

CA Identity Manager

Implementation Guide

r12



This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of the Documentation for their own internal use, and may make one copy of the related software as reasonably required for back-up and disaster recovery purposes, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the provisions of the license for the Product are permitted to have access to such copies.

The right to print copies of the Documentation and to make a copy of the related software is limited to the period during which the applicable license for the Product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

EXCEPT AS OTHERWISE STATED IN THE APPLICABLE LICENSE AGREEMENT, TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

The use of any product referenced in the Documentation is governed by the end user's applicable license agreement.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Copyright © 2008 CA. All rights reserved.

CA Product References

This document references the following CA products:

- CA Identity Manager
- CA SiteMinder® Web Access Manager
- CA Security Command Center (SCC)
- CA Audit
- eTrust® Directory, also known as CA Directory

Contact Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at <http://ca.com/support>.

Contents

Chapter 1: Managing Identities and Access	9
User Management and Application Access	9
Role-Based Entitlements	10
Admin Roles for User Account Management	11
Profile Management at the Attribute Level	12
Workflow Approval of Admin Tasks	13
Provisioning Roles for Additional Accounts	14
Password Management	15
Self Service Options for Users	16
Identity Manager Customization and Extensibility	16
 Chapter 2: Addressing Business Needs	 19
Processing Business Changes	19
Complying with Business Policies	20
User Entitlement Certification	20
Compliance Reports	21
Transforming Data in the User Store	23
Logical Attribute Handlers	23
Applying Custom Business Logic	23
Business Logic Task Handler Considerations	24
Workflow Process Considerations	25
Approving Business Changes	25
 Chapter 3: Identity Manager Architecture	 27
Identity Manager Components	27
Servers	27
User Store and Provisioning Directory	28
Databases	30
Connector Components	31
Additional Components	33
Sample Identity Manager Installations	35
Basic Installation	35
Installation with Provisioning Components	36
Installation with SiteMinder Policy Server	37

Chapter 4: Planning Your Implementation **39**

Decide What to Manage	39
User Identities	39
Provisioning Accounts from Other Applications	40
Determine Audit Requirements	41
Identity Manager Auditing Considerations	42
CA Audit Considerations	43
CA Security Command Center (SCC) Considerations	43
Decide User Store Requirements	43
How to Choose a User Store Solution	44
Managing Multiple User Stores	44
Select Components to Install	45
Decide Hardware Requirements	46
Requirements for Basic Installations	47
Additional Requirements for Provisioning	47
Additional Requirements for SiteMinder Integration	48
Choose a Method to Import Users	48
How to Import Users into a New User Store	48
Synchronize Global Users with the Identity Manager User Store	52
Develop a Deployment Plan	53
Deploy Self-Service and Password Management	53
Deploy Identity Policies	54
Deploy Workflow Approvals	56
Deploy Delegated Administration for Users, Groups and Organizations	57
Deploy Delegated Administration for Roles	57

Chapter 5: Integrating with SiteMinder **59**

SiteMinder Integration	59
SiteMinder Authentication	60
Password Policies with SiteMinder	61

Chapter 6: Optimizing Identity Manager **63**

Identity Manager Performance	63
Role Optimizations	64
How Role Evaluation Affects Performance at Login	64
Role Objects and Performance	65
Optimize Role Policy Evaluation	67
Guidelines for Policy Rule Creation	68
Task Optimizations	72
Task Scope Evaluation and Performance	73

How Identity Manager Renders Relationship Tabs	73
Relationship Tabs and Performance	75
Guidelines for Optimizing Tasks	76
Guidelines for Group Member\Administrator Optimizations	77
Identity Policy Optimizations	79
How Users and Identity Policies Are Synchronized	79
Design Efficient Identity Policies	81
Limit the Tasks that Trigger User Synchronization	82
Optimize Identity Policy Rule Evaluation	82
User Store Tuning	83
Tuning for Provisioning Components	84
Runtime Components Tuning	85
Tuning Identity Manager Databases	85
JMS Settings	86

Chapter 7: Transitioning From eTrust Admin to Identity Manager 91

Develop a Transition Plan	91
Business Changes in Identity Manager	92
Provisioning Terminology Changes	93
eTrust Admin Management Interfaces	93
SAWI/DAWI Considerations	94
Password Management Considerations	94
IA Manager Considerations	94
Provisioning Manager Considerations	94
Batch Processing in Identity Manager Implementations	95
Custom Endpoint Connectors	96
Next Steps	96

Appendix A: Third-Party Acknowledgements 97

Apache	97
ANTLR	101
DOM4J	102
Jaxen	103
Rhino	104
JDOM	113
SAXPath	115

Index 117

Chapter 1: Managing Identities and Access

This section contains the following topics:

[User Management and Application Access](#) (see page 9)

[Role-Based Entitlements](#) (see page 10)

[Admin Roles for User Account Management](#) (see page 11)

[Provisioning Roles for Additional Accounts](#) (see page 14)

[Password Management](#) (see page 15)

[Self Service Options for Users](#) (see page 16)

[Identity Manager Customization and Extensibility](#) (see page 16)

User Management and Application Access

The typical Information Technology (IT) department faces a constant demand to maintain user accounts. IT administrators must address urgent needs of users, such as resetting forgotten passwords, creating new accounts, and providing supplies and office equipment.

Simultaneously, IT administrators must provide users with various levels of access to applications. For example, a department manager generates purchase orders and needs an account in a financial application.

To address the escalating demands on IT, Identity Manager provides an integrated method of managing users and their access to applications, including:

- Assignment of privileges through roles. Specifically:
 - Roles that enable administrators to create and maintain user accounts
 - Roles that provision additional accounts to existing users (requires provisioning support)
- Delegation of the management of users and application access
- Self-service options so users can manage their own accounts
- Integration of business applications with Identity Manager
- Options to customize and extend Identity Manager

Role-Based Entitlements

With Identity Manager, you assign privileges to users by assigning roles. A *role* contains tasks that correspond to application functions in Identity Manager or account templates that correspond to additional accounts. When you assign a role to a user, that user can perform the tasks contained in the role or use the accounts associated with the role.

Identity Manager provides these types of roles:

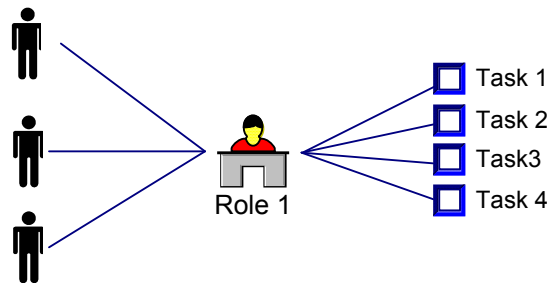
- User management roles are called *admin roles*.
- Account assignment roles are called *provisioning roles*.

Roles simplify privilege management. Instead of associating a user with each task that he performs or each account that he needs, you can assign a role to the user. The user can perform the tasks in the role or use the accounts associated with the role.

You can then edit the role by adding tasks or account templates, which define the accounts. Every user who has the role can now perform the new task or use the new account. If you remove a task or account template from a role, the user can no longer perform that task or use the account.

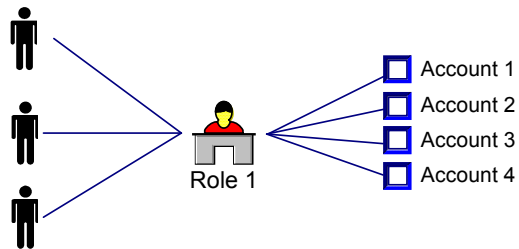
Tasks enable users to perform Identity Manager functions, such as modifying a profile.

The following illustration shows several tasks which are combined into a single admin role and assigned to multiple users:



For a provisioning role, you combine several accounts, such as an email account, a database account, and an Active Directory account. You can assign the role to several users, who each need these accounts.

The following illustration shows several accounts which are combined into a single provisioning role and assigned to multiple users.



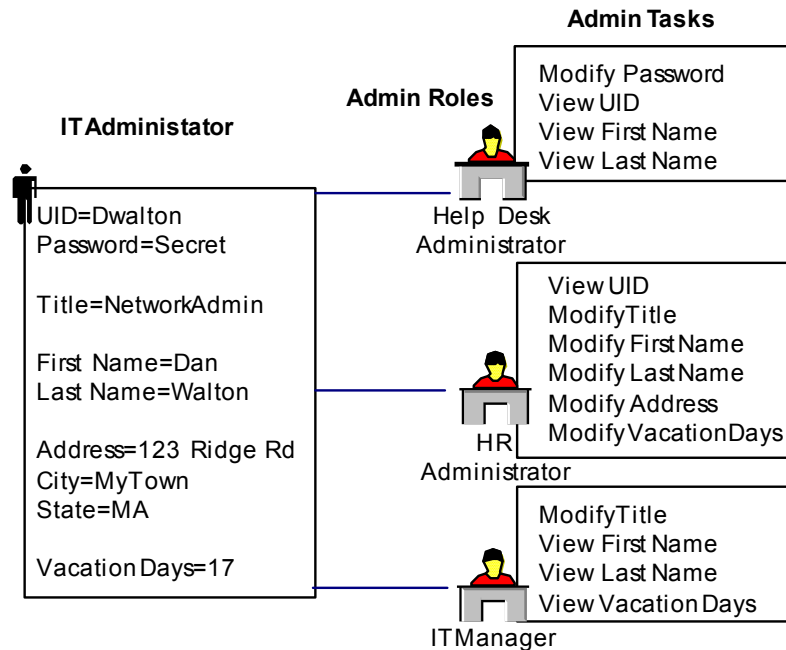
Note: In this figure, each user receives four accounts, when you assign the provisioning role to that user.

Admin Roles for User Account Management

In Identity Manager, you manage user store objects (users, groups, and organizations) through admin roles. You also use admin roles to manage the roles and tasks through which you manage user store objects. For example, you use admin roles to modify profile attributes of users, give users options for managing their own accounts, and to approve tasks that use workflow.

Profile Management at the Attribute Level

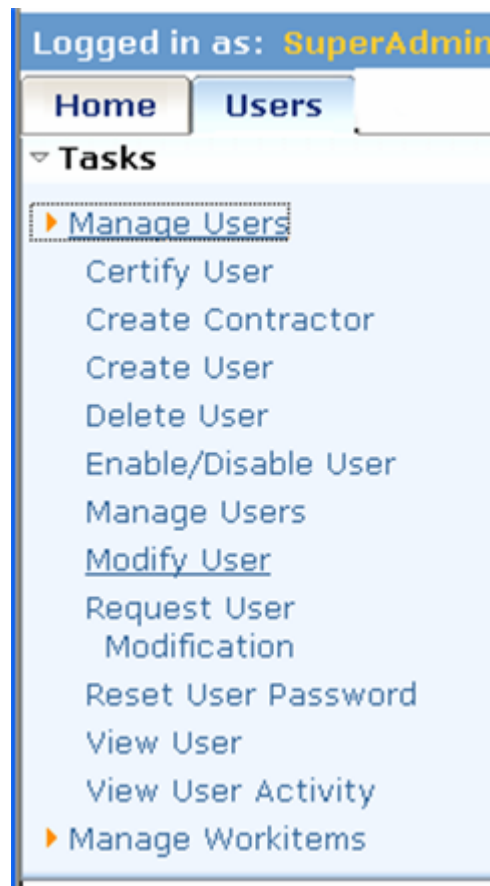
You can create admin roles for different administrators who need to read or write different profile attributes. For example, a company may have several employees who perform operations on user profiles, each accessing different attributes. The following figure shows three roles and their associated tasks. Each role has different access to profile attributes.



In this example, three roles can manage different attributes for the same user, Dan Walton:

- A Help Desk administrator views user names and addresses and resets user passwords.
- A Human Resources administrator modifies user IDs, user names, addresses, titles, and number of vacation days.
- An IT manager modifies the title of users and views their name and number of vacation days.

Whatever roles you have when you log in to Identity Manager, a series of tabs, called categories, appear based on the admin role assigned to your Identity Manager account. You click a tab to see the tasks that you can perform in that category as shown in the following figure:



The categories and the tasks in those categories that a user sees are determined by the user's admin roles.

Workflow Approval of Admin Tasks

To help automate business processes, you can design an admin task to generate a workflow process. A *workflow process* automates a well-defined procedure that a company repeats frequently. Identity Manager includes the WorkPoint workflow engine.

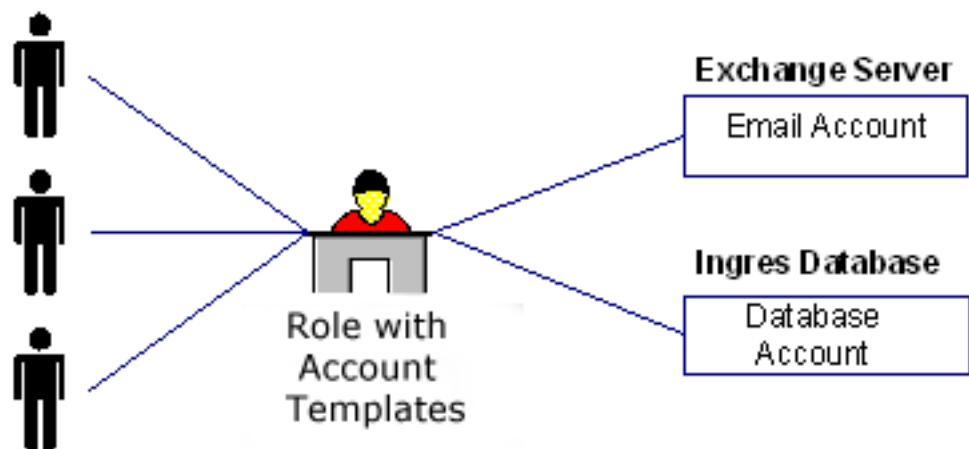
Workflow processes are triggered by Identity Manager events which are part of an admin task. For example, the Create User task includes events called `CreateUserEvent` and `AddToGroupEvent`. When an event occurs, the workflow engine can:

- **Require approvals**--An approver must approve an event, such as modifying a user profile, before Identity Manager updates a user store. Approvers are administrators who have the Approver role for a particular task.
- **Send notifications**--The workflow engine can notify users of an event's status at different stages of a process, such as when a user initiates an event or when an event is approved.
- **Generate work lists**--Work lists specify the tasks that a particular user must perform. The workflow engine updates administrators' work lists automatically.

