dir]/data/tls/et2_cacert.pem [Provisioning Manager install dir]/data/tls/et2_cacert.pem *cs_install*/ccs/data/tls/ et2_cacert.pem *conxp_install*/lib/jiam.jar [Application Server install dir]/iam_im.ear/library/jiam.jar	Certificate is Imported into Connector Xpress keystore located at [Connector Xpress install dir]/conf/ssl.keystore. The certificate must also be imported into the jiam.jar keystore. To import, extract the jar, import the certificate into admincacerts.jks and then repackage the jar contents. The keystore password of admincacerts.jks is "changeit". Verify that all copies of jiam.jar are replaced.

Useful Commands

The OpenSSL program is a command line tool for using the various cryptography functions of OpenSSL's library. This tool is shipped with IMPS located in [Provisioning Server install dir]/bin.

The following table shows few useful commands of OpenSSL program to execute various commands that are related to managing certificates:

Commands	Description
`openssl x509 –in cert.pem –text –noout`	Prints contents of .pem certificate.
`openssl.exe pkcs12 -in my.pkcs12 –info`	Prints contents of .p12 file.

Commands	Description
`openssl.exe pkcs12 -export -chain –inkey key.pem –in cert.pem -CAfile cacert.pem -out my.p12`	Converts .pem cert/keypair to .p12.

Commands	Description
`keytool –list –v -keystore my.keystore`	Prints contents of a java keystore.
`keytool –list –v –alias myalias –keystore my.keystore`	Prints contents of a specific alias in a java keystore

Commands	Description
`keytool –delete –alias myalias –keystore my.keystore`	Deletes an alias from a java keystore

Commands	Description
`keytool –importkeystore –destkeystore my.keystore –srckeystore src.p12 –srcstoretype PKCS12 –srcalias 1 –destali`	Imports a .p12 file into a java keystore.

Commands	Description
`keytool -import -trustcacerts -alias myrootca -file rootcacert.pem -keystore my.keystore`	Imports a .pem root ca certificate into a java keystore.

Index

W