For common events, you can use the workflow processes supplied with Identity Manager. Alternatively, you can create custom workflow processes.

Provisioning Roles for Additional Accounts

In Identity Manager, you provide additional accounts to users by using provisioning roles. Provisioning roles contain account templates, which define accounts that exist in managed endpoints, such as an email server. Once you have users in Identity Manager, you can assign provisioning roles to some of those users. The user receives the accounts defined by the templates in the role.



The account templates define the characteristics of the account. For example, an account template for an Exchange account might define the size of the mailbox. The account templates also define how user attributes are mapped to accounts.

To be able to use provisioning roles, you must install the Provisioning Server with the Identity Manager server. Then, you create account templates in the Provisioning Manager, a user interface for managing the Provisioning Server.

Password Management

Identity Manager includes several features for managing user passwords:

- Password Policies—These policies manage user passwords by enforcing rules and restrictions governing password expiration, composition, and usage.
Note: For advanced password policies, configure integration with SiteMinder. For more information, see the *Installation Guide*.
- Password Managers—Administrators who have the Password Manager role can reset a password when a user calls the Help desk.
- Self-Service Password Management—Identity Manager includes several self-service tasks that allow users to manage their own passwords. These tasks include:
 - Self Registration—Users specify a password when they register at a corporate web site.
 - Change My Password—Users can modify their passwords without help from IT or Help Desk personnel
 - Forgotten Password—Users can reset or retrieve a forgotten password after Identity Manager verifies their identity.
 - Forgotten User ID—Users can retrieve a forgotten user ID after Identity Manager verifies their identity.
- Password Synchronization (for use with provisioning only)—Password changes are synchronized in Identity Manager and in accounts on target systems called endpoints. New passwords are verified against Identity Manager password policies.

Self Service Options for Users

To further reduce the IT workload, Identity Manager includes features for registering new users and supplying a forgotten password. These features require no administrator involvement. The user gains access to Identity Manager through a *public console*, which requires no login account. Through this console, a user can self-register at a site or request a reminder about a forgotten password.

To save the time of IT administrators, Identity Manager users can manage their own accounts. Because users have a self-management role, they can:

- Maintain personal information
- Change their own password
- Join self-subscribing groups

Identity Manager Customization and Extensibility

You customize these Identity Manager features:

- The Identity Manager directory, which describes a user store structure to Identity Manager.
- The appearance and functionality of the user interface.
- User entry screens, which determine the fields and layout of each task screen.
- Validation of user data entry, through regular expression, JavaScript, or Java implementations.
- Workflow, which defines automated workflow processes. Create or modify processes by linking approvers and actions in the WorkPoint Process Designer.
- Email messages, which inform users of a task's status.
- Task submission, which can be sent by a third-party application to the Identity Manager Task Execution Web Service (TEWS). TEWS processes the remote task request. Remote task requests comply with WSDL standards.

You can extend Identity Manager's functionality using the following APIs:

- Logical Attribute API—Enables you to display an attribute differently than how it is stored physically in a user directory.
- Business Logic Task Handler API—Allows you to perform custom business logic during data validation or transformation operations.

- Workflow API—Provides information to a custom script in a workflow process. The script evaluates the information and determines the path of the workflow process accordingly.
- Participant Resolver API—Enables you to specify the list of participants who are authorized to approve a workflow activity.
- Event Listener API—Enables you to create a custom event listener that listens for a specific Identity Manager event or group of events. When the event occurs, the event listener can perform custom business logic.
- Notification Rule API—Lets you determine the users who should receive an email notification.
- Email Template API—Includes event-specific information in an email notification.

Note: For more information on the Identity Manager APIs, see the *Programming Guide for Java*.

When Identity Manager includes provisioning, you can also extend provisioning functionality as follows:

- Custom connectors—Enable communication between a Provisioning Server and an endpoint system. The code that makes up a connector can include a GUI plug-in, server plug-in, and agent plug-in.

A dynamic connector can be generated by Connector Xpress, and a custom static connector can be developed in Java or C++.
- Program exits—Let you reference custom code from the Provisioning Server process flow.

Note: For more information about extending provisioning functionality, see the *Programming Guide for Provisioning*.

Chapter 2: Addressing Business Needs

This section contains the following topics:

- [Processing Business Changes](#) (see page 19)
- [Complying with Business Policies](#) (see page 20)
- [Transforming Data in the User Store](#) (see page 23)
- [Applying Custom Business Logic](#) (see page 23)
- [Approving Business Changes](#) (see page 25)

Processing Business Changes

You can automate the processing of certain identity management tasks by using identity policies. An identity policy is a set of business changes that occurs when a user meets a certain condition or rule. You can use identity policy sets to:

- Automate certain identity management tasks, such as assigning roles and group membership, allocating resources, or modifying user profile attributes.
- Enforce segregation of duties. For example, you can create an identity policy set that prohibits members of the Check Signer role from having the Check Approver role, and restricts anyone in the company from writing a check over \$10,000.
- Enforce compliance. For example, you can audit users who have a certain title and make more than \$100,000.

Identity policies that enforce compliance are called *compliance policies*.

The business changes associated with an identity policy include:

- Assigning or revoking roles, including provisioning roles (when Identity Manager includes provisioning)
- Assigning or revoking group membership
- Updating attributes in a user profile

For example, a company may create an identity policy which states that all Vice Presidents belong to the Country Club Member group and have the role Salary Approver. When a user's title changes to Vice President and that user is synchronized with the identity policy, Identity Manager adds the user to the appropriate group and role. When a Vice President is promoted to CEO, she no longer meets the condition in the Vice President identity policy so the changes applied by that policy are revoked, and new changes based on the CEO policy are applied.

The change actions that occur based on an identity policy contain events which can be placed under workflow control and audited. In the previous example, the Salary Approver role grants significant privileges to its members. To protect the Salary Approver role, the company can create a workflow process that requires a set of approvals before the role is assigned, and they can configure Identity Manager to audit the role assignment.

To simplify identity policy management, identity policies are grouped together in an identity policy set. For example, the Vice President and CEO policies may be part of the Executive Privileges identity policy set.

Complying with Business Policies

Compliance is a corporate governance that includes a wide range of procedures that ensure a company and its employees comply with business policies. These compliance procedures often involve documenting, automating, and auditing the allocation of entitlements to applications and systems.

Identity Manager includes the following features, which support compliance management:

- **Identity policies**

You can create a compliance policy, a type of [identity policy](#) (see page 19), which prohibits users from having certain privileges if they have other privileges. For example, you can prohibit users who can approve checks from issuing checks.

Compliance policies enforce a segregation of duties in your environment.

- **User entitlement certification**

Forces users, such as managers, to review and certify the roles of the users they manage.

- **Compliance reports**

Identity Manager includes sample reports that display the compliance status for users in your environment. Using these reports, you can see which users are not compliant with your business policies.

User Entitlement Certification

User certification requires business managers to periodically review and approve the roles of the users they manage. Using the certification tasks, administrators can:

- Identify a set of users that require certification
- Specify a certification period, such as a fiscal quarter, in which all users must be certified

- Notify a user, such as a manager, that he has pending certifications
- Allow a privileged user, such as a manager, to certify or remove privileges
- Disable privileges for users who have not been certified after a certain period of time

Compliance Reports

Identity Manager includes the sample reports in the following table that you can use to monitor compliance with corporate business policies.

Report	Description
Role Members	Displays the roles in the report database and lists the members of those roles
Roles	Displays the following information for each role in the report database: ■ Tasks associated with the role ■ Member policies and role members ■ Administrator policies and role administrators ■ Owner policies and role owners
Tasks Roles	Displays the tasks in the reporting database and the roles with which they are associated
User Roles	Displays the users in the reporting database and lists each user's roles
Non-Standard Accounts Trend	Displays non-standard accounts trends for orphan accounts, system accounts, and exception accounts
Non-Standard Accounts	Displays all orphan, system, and exception accounts
Orphan Accounts	Displays all endpoint accounts with no global user in the Provisioning Server
Policies	Displays all identity policies

Report	Description
User Profile	Displays the following information for users: ■ Name ■ User ID ■ Groups where the user is a member or administrator ■ Roles where the user is a member, administrator, or owner
Endpoint Accounts	Displays accounts per endpoint (you can choose which endpoint to view)
Role Administrators	Displays roles and their administrators
Role Owners	Displays roles and their owners
Snapshots	Displays all exported snapshots
User Account	Displays a list of users and their accounts
User Entitlements	Displays user's roles, groups and accounts
User Policy Sync Status	Displays the user's status per policy (which policies should be allocated, deallocated or reallocated)

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

Transforming Data in the User Store

In some cases, you may want Identity Manager to transform data before it is stored in the user store. For example, you may want to store information in a different format than it is entered, or you may want changes applied when certain types of information are present.

Identity Manager includes the following features for transforming data:

- Identity Policies
- Logical Attribute Handlers

Note: You can also use identity policies and logical attribute handlers to implement custom business logic.

Logical Attribute Handlers

Logical attribute handlers are custom Java code that transform user attribute values used on Identity Manager task screens. Using logical attribute handlers, you can control how a physical attribute is displayed on a task screen. You can also use logical attribute handlers to transform a display value, such as cost, on the task screen to one or more physical attributes, such as unit price and quantity, that are stored in the user store.

Note: For more information about logical attribute handlers, see the *Programming Guide for Java*.

Applying Custom Business Logic

You can customize Identity Manager to implement the business logic that your company requires. Identity Manager includes the following options for implementing custom business logic:

- Identity Policies—You can use identity policies to define a set of business changes that occur when a user meets a certain condition or rule. For example, identity policies can automate certain identity management tasks, such as assigning roles, or enforce business rules, such as preventing users from signing and approving checks over \$20,000.

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

- Logical Attribute Handlers—You can associate these handlers with Identity Manager task screens to control the display and modification of attribute values.

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

- **Business Logic Task Handlers**—Enable you to perform custom business logic, such as the following, during data validation operations for an Identity Manager task:
 - Enforcing custom business rules (for example, an administrator cannot be allowed to manage more than five groups).
 - Validating customer-specific task screen fields (for example, the value of an Employee ID field must exist in the master Human Resources database).

Business logic task handlers can be implemented in Java or JavaScript.

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

- **Workflow**—Allows you to create custom process definitions, which are associated with an Identity Manager event.

Note: Before deciding whether to implement business logic in a business logic task handler or a workflow process, see the following sections:

- [Business Logic Task Handler Considerations](#) (see page 24)
- [Workflow Process Considerations](#) (see page 25)

Business Logic Task Handler Considerations

Business Logic Task Handlers perform business logic validation during the synchronous processing phase of the task, which occurs prior to event generation. This allows you to:

- Perform task-level validation. For example, you can add or remove members of a group based on their office location, which is specified in the user profile screen.
- Prevent a task from being submitted if the validation fails.
- Automatically transform all of the information on a task screen so that it conforms to your business policies prior to task submission.

Note: You should not implement activities that take a long time to complete in a Business Logic Task Handler. Long running activities delay the submission of the task and are not well-suited for the synchronous phase where user interaction occurs. Instead, use a workflow process, which executes during the asynchronous phase of the task.

Workflow Process Considerations

Workflow processes are called during the asynchronous phase of the task and are associated with the execution of individual events. This allows you to:

- Execute approval activities based on the individual event data
- Execute long running custom business logic activities

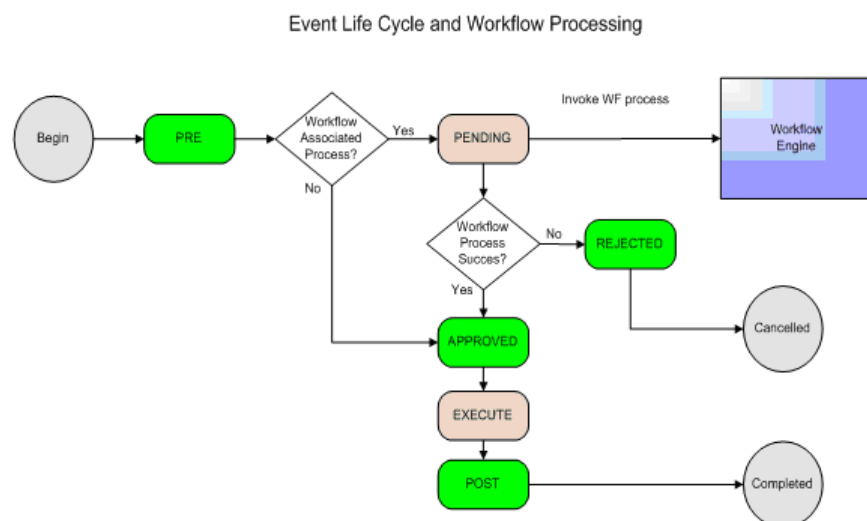
While the Workflow API allows you to obtain task-level data from a Workflow Activity, typically you are operating in the context of that specific event under workflow.

Approving Business Changes

Workflow describes a process that consists of one or more steps that must be performed in order to accomplish some business objective, such as executing a hiring procedure, or obtaining a user's credit score from an external system. Typically, one of the steps in a workflow process involves approving or rejecting the business change.

In Identity Manager, a workflow process is associated with an event, an action that occurs during task processing. When an event enters the Pending state in its lifecycle, Identity Manager invokes any associated workflow process and pauses the event execution until the process completes. Identity Manager then performs or rejects the event based on the results of the workflow process.

This sequence is shown in the following diagram:



Identity Manager includes the InSession WorkPoint workflow engine for creating and managing workflow processes.

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

Chapter 3: Identity Manager Architecture

This section contains the following topics:

[Identity Manager Components](#) (see page 27)

[Sample Identity Manager Installations](#) (see page 35)

Identity Manager Components

An Identity Manager implementation may include some or all of the following components:

- Servers
- User Stores
- Databases
- Connectors

Servers

An Identity Manager implementation includes one or more types of servers, depending on the functionality you need.

Identity Manager Server (required)

Executes tasks within Identity Manager. The J2EE Identity Manager application includes the Management Console and the User Console.

Identity Manager Provisioning Server

Manages accounts on endpoint systems.

This server is required if the Identity Manager installation will support account provisioning.

Note: You must have the Provisioning Directory installed remotely (or locally for a demonstration environment only) on an eTrust Directory Server before installing the Provisioning Server.

SiteMinder Policy Server

Provides advanced authentication for Identity Manager, and provides access to SiteMinder features, such as Password Services and Single Sign-On.

This server is optional.

User Store and Provisioning Directory

To provide options for managing users and automatic provisioning of additional accounts for those users, Identity Manager coordinates two user stores:

- The *Identity Manager user store*, the user store maintained by Identity Manager. Typically, this is an existing store that contains the user identities that a company needs to manage.

The user store can be an LDAP directory or a relational database.

In the Management Console, you create an Identity Manager Directory object to connect to the user store and to describe the user store objects that Identity Manager will maintain.

- The *Provisioning Directory*, the user store maintained by the Provisioning Server.

It is an instance of eTrust Directory and includes global users, which associate users in the Provisioning Directory with accounts on endpoints such as Microsoft Exchange, Active Directory, and Ingres.

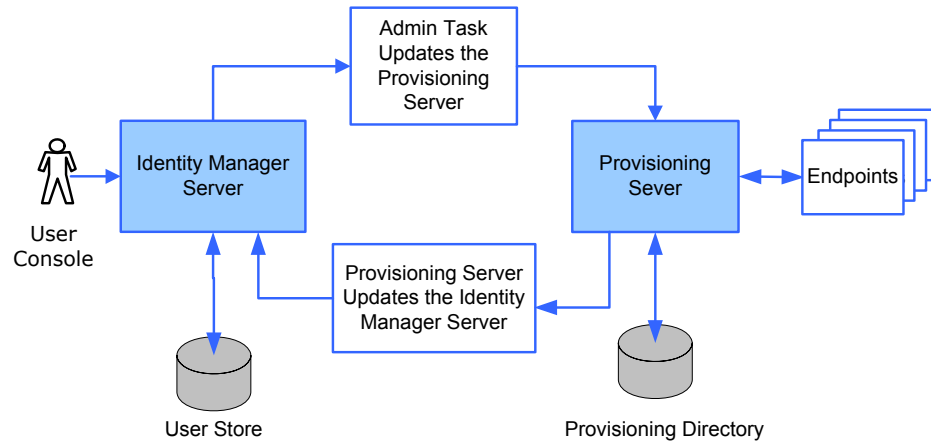
Only some Identity Manager users have a corresponding global user. When a Identity Manager user receives a provisioning role, the Provisioning Server creates a global user.

Separate User Store and Provisioning Directories

The following figure shows a separate user store and Provisioning Directory. In this figure:

- An Identity Manager administrator uses an admin task that edits a user in the user store, which affects the Provisioning Directory.
This change may also update an endpoint (such as an email server) which has a connector to the Provisioning Server.
- A change made in the Provisioning Server (or an endpoint with a connector to the Provisioning Server) updates the Identity Manager user store and Provisioning Directory.

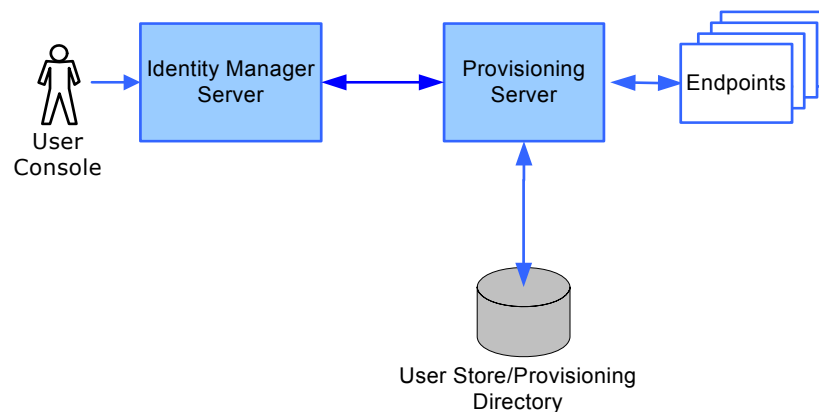
For example, an endpoint, such as a Human Resources application, might update the email addresses of users.



Combined User Store and Provisioning Directory

If you select an instance of eTrust Directory for both user stores, the directory functions as both a user store and Provisioning Directory. Users you create in Identity Manager are stored in that directory, but you can still modify that directory as a Provisioning Directory. For example, you can select the user attribute to use in the Provisioning Server for a specific user attribute used by Identity Manager.

The following figure shows the use of one directory for both the user store and Provisioning Directory.



In this situation, changes to the user store/Provisioning Directory can be initiated by Identity Manager, the Provisioning Server, or an endpoint.

Databases

Identity Manager uses data sources to connect to databases that store information required to support Identity Manager functionality. These databases can reside in a single physical instance of a database, or in separate instances.

Object Database (required)

Contains Identity Manager configuration information.

Task Persistence Database (required)

Maintains information about Identity Manager activities and their associated events over time. This allows the system to accurately track Identity Manager activities, even if you restart the Identity Manager Server.

Workflow Database

Stores workflow process definitions, jobs, scripts, and other data required by the Workflow Engine.

Audit Database

Provides a historical record of operations that occur in an Identity Manager environment.

Note: You can configure the amount and type of information that Identity Manager stores in the audit database. See the *Configuration Guide* for more information.

Reporting Database

Stores snapshot data, which reflects the current state of objects in Identity Manager at the time the snapshot is taken. You can generate reports from this information to view the relationship between objects, such as users and roles.

When you use the Identity Manager Installer, Identity Manager configures a connection to a single database, called the Identity Manager Database, which contains the tables for each database type.

Note: You can create a data store for task persistence, workflow, auditing, or reporting in a separate database and configure Identity Manager to connect to it. For more information, see the *Installation Guide*.

Connector Components

A connector is the software interface to an endpoint. The Provisioning Server uses the connector to communicate with the endpoint. It translates Provisioning Server actions into changes on the endpoint, such as "Create a new email account on a Microsoft Exchange endpoint."

Examples of endpoints are UNIX workstation, Windows PC, or an application such as Microsoft Exchange (for email).

A connector server works with multiple endpoints. For example, if you have many UNIX workstation endpoints, you might have one Connector Server that handles all connectors that manage UNIX accounts. Another connector server might handle all connectors that request Windows accounts.

The distributed connector server works with multiple Connector Servers. It provides load balancing when one connector server is busy and high availability when a connector server is down.

Connector Servers

A connector server is a Provisioning Server component that manages connectors. It can be installed on the Provisioning Server system or on a remote system.

There are two types of connector servers:

- The Java Connector Server (Java CS) manages connectors written in Java
- The C++ Connector Server (CCS) manages connectors written in C++

C++ Connector Server

The C++ *Connector Server* is a connector server that manages C++ connectors. It can be installed on the Provisioning Server or on a remote system. The C++ Connector Server provides an object-oriented application framework that simplifies development of connectors, which are responsible for communication between the C++ Connector Server and the endpoint.

Java Connector Server

The *Java Connector Server (Java CS)* is a server component which handles hosting, routing to, and management of Java connectors. The Java CS provides a Java alternative to the C++ Connector Server. It is architecturally and functionally similar to the C++ Connector Server, except that it has a Java API instead of a C++ API, which allows your connectors to be implemented in Java. In addition, the Java CS is data-driven rather than code-driven, which allows more functionality to be addressed by the container (or Java CS) instead of by connectors themselves.

The Provisioning Server handles provisioning of users, and then delegates to connectors (using the C++ Connector Server or Java Connector Server) to manage endpoint accounts, and groups.

Connectors and Agents

Identity Manager Connectors run as part of the wider Provisioning Server architecture and communicate with the systems managed in your environment. A connector acts as a gateway to a native endpoint type system technology. For example, machines running Active Directory Services (ADS) can be managed only if the ADS connector is installed on a Connector Server with which the Provisioning Server can communicate. Connectors manage the objects that reside on the systems. Managed objects include accounts, groups, and optionally, endpoint type specific objects.

Connectors are installed on the Connector Server and some components are installed on the Provisioning Server (for example, Server plug-in) or Provisioning Manager (user interface plug-ins),

Some connectors require an agent on the systems they manage in order to complete the communication cycle, in which case, they can be installed using the Provisioning Installer. Agents can be separated into the following categories:

Remote Agents

- Installed on the managed endpoint systems

Environment Agents

- Installed on systems, such as CA ACF2, CA Top Secret, and RACF

Certain components work on UNIX and Windows, including the following C++ Connector Server-based options:

- UNIX (ETC, NIS)
- Access Control (ACC)

Note: The UNIX ACC connector can manage only UNIX ACC endpoints. The Windows ACC connector is required to manage the Windows ACC endpoints but can also manage UNIX ACC endpoints.

- OpenVMS
- CA-ACF2
- RACF
- CA-Top Secret

The other C++ Connector Server-based connectors can be accessed from the Solaris Provisioning Server by relying on the Connector Server Framework (CSF). The CSF allows a Provisioning Server on Solaris to communicate with connectors running on Windows.

Note: The CSF must run on Windows to use these connectors.

Additional Components

Identity Manager includes some additional components, which support Identity Manager functionality. Some of these components are installed with Identity Manager and some must be installed separately.

WorkPoint Workflow

WorkPoint workflow engine and WorkPoint Designer are installed automatically when you install Identity Manager.

These components enable you to place an Identity Manager task under workflow control, and to modify existing workflow process definitions or create new definitions.

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

Provisioning Manager

The Identity Manager Provisioning Manager manages the Provisioning Server through a graphical interface. This is used for administrative tasks such as acquiring endpoints, installing endpoint types, and managing provisioning server options.

The Provisioning Manager is installed as part of the Identity Manager Administrative Tools.

Note: This application runs on Windows systems only.

For more information about the Provisioning Manager, see the *Provisioning Guide*.

iRecorder for Auditing

The iRecorder is an optional component that sends Identity Manager events to CA Audit. Once in CA Audit, the events can be monitored by CA Audit or CA Security Command Center (SCC).

You install the iRecorder from the Identity Manager installer.

Note: For more information about the iRecorder, see the *iRecorder Reference Guide for CA Identity Manager*. For more information about viewing Identity Manager events in SCC, see the *Configuration Guide*. These guides are available in the Identity Manager Bookshelf.

IAM Report Server

Identity Manager provides reports that you can use to monitor the status of an Identity Manager environment. To use the reports provided with Identity Manager, you install the IAM Report Server, which is included with Identity Manager.

IAM Report Server is powered by Business Objects Enterprise XI. If you have an existing Business Objects server, you can use that instead of the IAM Report Server to generate Identity Manager reports.

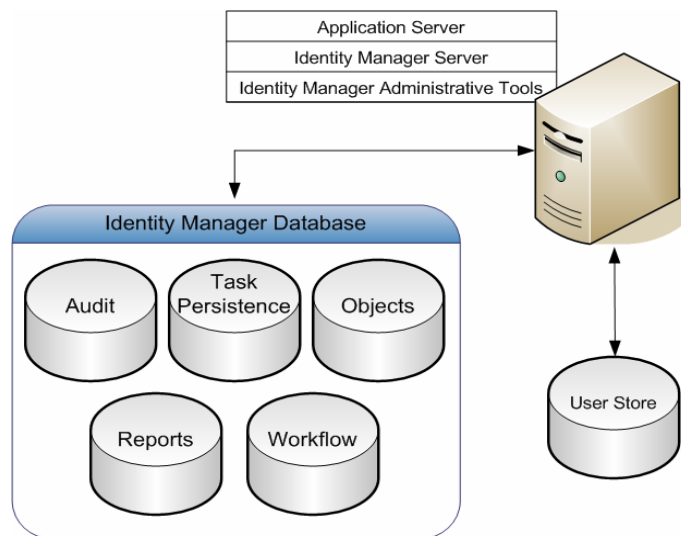
Note: For installation instructions, see the *Installation Guide*.

Sample Identity Manager Installations

Based on the functionality you want to implement, you can choose which components of Identity Manager you want to install in your environment. The following section illustrates some examples of Identity Manager implementations at a high level.

Basic Installation

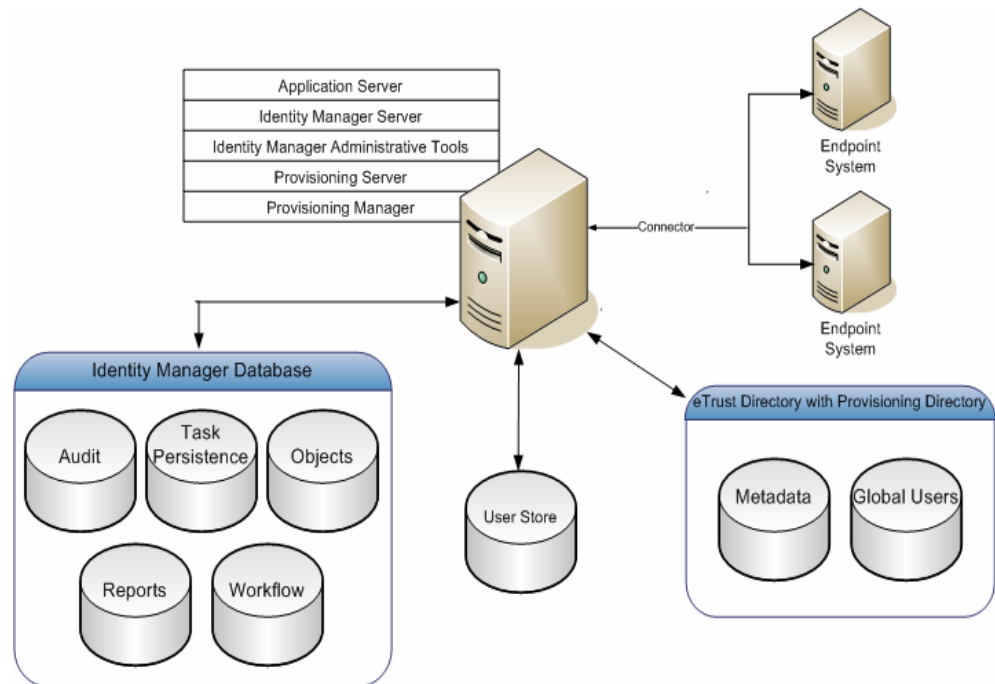
In all Identity Manager installations, the Identity Manager Server is installed on an application server. After you install the application server, you use the Identity Manager Installer to install all the software on the same system. The following figure is an example of a basic Identity Manager installation:



Installation with Provisioning Components

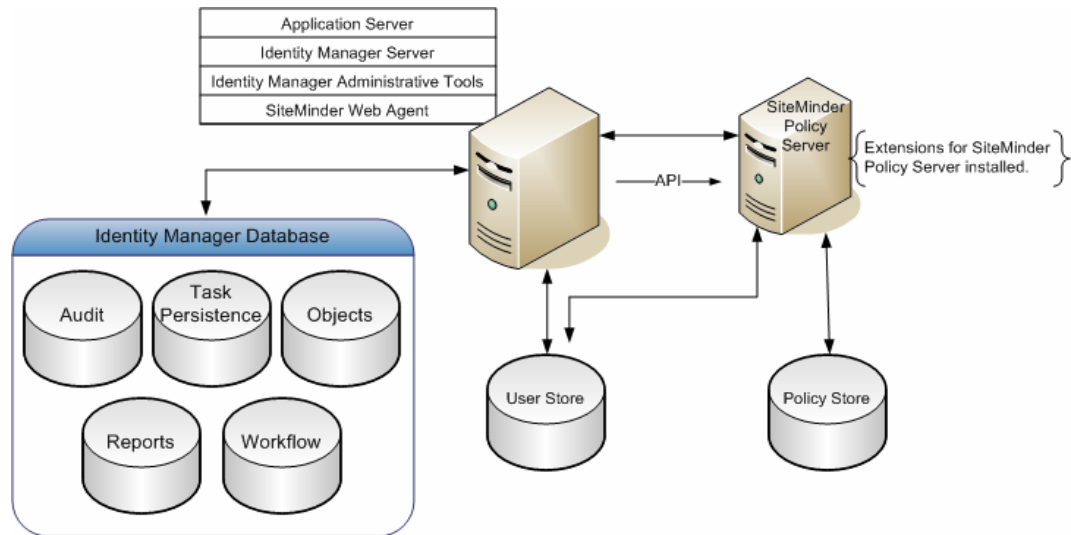
Identity Manager provisioning allows you to create an Environment that connects to a Provisioning Server for provisioning accounts to various endpoint systems. You can assign provisioning roles to users you create through Identity Manager. Provisioning roles are roles with account templates that define accounts that users can receive on endpoint systems. Accounts provide users with access to additional resources, such as an email account.

When you assign a provisioning role to a user, that user receives the accounts defined by the account templates in the role. The account templates also define how user attributes are mapped to accounts. The accounts exist in managed endpoints defined by the account templates. The following figure is an example of an Identity Manager installation with provisioning:



Installation with SiteMinder Policy Server

A SiteMinder Policy Server provides advanced authentication and protection for your Identity Manager environment. The following figure is an example of an Identity Manager installation with a SiteMinder Policy Server:



An Identity Manager implementation that includes SiteMinder includes all of the components in the basic installation or the installation with provisioning, plus these additional components:

SiteMinder Web Agent

Works with the SiteMinder Policy Server to protect the User Console. The Web Agent is installed on the system with the Identity Manager Server.

SiteMinder Policy Server

Provides advanced authentication and authorization for Identity Manager, as well as other functionality such as Password Services and Single-Sign On.

Extensions for SiteMinder Policy Server

Enables a SiteMinder Policy Server to support Identity Manager. Install the extensions on each SiteMinder Policy Server system in your Identity Manager implementation.

SiteMinder Policy Store

Stores information that SiteMinder needs to manage access to Web resources.

When Identity Manager integrates with SiteMinder, the policy store also includes information about Identity Manager directories and environments so that SiteMinder can provide advanced authentication.

Chapter 4: Planning Your Implementation

To plan an Identity Manager implementation, you decide how Identity Manager will manage users and what functionality you need to accomplish your business objectives. Some questions to consider are:

- How do I manage users?
- Do I need account provisioning?
- What are my custom business requirements and should I implement them using workflow?

Based on the decisions you make, you can determine the best way to implement Identity Manager for your environment.

This section contains the following topics:

[Decide What to Manage](#) (see page 39)

[Determine Audit Requirements](#) (see page 41)

[Decide User Store Requirements](#) (see page 43)

[Select Components to Install](#) (see page 45)

[Decide Hardware Requirements](#) (see page 46)

[Choose a Method to Import Users](#) (see page 48)

[Develop a Deployment Plan](#) (see page 53)

Decide What to Manage

Deciding what you want to manage will help you determine which components you want to install. Using Identity Manager, you can manage the following:

- User identities
- Access to accounts on endpoint systems

User Identities

User identities represent the people that a company needs to manage, such as employees, contractors, suppliers, and others.

To manage user identities, you need to install only the Identity Manager Server and the Administrative Tools.

Provisioning Accounts from Other Applications

The decision to implement provisioning depends on the type of information that you need to manage. If you are managing a central user directory and you do not want to manage user accounts in other systems, you do not need provisioning. If you want to manage user accounts over a variety of systems, then you should implement provisioning support.

Provisioning capabilities are provided through the Provisioning Server, which is integrated with Identity Manager. The Provisioning Server provides the following functionality for account provisioning:

- Endpoint Management
- Account Synchronization
- Account Templates
- Explore and Correlate Functionality

Note: Provisioning information is stored in a Provisioning Directory. If Identity Manager maintains users in another type of directory, your deployment will include an Identity Manager user store and a provisioning directory.

Endpoint Management

To provision accounts, you define and manage endpoints in the Provisioning Manager. An *endpoint* is a system for which users need access. Examples of endpoints include Oracle databases, UNIX NIS servers, Windows servers, and Microsoft Exchange servers.

Account templates (see page 41) create accounts and determine the user capabilities in managed endpoints.

Account Synchronization

You can synchronize user accounts across multiple managed endpoints. When account synchronization is enabled, a change made to a user profile in the Provisioning Server is propagated to all of the endpoints where that user has an account.

Note: You specify account synchronization settings on the Profile tab for an Identity Manager task. For more information about configuring account synchronization, see the *Administration Guide*.

Account Templates

Account templates define how a user is represented in a managed endpoint. For example, a template for an Exchange account could define the format of a user's email address, such as <first initial><last name>@mycompany.com.

Account templates also determine the privileges a user has within a managed system. For example, in addition to defining the format of an email address, a template for an Exchange account may also limit a user's mailbox size.

Explore and Correlate Functionality

The Explore and Correlate features simplify endpoint management by discovering and synchronizing changes in managed systems.

The Explore feature finds objects, including accounts, in endpoints, and stores references to them in the Provisioning Directory. You can use the Explore feature to detect any new objects to be managed. For example, if you provision accounts in an LDAP directory and new organizations are added in that directory, you can use the Explore feature to introduce those new organizations for use in account templates.

The Correlate feature associates an account in a managed endpoint with a global user in the Provisioning Directory. When a change is made to the account through the endpoint, the Correlate feature can synchronize those changes with the global user account.

Note: For more information about the Explore and Correlate functionality, see the *Provisioning Guide*.

Determine Audit Requirements

Identity Manager includes auditing capabilities that allow you to monitor activities in an Identity Manager environment.

This information is stored in an audit database. The amount and type of information that is stored in the audit database is configurable.

You view audit data in the User Console through a task called View Submitted Tasks. This task allows administrators to search for and view tasks that occur in the system. Administrators can view task information at a high level or view task and event details.

For advanced auditing, Identity Manager can integrate with CA Audit and CA Security Command Center (SCC). CA Audit provides security routing, filtering, and monitoring capabilities such as event aggregation, reduction, and correlation. CA Security Command Center provides custom reporting.

Identity Manager Auditing Considerations

Audit data provides a historical record of operations that occur in an Identity Manager environment. To audit data in Identity Manager, you need the following:

- An auditing database
- An audit settings file

Audit Database

When you use the Identity Manager Installer, Identity Manager configures a connection to a single database, called the Identity Manager Database, and creates a data source to connect to the database tables for auditing.

Note: The Identity Manager Database also includes data that is used by other Identity Manager functionality, including task persistence, workflow, and reporting. For scalability purposes, you can create a new, separate instance of a database for auditing.

Note: For more information about the auditing database, see the *Installation Guide*.

Audit Settings

You configure audit settings in an audit settings file. An audit settings file determines the amount and type of information that Identity Manager audits. You can configure an audit settings file to do the following:

- Enable auditing for an Identity Manager environment.
- Enable auditing for some or all of the Identity Manager events generated by admin tasks.
- Record event information at specific states, such as when an event completes or is cancelled.
- Log information about attributes involved in an event. For example, you can log attributes that change during a `ModifyUserEvent` event.
- Set the audit level for attribute logging.

Note: For more information about configuring auditing, see the *Configuration Guide*.

CA Audit Considerations

CA Audit is an audit management system that enables you to collect and store security related data for auditing, reporting, compliance verification and event monitoring.

To integrate with CA Audit, you install the iRecorder component when you install the Identity Manager server. The iRecorder retrieves events from Identity Manager. Based on policies in the CA Audit Policy Manager, the iRecorder ignores the event or routes it through to CA Audit.

CA Security Command Center (SCC) Considerations

SCC collects and normalizes security data from across the enterprise. It enables you to monitor, correlate, and analyze security events, across a wide range of security systems.

To manage Identity Manager data with SCC, you install a *product integration kit* (PIK), a set of tools and services used by SCC to provide integration with Identity Manager. The PIK component customizes the display of Identity Manager events and status.

Note: For more information on the PIK, see the *Product Information Kit for CA Identity Manager Guide*.

Decide User Store Requirements

An Identity Manager implementation must include a user store that contains the user identities that Identity Manager maintains. Typically, this is an existing user store that an enterprise uses to store information about its users, such as employees and customers.

If your implementation includes provisioning, Identity Manager also requires a provisioning directory that includes global users, which are associated with accounts on endpoints such as Microsoft Exchange, Active Directory, and Ingres.

How to Choose a User Store Solution

To manage an existing user store, use Identity Manager to manage it, and then create a separate Provisioning Directory. However, in implementations where you can choose a user store option, consider the key differences in the following table.

Capability	Separate User Store and Provisioning Directory	Single User Store and Provisioning Directory
User Object Support	Yes	Yes (custom schema)
Group Object Support	Yes	Yes (custom schema)
Group Membership Support	Yes	Yes (proprietary model)
Organization Object Support	Yes	No
Direct Provisioning Support	No	Yes
Additional user store required for provisioning	Yes	No
Recommended number of users	Millions of users	Tens of thousands of users

Managing Multiple User Stores

An enterprise may maintain multiple user stores. In each user store, the user identity allows access to different corporate resources. You can use one of the following methods to manage multiple user stores:

- Use Identity Manager to directly manage the Provisioning Directory and use the Provisioning Server to indirectly manage the users and accounts in the different user stores.

This approach allows you to:

- Centrally manage users who can be assigned various enterprise resources from one location
- Implement common security and business rules across enterprise resources. This may include the following:
 - Role-based access control
 - Delegated administration
 - Tasks and screens that are customized based on the type of corporate identities they manage
 - Identity policies for rule-based identity management
 - Customization and extensibility

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

- Create a separate Identity Manager environment to manage each user store

With this method, information is not shared between environments.

Select Components to Install

The following table lists the components to install to support the functionality that you want to implement.

Note: For instructions on installing these components, see the *Installation Guide*.

If you want to...	Install these components
Manage user identities in an existing corporate user store	■ Identity Manager Server
Provision accounts in endpoint systems	■ Provisioning Server ■ Provisioning Directory ■ Provisioning Manager ■ Connectors Note: For instructions on installing connectors, see the <i>Connector Guide</i> for the type of connectors that you want to install.

If you want to...	Install these components
Implement one or more of the following features: ■ Advanced authentication ■ Advanced password policies ■ Console skins for different sets of users ■ Configure locale preferences	■ SiteMinder Policy Server ■ Policy store ■ SiteMinder Web Agent ■ Identity Manager Extensions to the Policy Server Note: For instructions on installing the SiteMinder Policy Server and policy store, see the <i>CA SiteMinder Web Access Manager Policy Server Installation Guide</i>. For instructions on installing the Web Agent, see the <i>CA SiteMinder Web Access Manager Web Agent Installation Guide</i>.
Generate reports on Identity Manager activity	IAM Report Server

Decide Hardware Requirements

The hardware that you need for an Identity Manager installation depends on the functionality that you want to implement.

The following sections describe typical Identity Manager implementations and their required hardware.

Requirements for Basic Installations

A basic Identity Manager implementation includes the following components:

- Application server
- Identity Manager Server
- Identity Manager Administrative Tools
- Identity Manager Database

In a development environment, all of these components can be installed on a single machine; however, in a production environment, the Identity Manager Server should be installed on one machine with the application server, and the database should be installed on a separate machine.

The Administrative Tools can be installed on any machine that has access to the Identity Manager server.

Additional Requirements for Provisioning

In addition to the components required for a basic Identity Manager implementation, the following additional components are required when Identity Manager includes provisioning:

- Provisioning Server
 - Can be installed on the same machine as the Identity Manager server.
- Provisioning Directory Initialization
 - Important! The Provisioning Directory Initialization must be installed on eTrust Directory.**
- Provisioning Manager
 - Can be installed on any Windows machine that can access the Provisioning Server.

Note: In a development environment, these components can be installed on one machine that also includes the basic installation components.

Additional Requirements for SiteMinder Integration

When Identity Manager integrates with SiteMinder, the implementation must include the components in the basic Identity Manager installation, plus the following additional components:

- **Policy Server**

Provides policy management, authentication, authorization, and accounting services.

The Policy Server can be installed on the same machine as the Identity Manager Server, if the Policy Server is dedicated to Identity Manager. If the Policy Server is protecting other applications, we recommend installing it on a separate machine to ensure best performance.

- **Policy Store**

Contains all of the Policy Server data. You can configure a policy store in a supported LDAP or relational database. In high availability implementations, we recommend installing the policy store on a separate server.

- **Extensions to the Policy Server**

Enables a SiteMinder Policy Server to support Identity Manager. Install the extensions on each SiteMinder Policy Server system in your Identity Manager implementation.

- **SiteMinder Web Agent**

Works with the SiteMinder Policy Server to protect the User Console. Installed on the system with the Identity Manager Server.

Choose a Method to Import Users

If you need to import users into an existing user store, the method you select should be based on your business requirements.

The following sections describe options for importing users.

How to Import Users into a New User Store

After you decide how to store user data, you may need to import users from one store to another. Depending on your implementation, you can use different methods to import users.

Note: After importing users into a new user store, you can use [identity policies](#) (see page 50) to apply changes to imported users.

Import Users Through Identity Manager

Identity Manager provides the following methods for adding users to a user store that it manages directly.

Method	Features	Limitations
Bulk Loader	Allows you to use the Bulk Loader task in the User Console to upload feeder files that are used to manipulate large numbers of managed objects simultaneously. The advantage of the Bulk Loader method is that you can automate the process of manipulating a large number of managed objects using an information (feeder) file. The Bulk Loader task can also be mapped to a workflow process.	If you are using the Bulk Loader, you may see out-of-memory exceptions depending on the number of users you are importing. To address this issue, increase the JVM memory settings.
Remote Task Invocation via Task Execution Web Service (TEWS)	Allows execution of any Identity Manager task that is enabled for Web Services, including the Create User task. If the task is configured for User Synchronization, Identity Manager will execute any applicable identity policies.	Performance characteristics of web service model may not be well-suited for high-throughput requirements of bulk import operations
IM API	■ Provides User-based APIs that can be invoked directly for creating users via a Java client ■ Provides the highest throughput capabilities.	■ Bypasses audit and security mechanisms provided by the Task Server. ■ Does not support execution of Identity Policies.

Note: For more information about the Bulk Loader, see the *Administration Guide*. For more information about TEWS and the IM API, see the *Programming Guide for Java*.

Execute Identity Policies on Imported Users

An *identity policy* is a set of business changes that occur when a user meets a certain condition or rule. These changes can include assigning or revoking roles (including provisioning roles for users in the provisioning directory), assigning or revoking group membership, and updating attributes in a user profile.

You can use identity policies to apply changes to user accounts after they have been imported to a new user store.

This section describes methods for executing identity policies for imported users in one or two steps.

One-Step Approach

You can use the following import methods to execute identity policies on users that you import into a new user store in a single step:

- Bulk Loader in the User Console
- Create User Task Execution via TEWS
- Inbound Synchronization

Two-Step Approach

Using a two step approach, you first import users and then execute identity policies against the those users. You can use this method when Identity Manager manages users in the Provisioning Server. This method may provide more flexibility, depending on your import requirements.

1. Use one of the import tools for adding users into the Provisioning Directory.
2. Invoke the Identity Manager Synchronize User Task through TEWS on each of the imported users.

Import Users Through the Provisioning Server

The Provisioning Server includes bulk import options for adding and managing users in the Provisioning Directory. The following tables describe the methods for importing users into the Provisioning Directory.

Method	Features	Limitations
Batch utility (etautil)	A command line interface utility that allows you to manage objects in the Provisioning Directory	■ Currently supported for Windows systems only
Explore and Correlate	■ Discovers new objects that the Provisioning Server can manage in a known endpoint (including users) ■ Provides correlate capabilities for object instances that exist in the endpoint and the Provisioning Server. Additional information exists in Explore and Correlate Functionality.	■ By default, the Explore and Correlate functionality is available for the currently supported connectors. Can be extended with custom connectors ■ The Correlate option may affect scalability when working with large user populations. If you select this import option, be sure to evaluate the performance and scalability implications.

After importing users, you can use the method in the following table to make changes to user accounts in the Provisioning Server:

Method	Features	Limitations
Service Provisioning Markup Language (SPML)	You can generate SPML requests that are submitted directly to the Provisioning Server without having to convert user data to a CSV or XML input file.	■ Static solution that requires an application that can generate and submit SPML-formatted XML payloads as SOAP requests ■ Requires export of HRMS/ERP data into a file; additional data manipulation required

Synchronize Global Users with the Identity Manager User Store

After you import users into the Provisioning Server, you can use the following methods to add those users to the Identity Manager user store:

■ Inbound Synchronization

Inbound Synchronization keeps Identity Manager users up to date with changes that occur in the Provisioning Directory. Changes in the Provisioning Directory include those made using Provisioning Manager or systems with connectors to the Provisioning Server.

Note the following when using inbound synchronization to import users:

- In the Identity Manager Management Console, you can customize how the attributes from the inbound request are mapped to attributes in the Identity Manager task.

Note: For more information, see the *Provisioning Guide*.

- Consider which Provisioning Server changes require synchronization with the corporate user store. Synchronizing a large number of changes may impact performance and scalability.

■ **Provisioning Roles and Account Templates**

The Provisioning Server can manage accounts in the Identity Manager user store using provisioning roles and account templates. This requires that a managed endpoint, which points to the Identity Manager user store has been acquired and the appropriate account templates and roles exist. In this case, global users created through one of the options described in [Import Users Through the Provisioning Server](#) (see page 51) can be assigned a provisioning role that creates the user account in the Identity Manager user store.

Develop a Deployment Plan

When planning a large implementation, you should deploy Identity Manager functionality in stages. The following deployment order allows you to gain significant value from Identity Manager quickly, evaluate the changing needs of your implementation over time, and carefully construct your environment for best performance and scalability:

- Self-service and password management
- Identity policies
- Workflow approvals
- Delegated administration for user, group, and organization objects
- Delegated administration for role administration

After each deployment stage, be sure to evaluate performance and make adjustments before proceeding to the next stage. [Optimizing Identity Manager](#) (see page 63) provides information on performance, tuning, and scalability.

Deploy Self-Service and Password Management

Deploy self-service tasks and password management before deploying other Identity Manager features for the following reasons:

- Self-service tasks and password management are easy to deploy and provide significant value quickly.
- These features are independent of the delegated administration model and can be reconfigured as needed to address changing business needs.
- These features typically generate the highest volume of tasks that Identity Manager processes on a regular basis. Because of this, they provide a way to test the scalability of your implementation before you deploy additional features.

To deploy self-service tasks, you complete the following steps:

1. Configure the self-registration task.

This is a public task, which is enabled by default during installation. To configure this task, you add or remove fields from the default self-registration task, as needed.

2. Deploy the Self Manager role.

The member rule for this role should be configured to apply to all users, or should include a member rule that automatically assigns the role to new users. For example, you can create a member rule that assigns the Self Manager role to all full-time employees. When a user self-registers, Identity Manager can set the employee type to full-time (by using a logical attribute handler, or business task handler). The user meets the criteria in the member rule and receives the Self Manager role automatically.

Note: When you configure member rules for the Self Manager role, do not allow administrators to add or remove role members. Since the role is assigned automatically, there is no need for an administrator to explicitly assign the role.

To deploy password management capabilities, you complete the following steps:

1. Configure the public password management tasks, such as the Forgotten Password task.
2. Create password policies that determine how passwords are created and when they expire.
3. Deploy the Password Manager role, which enables role members to reset user passwords.

Note: For information on roles, tasks, and password management, see the *Administration Guide*.

Deploy Identity Policies

An identity policy is a set of business changes that occurs when a user meets a certain condition or rule. You can use identity policies to provide business-driven entitlements before a complete delegation model is deployed. For example, you can create an identity policy that assigns the Sales Manager provisioning role, which grants access to sales applications, to all users whose title is Sales Manager. When a sales representative is promoted to Sales Manager, he automatically receives access to all of the systems he needs to do his job without waiting for administrator involvement.

To deploy identity policies, you complete the following steps:

1. Configure identity policies that are triggered by changes to user profile attributes.
2. Configure the User Manager role to allow a small number of administrators to use user tasks, such as Create User and Modify User, to change the attributes that trigger the identity policies.

Be sure to configure the scope rules in the User Manager member policies to determine the set of users that role members can manage.

Note the following when deploying identity policies:

- Consider initially creating identity policies that grant entitlements that do *not* require workflow approvals. This allows you to deploy identity policies without having to define workflow processes, approval forms, and approver models.
- Before creating identity policies, you should be familiar with other methods of implementing business rules in Identity Manager, such as data validation rules, logical attributes, business logic task handlers, and workflow processes, to determine which method provides the best solution.

Note: For more information about these methods, see the *Administration Guide* and the *Programming Guide for Java*.

- Identity policies are an efficient way to assign entitlements in Identity Manager, however, they may [significantly impact performance](#) (see page 79).
- For the initial deployment of user tasks, consider removing or hiding relationship tabs, such as Roles tabs, that manage the same entitlements as identity policies. This prevents the risk of unauthorized entitlements and prevents the potential performance impact of improperly constructed roles.

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

Deploy Workflow Approvals

Workflow approvals can add an additional level of security and automation to your Identity Manager implementation.

Deploying workflow approvals requires the following tasks:

1. Decide which events or tasks require approvals.
2. Define the set of approvers, called participants, for each workflow process.

Note: All participants are determined dynamically by participant resolvers. To maintain good performance, limit the number of participants to thirty users.

3. Configure approval forms.
4. Define custom workflow processes, if needed.

Environment and Task Level Workflow Approvals

Identity Manager supports two types of approvals: environment-level approvals and task-level approvals. Environment-level approvals are defined for all instances of an event, regardless of the tasks they are associated with. Task-level approvals are defined for a specific event associated with a specific task. Task-level approvals take precedence over environment-level approvals.

Most approvals are defined at the environment level to ensure that the same workflow activities occur for an event, regardless of the task that it is associated with. However, in the following situations, consider implementing task-level workflow:

- You have specialized tasks that execute specific business changes that generate events, which do not require approvals.
- You have changes actions, triggered by identity policies, that generate events that do not require workflow approval.
- You need the flexibility to associate specific workflow processes with task-specific changes.

Environment-level approvals may require significant processing and system resources as the volume of transactions increases. This may eventually introduce performance and scalability issues. Using task-level approvals, where appropriate, may reduce or eliminate these issues.

Deploy Delegated Administration for Users, Groups and Organizations

Delegated administration is the management of users and their entitlements by having different Identity Manager users perform the functions of modifying, assigning, and using a role.

Note: Delegation models must be carefully constructed to ensure good performance and scalability in your Identity Manager implementation.

Delegation is enforced by scope rules, which are defined in member and admin policies for admin roles. A scope rule determines the objects on which a role member can use the role. For example, a scope rule may enable a User Manager to manage users in his department, but not in other departments.

Generally, scope rules should reflect the logical structure of the user store. For example, in a hierarchical LDAP user store, scope may be defined by organizations. In a relational database, scope can be defined using attributes such as department ID.

Note the following when deploying delegated administration for users, groups, and organizations:

- Limit access to relationship tabs, such as the Admin Roles and Provisioning Roles tabs, in user-related tasks. These relationship tabs are included in default user tasks, such as Create User and Modify User. Consider removing them from the default tasks and using them only in specialized tasks which are associated with a small number of admin roles.
- Identity Manager evaluates each scope rule dynamically; scope information is not cached. Consider creating scope rules that contain simple directory queries to ensure good performance.
- Evaluate the performance of scope rules by determining how long it takes Identity Manager to return the objects an administrator can manage.

Deploy Delegated Administration for Roles

Delegated administration of roles grants the most significant privileges in Identity Manager and can have the [greatest affect](#) (see page 64) on performance. For these reasons, you should consider deploying delegated administration for roles after you have deployed all other functionality.

When deploying delegated administration for roles, note the following:

- Limit the number of admin roles, admin role members, and admin role administrators to protect the environment and ensure good performance.
- Once you deploy delegated administration for roles, conduct performance and scalability tests. Optimize the environment as needed.

Chapter 5: Integrating with SiteMinder

This section contains the following topics:

[SiteMinder Integration](#) (see page 59)

[SiteMinder Authentication](#) (see page 60)

[Password Policies with SiteMinder](#) (see page 61)

SiteMinder Integration

When Identity Manager integrates with SiteMinder, SiteMinder can add the following functionality to an Identity Manager environment:

Advanced Authentication

Identity Manager includes native authentication for Identity Manager environments by default. Identity Manager administrators enter a valid username and password to log into an Identity Manager environment. Identity Manager authenticates the name and password against the user store that Identity Manager manages.

When Identity Manager integrates with SiteMinder, Identity Manager uses SiteMinder basic authentication to protect the environment. When you create an Identity Manager environment, a policy domain and an authentication scheme are created in SiteMinder to protect that environment.

When Identity Manager integrates with SiteMinder, you can also use SiteMinder authentication to protect the Management Console.

Directory Mapping

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 Identity Manager environment, the administrator must be authenticated using one directory and authorized to manage users in a second directory.

When Identity Manager integrates with SiteMinder, you can configure an Identity Manager environment to use different directories for authentication and authorization.

Note: For more information, see the *Configuration Guide*.

Advanced Password Policies

Identity Manager enables you to create basic password policies that manage user passwords by enforcing rules and restrictions governing password expiration, composition, and usage.

If you configure Identity Manager to integrate with SiteMinder, you can create advanced password policies that enable you to define the additional rules and restrictions.

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

Skins for Different Sets of Users

A skin changes the look of the User Console. When Identity Manager integrates with SiteMinder, you can enable different sets of users to see different skins. To accomplish this, 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 Identity Manager, to build the User Console.

Note: For more information, see the *Configuration Guide*.

Locale Preferences for a Localized Environment

When Identity Manager integrates with SiteMinder, you can define a user's locale preference using an `imlanguage` HTTP header. In the SiteMinder Policy Server, you set this header within a SiteMinder response and specify a user attribute as the header's value. This `imlanguage` header acts as the highest priority locale preference for a user.

Note: For more information, see the *Configuration Guide*.

Note: For information on the components in an Identity Manager implementation that includes SiteMinder, see [Installation with SiteMinder Policy Server](#) (see page 37).

SiteMinder Authentication

Identity Manager includes the following consoles, which should be protected:

User Console

Enables Identity Manager administrators to perform tasks in an Identity Manager environment.

Management Console

Enables Identity Manager administrators to create and configure an Identity Manager directory, Provisioning Directory, and an Identity Manager environment.

Identity Manager includes native authentication, which protects the User Console by default. The Management Console is not protected by default.

To protect the Management Console or to configure other types of authentication for the User Console, such as certificate or key authentication, Identity Manager must integrate with SiteMinder.

Note: For more information, see the *Configuration Guide*.

Password Policies with SiteMinder

Identity Manager enables you to create basic password policies that manage user passwords by enforcing rules and restrictions governing password expiration, composition, and usage.

If Identity Manager integrates with SiteMinder, you can create advanced password policies that enable you to define these additional rules and restrictions:

- Directory filters
- Password expiration:
 - Track failed or successful logins
 - Authenticate on login
 - Password expiration if not changed
 - Password inactivity
 - Incorrect password
 - Multiple regular expressions
- Password restrictions:
 - Minimum days before reuse
 - Minimum number of passwords before reuse
 - Percent different from last password
 - Ignore sequence when checking for difference
 - Profile attribute matching
 - Dictionary matching

Chapter 6: Optimizing Identity Manager

This section contains the following topics:

[Identity Manager Performance](#) (see page 63)

[Role Optimizations](#) (see page 64)

[Task Optimizations](#) (see page 72)

[Guidelines for Group Member\Administrator Optimizations](#) (see page 77)

[Identity Policy Optimizations](#) (see page 79)

[User Store Tuning](#) (see page 83)

[Tuning for Provisioning Components](#) (see page 84)

[Runtime Components Tuning](#) (see page 85)

Identity Manager Performance

Identity Manager performance depends on the individual performance of different features and components.

You can optimize the following functionality in an Identity Manager environment:

- Roles
- Tasks
- Group membership and management
- Identity policies

For additional performance gains, you can also tune the following components:

- User store
- Provisioning components
- Runtime components, including the databases, such as the task persistence database, and application server settings

To ensure best performance, configure the Identity Manager functionality using the guidelines in the following sections. Then, measure performance and tune components, as needed. Because the components work together, it may take several iterations before you find the best tuning settings for your environment.

Role Optimizations

Identity Manager includes three types of roles:

- Admin roles

Determine the privileges a user has in the User Console.

When a user logs into an Identity Manager environment, the user's account has one or more admin roles. Each admin role contains tasks, such as Create User, that a user can complete in that Identity Manager environment. The admin roles that a user has determine the presentation of the User Console, therefore users see only the tasks that are associated with their roles.

- Provisioning roles

Give users accounts in managed endpoints, such as an email system.

- Access roles

Offer an additional way to provide entitlements in Identity Manager.

Roles include policies that determine the following:

- Who can use the role (for admin and access roles only) and where they can use it
- Who can manage role members and administrators
- Who can modify the role definition

Evaluating roles and their associated privileges can have a significant impact on Identity Manager performance.

How Role Evaluation Affects Performance at Login

When an Identity Manager user attempts to log into the User Console, the following actions occur:

1. Identity Manager prompts the user to supply credentials, such as a user name and password.
2. The user's credentials are authenticated using one of the following methods:
 - Identity Manager native authentication
 - SiteMinder authentication, if the Identity Manager implementation includes SiteMinder

3. Identity Manager evaluates every member policy for every admin role in the environment to determine which admin roles apply to the user.

Note: This evaluation occurs only once for a given user. After the initial evaluation, Identity Manager caches the results. Identity Manager uses the cached information until a change occurs to the user or to the set of member policies, which causes Identity Manager to refresh the information in the cache.

4. The Identity Manager User Console displays the categories that the user can view based on his roles.

This process occurs for every user that logs into the User Console. If an Identity Manager environment contains a large number of roles, or inefficient member policies, role membership evaluation can significantly impact performance. In this case, the initial screen that users see when they log into the User Console may display slowly.

Note: Identity Manager does not need to evaluate member policies when a user accesses a public task to self-register or to request a forgotten password. In these cases, Identity Manager does not need a list of the user's roles because it does not display the complete User Console.

Role Objects and Performance

To support each role, Identity Manager creates a number of objects in the Identity Manager [object store](#) (see page 30), depending on the role configuration.

Identity Manager creates one base object for each role. In addition to the base object, Identity Manager creates two objects for each [member, admin, and owner policy rule](#) (see page 67), and two objects for each scope rule. The rule objects include:

- Rule definition object
Contains metadata about the rule, such as rule type
- Rule data object
Contains the expression to be evaluated

The following table illustrates the objects created for a single admin role.

Object Type	Base Object	Member Policy Objects	Admin Policy Objects	Owner Policy Objects
Admin Role	1	Member rules: 2 (1 rule definition object and 1 rule data object) Scope rules: 2 (1 rule definition object and 1 rule data object) Total: 4 objects	Admin rule: 2 (1 rule definition object and 1 rule data object) Scope rules: 2 (1 rule definition object and 1 rule data object) Total: 4 objects	Owner rule: 2 (1 rule definition object and 1 rule data object) Total: 2 objects

Note: This table assumes that there is only one member, admin, and owner policy.

For any admin role, Identity Manager creates at least 11 objects. If the member policy included 3 scope rules, the number would increase to 15 objects.

Large numbers of role objects may impact the performance of the object store searches and policy evaluations.

Object Store Performance

Identity Manager stores information that it needs to manage users and entitlements in an object store. Having a large number of role objects in the object store may cause the following issues:

- Searches for managed objects on Identity Manager tasks screens may take longer.

To reduce the impact on searches, [index attributes used in searches](#) (see page 83).

- Role management tasks may execute slowly.

Some examples of role management tasks that are affected by a large object store include the following:

- A Create Admin Role task is slow because Identity Manager must confirm that the role name is unique in the object store.
- The Delete Admin Role task must remove all objects created to support the role and the object cache must be updated.

- Identity Manager takes a long time to evaluate role policies.

Identity Manager caches information in the object store to improve performance.

Optimize Role Policy Evaluation

For each admin role, you can create three types of policies:

- Member policies

Define a member rule, which determines the users who receive the role, and scope rules, which determine the objects that role members can manage

- Admin policies

Define admin rules, scope rules, and administrator privileges for a role

- Owner policies

Define who can modify a role

To optimize performance when Identity Manager evaluates role policies, consider the following:

- Limit the number of admin roles in an Identity Manager environment.
- Follow the [guidelines for creating policy rules](#) (see page 68).
- Tune the user store. the user store.
- Tune the policy store, if Identity Manager includes SiteMinder.

Guidelines for Policy Rule Creation

One of the key factors in determining the overall performance of role policy evaluations is the amount of time it takes to evaluate any single policy rule. To improve policy rule evaluation time, note the following when you create a policy:

- When possible, limit the number of policy objects that Identity Manager creates and the number of user store searches that it performs by creating policy rules with complex expressions.

A single rule with a complex expression is more efficient than multiple rules with simple expressions.

- When possible, select the most efficient and scalable type of policy rule.
- Enable the in-memory evaluation option for policy rules.

The in-memory evaluation option significantly reduces policy evaluation time by retrieving information about a user to be evaluated from the user store and storing a representation of that user in memory. Identity Manager uses the in-memory representation to compare attribute values against policy rules.

Note: For more information about the in-memory evaluation option, see the *Configuration Guide*.

- Tune the user store.
- Tune the policy store, if your Identity Manager implementation includes SiteMinder.

Limit Policy Objects and User Store Searches

Each rule in a role policy requires a set of objects in the object store. When Identity Manager evaluates a rule, it loads these objects and performs any required user store searches.

The following example shows a member policy that includes three member rules. Each rule includes four scope rules.

Member Policies		
Member Rule	Scope Rules	
where (Department = "Engineering")	Access Role where (Name = "Development") Group where (Group Name = "Product Team") Provisioning Role where (Name = "Employee") User where (City = "Boston")	
where (Department = "Human Resources")	Access Role where (Name = "Development") Group where (Group Name = "Product Team") Provisioning Role where (Name = "Employee") User where (City = "Boston")	
where (Department = "Administration")	Access Role where (Name = "Development") Group where (Group Name = "Product Team") Provisioning Role where (Name = "Employee") User where (City = "Boston")	

In this example, Identity Manager creates the objects and performs the user store searches described in the following table when evaluating and applying the member policy.

Rule	Policy Objects	Potential User Store Searches
- Member rule: where (Department = "Administration") - User scope: City = "Boston" - Group scope: Group Name = "Product Team" - Provisioning role scope: Name = "Employee" - Access Task Scope:Name = "Development"	10 (one for each rule definition object, one for each rule data object)	5 (one for each rule definition object)
- Member rule: where (Department = "Engineering") - User scope: City = "Boston" - Group scope: Group Name = "Product Team" - Provisioning role scope: Name = "Employee" - Access Task Scope:Name = "Development"	10	5
- Member rule: where (Department = "Human Resources") - User scope: City = "Boston" - Group scope: Group Name = "Product Team" - Provisioning role scope: Name = "Employee" - Access Task Scope:Name = "Development"	10	5

In this example, Identity Manager creates 30 objects and executes 15 directory searches to determine membership and scope.

To limit the number of policy objects and user store searches that Identity Manager performs, combine rules into complex expressions. The following example specifies the same entitlements in the first example as one member rule.

Member Policies

Member Rule	Scope Rules
<code>where (Department = "Administration" or Department = "Engineering" or Department = "Human Resources")</code>	Access Role
	<code>where (Name = "Development")</code>
	Group
	<code>where (Group Name = "Product Team")</code>
	Provisioning Role
	<code>where (Name = "Employee")</code>
	User
	<code>where (City = "Boston")</code>

In this example, Identity Manager creates only ten policy objects and performs only five user store searches.

Rule	Policy Objects	Potential User Store Searches
- Member rule: where (Department = "Administration") OR where (Department = "Engineering") OR where (Department = "Human Resources") - User scope: City = "Boston" - Group scope: Group Name = "Product Team" - Provisioning role scope: Name = "Employee" - Access Task Scope:Name = "Development"	10	5

Select Scalable Policy Rule Types

In addition to the number of policy rules, the type of policy rule may also impact performance. Typically, policy rules are constructed based on how the user store is structured and how entitlements are determined. For example, you may create policy rules based on group membership, organization, or user attributes. However, when there are multiple ways to construct policy rules, consider the performance guidelines in the following table before deciding which type of rule to construct.

Note: The policy rule types in the following table are listed in order of performance, beginning with the most efficient rule type.

Policy Rule Type	Performance Notes
Organization	■ Best overall performance ■ Does not require a search in LDAP directories. Identity Manager uses the DN of the user being evaluated and the DN of the organization in the policy rule
Role	■ Identity Manager stores role object information and previous evaluations in the object store cache ■ In most cases, performance will be as good as organization policy rules
User attribute	■ Provides the best user store search performance, and is the least affected by large user populations ■ Allows you to enable in-memory evaluation for significant performance gains
Group Membership	■ Performance depends on group size and user store type

Task Optimizations

In Identity Manager, the tasks that a user sees in the User Console depend on that user's specific privileges. To display and execute tasks, Identity Manager must perform multiple security evaluations, which may have a significant impact on performance when applied over all of the users in an Identity Manager environment.

Identity Manager performs security evaluations when the following actions occur:

- A user logs into the User Console
In this case, Identity Manager must evaluate a user's roles to determine which tasks that user can access in the User Console.
- A user invokes a task
When a task is invoked, Identity Manager must determine which objects that user can manage with that task.

- A user accesses a relationship tab
A relationship tab is any tab where a user can view or manage a one-to-many relationship between the task's subject and a set of entitlements. An example of a relationship tab is the Admin Roles tab, which displays the roles that a user has.
- A user adds objects on a relationship tab
For example, Identity Manager performs additional security checks when a user adds additional roles to another user on the Admin Roles tab.

Task performance is affected by the following:

- Task scope, which determines where an administrator can use a task
- Relationship tabs, which display an object's relationship to other objects

Task Scope Evaluation and Performance

When an administrator uses an admin task that involves searching for a managed object, such as a user, group, organization, task, or role, Identity Manager evaluates and applies task scope rules. These rules can significantly impact the amount of time Identity Manager takes to display the list of objects to select for the task.

Note: Unlike member, admin, and owner policy evaluations, information about scope rule evaluations is not stored in a cache.

Task scope is determined by the following:

- The type of object that the task manages.
- Scope rules that apply to the admin role that includes the task. Scope rules are defined in member, owner, and admin policies.
- Any user-defined search criteria.

For example, consider a Modify User task, which is included in the User Manager role. The User Manager role has a member policy with a scope rule that allows User Managers to manage users in the Employees organization. An administrator opens the Modify User task and enters the search criteria: Last Name starts with A. In this case, the scope for the Modify User task is all users in the Employees organization whose last name starts with A.

How Identity Manager Renders Relationship Tabs

A relationship tab allows users to view and manage the relationship that a task's subject has with a set of entitlements. For example, the Provisioning Roles tab shows the provisioning roles that a user has.

To determine the objects that appear on a relationship tab, Identity Manager performs numerous security evaluations, which can significantly impact performance.

The following example shows the steps that Identity Manager takes to render the Provisioning Roles tab:

1. An administrator clicks the Provisioning Roles tab in the Modify User task.
2. Identity Manager retrieves the provisioning roles where the selected user is a member.
3. If the tab is configured to allow management of role administrators, Identity Manager makes a second call to retrieve the list of provisioning roles where the selected user is an administrator.
4. Identity Manager evaluates each provisioning role that the user has to see if the administrator who initiated the task can manage membership for that role.

If the administrator can manage role members, Identity Manager displays an active check box in the Membership column for that role in the list of roles on the tab.

5. Identity Manager evaluates each provisioning role that the user has to see if the administrator who initiated the task can manage administrative rights for that role.

If the administrator can manage administrative rights, Identity Manager displays an active check box in the Administrator column for that role in the list of roles on the tab.

Identity Manager must complete steps 2-5 to display the provisioning roles the user currently has. If the administrator needs to assign a new provisioning role, the following additional steps are required.

6. The administrator clicks the Add button to locate new provisioning roles to assign.
7. Identity Manager displays a search screen that the administrator can use to search for the role to add.
8. The administrator enters a search filter to find the role to add.

9. Identity Manager returns the list of provisioning roles that meet following criteria:
 - The roles match the search filter entered by the administrator.
 - The administrator can manage membership for the roles.
 - The user is in the administrative scope of the administrator for the roles.
 - The user does not already have the provisioning roles.
10. Identity Manager repeats step 9 to determine the roles where the administrator can manage administrative privileges.

Relationship Tabs and Performance

Because of the number of security evaluations Identity Manager performs, rendering a relationship tab can significantly impact performance. The factors that determine performance vary depending on the type of tab.

For role relationship tabs, the following factors can impact performance:

- Number of roles where the task's subject is a member
- Number of roles where the task's subject is an administrator
- Number of total objects in the system that Identity Manager requires to calculate the subject's roles
- Number of member/admin policies per role
- Complexity of the member/admin policy scope rules
- The ability to maintain cached authorizations for the task invokers to limit the effect of the security enforcements

To determine group membership and administrative privileges on group relationship tabs, Identity Manager must search all of the groups in the user store. Performance of these searches depends on the following factors:

- Number of group objects in the user store
- Number of members in any group
- Performance of the database or directory where the user store exists

Guidelines for Optimizing Tasks

The default tasks, which Identity Manager deploys when you create an Identity Manager environment, are configured to support a wide range of administration use cases. Most Identity Manager implementations do not require all of the functionality provided in the default tasks. After creating an Identity Manager environment, modify these tasks to suit specific administration needs.

The following steps provide guidelines for modifying tasks:

■ **Create specialized user management tasks**

The default Create User, Modify User, and View User tasks provide full administrative capabilities. In most implementations, only a small number of administrators need all of the available capabilities.

Create new tasks that include only the required capabilities. For example, if most user management tasks involve only profile and group management, create a new Modify User task that includes only the Profile and Group tabs. Remove the Admin Roles, Access Roles, and Provisioning Roles tabs, which are available in the default Modify User task.

Unused tabs can cause significant overhead if they are left in frequently used tasks. This is especially true when using a Task Execution Web Service (TEWS) client, where these tabs may be inadvertently activated through the tab java class, which is provided with Identity Manager.

The specialized tasks that you create should match the [delegated administration model](#) (see page 57) that you defined for your environment.

■ **Disable Manage Administrators in relationship tabs**

By default, all relationship tabs provide the ability to manage administrative rights for the object that the tab manages, such as roles and groups. Most implementations do not need to provide this functionality to administrators.

To eliminate the additional overhead that occurs when Identity Manager evaluates administrative rights, clear the Manage Administrators option on the following tabs, if this functionality is not required:

- Admin Roles
- Provisioning Roles
- Access Roles
- Groups

To enable users to manage administrative rights on specific tabs, create copies of the default tabs, enable the Manage Administrators option, and disable the Manage Members option. Add the new tabs to specialized tasks, which are only used by the administrators who need them.

- **Enable scoped searches in role relationship tabs**

You can configure each role tab to include searches that allow administrators to specify criteria for new roles to assign to a user. Role searches limit the number of member and admin policy rules that Identity Manager must evaluate to determine which roles an administrator can assign to a user.

- **Set task synchronization options**

For each Identity Manager task, you can specify a user synchronization option, which synchronizes users with identity policies, and a provisioning account synchronization option, which synchronizes users with provisioned accounts. The options enable you to synchronize users when a task completes, or when an event completes.

To eliminate evaluation and processing time, set the synchronization to occur when a task completes, instead of when events complete.

Guidelines for Group Member\Administrator Optimizations

To improve performance of searches for group members and administrators, consider the following:

- Define well-known attributes in the directory configuration file (directory.xml), which describes the user store structure and contents to Identity Manager.

A well-known attribute is an attribute that has a special meaning in Identity Manager.

To improve group member\administrator searches, define the following well-known attributes for the user object:

%MEMBER_OF%

Identifies an attribute on the user object that stores a list of groups where the user is a member.

When defined, this attribute can prevent Identity Manager from searching all of the members in all of the groups in the user store. Group searches can significantly affect performance in very large groups.

%ADMINISTRATOR_OF%

Identifies an attribute on the user object that stores a list of groups where the user is an administrator.

Like the %MEMBER_OF% attribute, this well-known attribute can eliminate lengthy group searches.

- Specify the Group Type in the directory configuration file

Identity Manager supports three types of groups: standard groups, nested groups, and dynamic groups.

When you define the group object in the directory configuration file, you can specify the type of groups that the user store supports. If your implementation does not support nested or dynamic groups, set the Group Type attribute as follows:

GroupType = NONE

The setting NONE specifies support for standard groups.

The default Group Type setting is ALL, which may impact performance.

Note: For more information about well-known attributes and group types in the directory configuration file, see the *Configuration Guide*.

- Set the Provisioning Directory cache indices to improve GlobalGroup performance

For Identity Manager implementations that include a combined user store and Provisioning Directory, GlobalGroup membership can be optimized for policy rule evaluation for roles and identity policies.

To enable this optimization, you index the following attributes, which the Provisioning Server uses to resolve group membership, in the Provisioning Directory cache:

eTID

The unique object ID attribute. For group membership lookups, the value is a specific user or group involved in the lookup.

eTPID

The parent ID of the object used when searching for membership relationships.

eTPID

The child ID of the object used when searching for membership relationships.

Additionally, add the following hash entries:

eTSuperiorClass

The type of the parent object in a membership lookup

eTSubordinateClass

The type of the child object in a membership lookup

Note: For more information about the Provisioning Directory cache, see the *High Availability Guide*.

Identity Policy Optimizations

An *identity policy* is a set of business changes that occurs when a user meets a certain condition or rule. These changes can include assigning or revoking roles, assigning or revoking group membership, and updating attributes in a user profile.

Identity Manager evaluates identity policies when user synchronization occurs.

Identity policy performance is affected by the following:

- How the identity policies are configured
- How often user synchronization occurs

How Users and Identity Policies Are Synchronized

When using identity policies, it is important to understand how Identity Manager evaluates and applies the policies to users. Without a thorough understanding of the user synchronization process, you may configure identity policy sets that yield unexpected results.

The following procedure describes how Identity Manager evaluates and applies identity policies:

1. The user synchronization process begins:
 - **Automatically**—You can configure Identity Manager tasks to automatically trigger user synchronization
 - **Manually**—Use the Synchronize User task in the Identity Manager User Console to synchronize a user.
2. Identity Manager determines the set of identity policies that apply to a user.
3. Identity Manager compares the set of identity policies that apply to a user with the list of policies that have already been applied to that user.

Note: The list of policies that have been applied to a user is stored in the %IDENTITY_POLICY% well-known attribute in the user profile. For information on configuring this attribute, see the *Configuration Guide*.

 - If an identity policy is on the list of applicable policies, *and* the policy has *not* been applied to the user previously, then Identity Manager adds the policy to an allocation list.
 - If an identity policy is on the list of applicable policies, the policy has been previously applied to the user, and the Apply Once setting for the policy is disabled, Identity Manager adds the policy to a reallocation list.

- An identity policy is not on the list of applicable policies, and the policy has been applied to the user, the user no longer matches the policy condition. Identity Manager adds these policies to a deallocation list.
4. After Identity Manager evaluates all of the policies for a user, it applies policies in the following order:
 - a. Identity policies from the deallocation list
 - b. Identity policies from the allocation list
 - c. Identity policies from the reallocation list
 5. After the identity policies have been applied, Identity Manager reevaluates the policies to see if any additional changes are needed based on changes that occurred in the first synchronization process (steps 2-4).

This is to ensure that changes made by applying identity policies did not trigger other identity policies.
 6. Identity Manager continues to reevaluate and apply identity policies until the user is synchronized with all applicable policies, or until Identity Manager reaches the maximum recursion level, which is defined in the Identity Manager Management Console.

For example, an identity policy may change a user's department when the user is assigned a role. The new department triggers another identity policy. However, if the recursion level is set to 1, the subsequent change is not made until the user is synchronized again.

For more information about setting the recursion level, see the *Management Console Online Help*.

Design Efficient Identity Policies

Use the following guidelines when you create identity policies:

- **Limit the number of policy objects**

Identity Manager creates objects in the object store that support identity policies. To reduce the number of objects in the object store, create identity policies with complex expressions.

A similar approach is recommended for [role policies](#) (see page 69).

- **Limit identity policy set iterations**

You can configure the recursion level for an identity policy, which determines the number of times that Identity Manager evaluates and applies identity policies when a user is synchronized. For example, an identity policy may change a user's department when the user is assigned a role. The new department triggers another identity policy. However, if the recursion level is set to 1, the subsequent change is not made until the user is synchronized again.

Setting the recursion level limits the number of times that Identity Manager must evaluate identity policies.

- **Limit dependencies between identity policy rules**

You can create an identity policy where the change action (Action on Apply Policy or Action on Remove Policy) of one policy is used in the identity policy condition of another policy as shown in the following table.

Identity Policy Condition	Action on Apply Policy	Action on Remove Policy
where (Job Code = "100")	Make member of (provisioning role "Account Manager")	Remove member of (provisioning role "Account Manager")
Who are members of (provisioning role "Account Manager")	Make member of (group "Account Managers")	Remove member of (group "Account Managers")

When Identity Manager evaluates this type of policy, it must evaluate and apply changes at least twice to ensure that both conditions are met. The recursion level, which is set for an entire Identity Manager environment, must be greater than 1, which then causes additional evaluations for each identity policy set.

Limit the Tasks that Trigger User Synchronization

Identity policies are evaluated and applied during the user synchronization process. You can configure automatic synchronization by specifying one of the following user synchronization options for a task:

On Task Completion

Identity Manager starts the user synchronization process after all of the events in a task have completed.

On Every Event

Identity Manager starts the user synchronization process when each event in a task completes.

For best performance, limit the number of tasks that trigger automatic user synchronization.

Consider the following when configuring user synchronization:

- **Disable user synchronization for password tasks**

In most cases, passwords are not used in identity policy conditions.

- **Disable user synchronization for the Synchronize User task**

Since the Synchronize User task triggers identity policy evaluations, Identity Manager performs the evaluations again if the user synchronization option is enabled for this task.

- **Create specialized tasks**

When possible, create tasks that execute modifications that trigger identity policy conditions and enable user synchronizations for those tasks only.

Optimize Identity Policy Rule Evaluation

To reduce the evaluation time for identity policy conditions that include user-attributes, you can enable an in-memory evaluation option. When the in-memory evaluation option is enabled, Identity Manager retrieves information about a user to be evaluated from the user store and stores a representation of that user in memory. Identity Manager uses the in-memory representation to compare attribute values against policy conditions. This limits the number of calls Identity Manager makes directly to the user store.

Note: For more information about the in-memory evaluation option, see the *Configuration Guide*.

User Store Tuning

User store tuning involves a number of steps, including the following:

- Optimizing the structure of the user store
- Tuning underlying stores
- Implementing load balancing and replication

These steps depend on the type of user store that you are using. For tuning information in these areas, see the documentation for the database or directory that contains the user store.

In addition to the general tuning considerations, the following tuning considerations are specific to Identity Manager:

- **Measure user store search performance**

For optimum performance, Identity Manager policy evaluation searches should complete within 10-20 milliseconds.

To ensure that Identity Manager can consistently complete these searches in the recommended time, consider testing search performance under multiple load conditions.

You can also use this measurement to determine when a user store reaches its physical limits and additional servers are required for load balancing.

- **Index attributes**

Index each attribute that is used in a role policy or identity policy. Indexing attributes can provide significant performance improvements.

Note: For information on indexing attributes, see the documentation for the LDAP directory or relational database that contains the user store.

- **Cache LDAP Binds**

In Identity Manager, all directory LDAP binds are executed by the proxy user defined on the Identity Manager Directory object. For each connection, the same LDAP bind occurs for this same user repeatedly.

If you are using an LDAP directory as a user store, configure the directory to cache LDAP binds (or sessions), if the directory supports it.

- **Enable user store caches**

When Identity Manager evaluates the policy decisions for a user, that information is stored in an authorization cache. When the cached information expires, Identity Manager evaluates all policies for that user again.

To improve performance of user store searches in subsequent policy rule evaluations, enable the user store to cache searched data, if your user store supports it.

eTrust Directory includes a cache, called dxCache, which is an in-memory database implementation that can search across cached data.

Note: For more information about eTrust Directory, see the *eTrust Directory Administrator Guide*.

Tuning for Provisioning Components

When an Identity Manager implementation includes provisioning, use the following optimizations to ensure the best performance:

- Optimize the connection between the Identity Manager Server and the Provisioning Server

Identity Manager communicates with the Provisioning Server using the Java IAM (JIAM) API. To improve communication performance, you configure the following:

- JIAM session pool for multiple connections to the Provisioning Server

Note: CA recommends setting the initial sessions value to 8, and the maximum sessions to 128.

- JIAM cache for objects retrieved from the Provisioning Server

Note: For information on JIAM configuration settings, see the *Provisioning Guide*.

- [Set account synchronization to occur at the end of a task](#) (see page 76), instead of the end of each event

- Tune the Provisioning Server

Note: See the *Provisioning Guide* and *High Availability Guide* for more information.

Runtime Components Tuning

Business changes in Identity Manager are accomplished through tasks. A task includes one or more events, which represent activities that Identity Manager performs to complete the task. For example, a Create User task may include the CreateUserEvent and the AddToGroupEvent.

Identity Manager includes the following components, which process tasks and events at runtime:

- Identity Manager databases, which support Identity Manager functionality
- JMS messages, which are responsible for processing events

Tuning Identity Manager Databases

When executing tasks, Identity Manager uses the following databases:

- Task persistence

Maintains information about Identity Manager tasks and events over time. This allows Identity Manager to restore the last known state of events and tasks in the case of system failure.

Note: This database has the most significant impact on Identity Manager performance because the task and its events are saved and retrieved from the database during state transitions.

- Audit

Provides a historical record of operations that occur in an Identity Manager environment.

- Workflow

Stores workflow process definitions, jobs, scripts, and other data required by the workflow engine.

- Reporting

Stores snapshot data, which reflects the current state of objects in Identity Manager at the time the snapshot is taken.

Identity Manager communicates with each database through a JDBC connection pool. You create and configure a JDBC connection pool in the application server that hosts Identity Manager. When you configure the JDBC connection pool, note the following:

- Consider the number of concurrent tasks that will execute at any one time.
- Consider the other runtime components when you configure the JDBC connection pool size. Each runtime component works in conjunction with the other runtime components.

Note: CA recommends setting the initial connection pool value to 128.

- For the task persistence database, the number of database connections in the pool must allow each executing task to retrieve and update task and event data throughout the lifetime of the task.
- The task persistence database uses prepared statements. Be sure to configure the prepared statement cache for the database that you are using to store task persistence data.

Note: See the documentation for the database that you are using for task persistence for information on configuring the prepared statement cache.

JMS Settings

An Identity Manager task includes events, actions that Identity Manager performs to complete a task.

During an event's life cycle, it transitions through the following states:

- BEGIN
- APPROVED
- EXECUTING
- COMPLETED

Identity Manager uses JMS messages to control these state transitions.

How JMS Messages Drive Event Transactions

Identity Manager uses JMS messages to drive an event's state transitions. The following procedure describes the steps involved:

1. A user submits a task.
2. The task generates one or more events.

3. When an event is ready for processing, Identity Manager sets the event's state to BEGIN and the event is persisted in the task persistence database.
4. Identity Manager creates a JMS message containing the event ID and posts that message to the Event Message Queue.
5. Upon receiving the message, JMS then invokes an instance of the Event Message Driven Bean, which is an implementation of the Event Controller.
6. The Event Controller uses the event ID in the message to retrieve the event from the task persistence database, and executes the actions for the event's current state.
7. Upon completion of that state, the event is set to the next state, persisted in the task persistence database, and a new JMS message is posted for processing the next state.

This cycle continues until the event has completed its state machine.

JMS Messages and Performance

For any event, there are four states that require JMS messages for state transition:

- BEGIN
- APPROVED
- EXECUTING
- COMPLETED

To process a single event, the following actions take place:

- Four posts to the Events Message Queue
- Four invocations of the Message Driven Bean
- Eight connections to the task persistence database (one read action and one write action per state)

These actions may impact the amount of time it takes Identity Manager to process a task.

To ensure best performance during state transitions, tune the JMS resources in the application server that hosts Identity Manager so that adequate JMS resources are available.

Tuning JMS Settings

The following application server JMS tuning parameters define Queue connections and Message Driven Bean instance pools.

■ WebSphere JMS Tuning

WebSphere provides two parameters that you can configure to improve performance of event state transactions:

- The number of connections in the Queue Connection Factory that are available for posting JMS messages
- The number of instances of Message Driven Bean objects for processing the posted messages

Use the WebSphere Administration Console to set the following properties:

- Set the maximum session pool connections for the EventMsgQueue Queue Connection Factory to 128
- Set the maximum sessions for the SubscriberMessageEJBPort Listener Port in the Message Listener Service to 128

■ WebLogic Tuning

In WebLogic application servers, Queue Connection Factories obtain connection-handling threads from the server's JMS Thread Pool or the default execute pool, depending on the JMS Thread Pool size. If the JMS Thread Pool size is 0, then WebLogic uses the threads in the execute pool.

CA recommends setting the number of JMS Thread Pool threads equal to the maximum Bean Pool size for the Identity Manager Event Message Driven Bean, which is set to 128 by default.

You use the WebLogic Server Console to set the JMS Thread Pool size in the JMS Services properties for the domain and server where Identity Manager is installed.

The Identity Manager Event Message Driven Bean pool size is set by modifying the max-beans-in-free-pool setting in the descriptor file in the following location:

WebLogic_home\domain\applications\IdentityMinder.ear\identityminder_ejb_jar\META-INF\weblogic-ejb-jar.xml

```
<weblogic-enterprise-bean>
  <ejb-name>SubscriberMessageEJB</ejb-name>
  <message-driven-descriptor>
    <pool>
      <max-beans-in-free-pool>128</max-beans-in-free-pool>
      <initial-beans-in-free-pool>16</initial-beans-in-free-pool>
    </pool>
    <destination-jndi-name>com.netegrity.ims.msg.queue</destination-jndi-name>
  </message-driven-descriptor>
</weblogic-enterprise-bean>
```

■ JBoss Tuning

In JBoss application servers, Queue Connection Factories obtain connection-handling threads from the server's Standard JMS Pool session factory. By default, the number of maximum threads is set to 15.

CA recommends setting this value to match the maximum size value of the Standard Message Bean Container.

The JMS Session Pool section factory is set in the MaximumSize element of the JMSContainerInvoker in the following file:

jboss_home\server\default\conf\standardjboss.xml

```
<invoker-proxy-binding>
  <name>message-driven-bean</name>
  ....
  <proxy-factory-config>
    <JMSPProviderAdapterJNDI>DefaultJMSPProvider</JMSPProviderAdapterJNDI>
    <ServerSessionPoolFactoryJNDI>StdJMSPool</ServerSessionPoolFactoryJNDI>
    <MaximumSize>128</MaximumSize>
    <MaxMessages>1</MaxMessages>
    ....
  </proxy-factory-config>
</invoker-proxy-binding>
```

The Identity Manager Event Message Driven Bean pool size is set by modifying the maximum size value in the following descriptor file:

jboss_home\server\default\conf\standardjboss.xml

```
<container-configuration>
  <container-name>Standard Message Driven Bean</container-name>
  <call-logging>false</call-logging>
  <invoker-proxy-binding-name>message-driven-bean</invoker-proxy-binding-name>
  .....
  <container-pool-conf>
    <MaximumSize>128</MaximumSize>
  </container-pool-conf>
</container-configuration>
```


Chapter 7: Transitioning From eTrust Admin to Identity Manager

This section contains the following topics:

[Develop a Transition Plan](#) (see page 91)
[Business Changes in Identity Manager](#) (see page 92)
[Provisioning Terminology Changes](#) (see page 93)
[eTrust Admin Management Interfaces](#) (see page 93)
[Batch Processing in Identity Manager Implementations](#) (see page 95)
[Custom Endpoint Connectors](#) (see page 96)
[Next Steps](#) (see page 96)

Develop a Transition Plan

When planning to transition from eTrust Admin to Identity Manager, consider the following:

1. Become familiar with the features and functionality available in Identity Manager.
2. Identify the components and functionality that you use in your current eTrust Admin implementation. Common components and functionality include:
 - Management Interfaces
 - Feed Options and Web Services Interfaces
 - Advanced Workflow
 - Custom Endpoint Connectors
3. Compare the functionality in your existing eTrust Admin implementation with the functionality in Identity Manager.
4. Decide where to implement the functions that your business requires.

For assistance in making transition decisions, see the following sections:

- [eTrust Admin Management Interfaces](#) (see page 93)
- [Custom Endpoint Connectors](#) (see page 96)

Business Changes in Identity Manager

The Identity Manager Server, which users access through the Identity Manager User Console or through public tasks, typically manages a company's business tasks. These tasks may include user management, delegation, self service, approvals, business logic and password management.

The Identity Manager server offers the following benefits:

- A single location for user management.
- Fine-grained privilege management—Roles, scope rules, and delegated administration allow administrators to share their workload while maintaining control over which objects a delegated administrator can manage.
Note: For more information, see the *Administration Guide*.
- Customized user interfaces—You can customize the appearance and content of the Identity Manager task screens to suit your business needs. For example, you can create two sets of tasks—one for employees and one for customers. Each set of tasks can include different branding and can collect different information.
Note: For more information, see the *Configuration Guide*.
- Support for Multiple Platforms—Identity Manager runs on WebLogic, WebSphere, and JBoss application servers running on Windows or Solaris systems.
- Support for multiple user store types—Identity Manager can manage users in multiple types of LDAP user directories and relational databases.

For a complete list of supported user store types, see the Identity Manager support matrix.

To locate the support matrix:

1. Log into support.ca.com.
2. Click Support By Product or Solution.
3. Select CA Identity Manager in the Products section under Select a Product or Solution page.

The CA Identity Manager page opens.

4. Scroll to Recommend Readings.
5. Click CA Identity Manager Informational Documentation Index.

A page displays platform support matrices for supported versions of Identity Manager.

Provisioning Terminology Changes

Existing eTrust Admin customers may notice certain terms have changed now that eTrust Admin is part of CA Identity Manager. The following table shows these changes.

eTrust Admin Term	New Term in Identity Manager
eTrust Admin Server	Provisioning Server
eTrust Admin Manager	Provisioning Manager
Directory	Endpoint, Endpoints
Namespace	Endpoint Type
Policy or Provisioning Policy	Account Template
Roles	Provisioning Roles
Distributed SuperAgent Framework	Connector Server Framework
SuperAgent	C++ Connector Server
Option	Connector
Administrative Directory or Administrative Repository	Provisioning Directory
Identity Manager Corporate Directory	Identity Manager User Store
Corporate User	Inbound Administrator

eTrust Admin Management Interfaces

An eTrust Admin implementation may include user interfaces for user self-service, delegated administration, and endpoint and account management.

In an Identity Manager implementation, most of these functions are performed in the Identity Manager Server. For migration considerations, see the following sections:

- [SAWI/DAWI Considerations](#) (see page 94)
- [Password Management Considerations](#) (see page 94)
- [IA Manager Considerations](#) (see page 94)
- [Provisioning Manager Considerations](#) (see page 94)

SAWI/DAWI Considerations

In some eTrust Admin implementations, self service and delegated administration are provided through the Self Administration Web Interface (SAWI) and the Delegated Administration Web Interface (DAWI) respectively.

Identity Manager self-service tasks and delegated administration replace the functionality in the SAWI/DAWI components. You should not use the SAWI/DAWI components in an Identity Manager implementation.

Password Management Considerations

Like delegated administration and user self-service, passwords are managed in Identity Manager instead of eTrust Admin. For information on Identity Manager password management, see Password Management.

IA Manager Considerations

eTrust Admin included the IA Manager, which replaced SAWI/DAWI in some eTrust Admin implementations. The IA Manager allowed users to do the following:

- Self-register
- Manage their profiles, including changing passwords
- Reset forgotten passwords
- Delegate user administration tasks
- Access Advanced Workflow requests and approval worklists

Identity Manager enables users to perform self-service tasks, including password management, and delegated administration through the User Console.

Advanced Workflow is not supported in Identity Manager r12. However, Identity Manager enables you to perform all of the functions that you can perform using Advanced Workflow.

Provisioning Manager Considerations

In an Identity Manager implementation, most user management tasks are performed in the User Console. For example, you use the User Console to manage global users and groups, and to create and assign provisioning roles. Although you can still perform these functions in the Provisioning Manager (previously called the Admin Manager), the User Console provides a single location for performing business changes.

After migrating user management tasks to the Identity Manager User Console, some provisioning tasks will still require the Provisioning Manager:

- Managing endpoint types and endpoints.
- Creating and managing account templates (previously called provisioning policies).

In the Identity Manager User Console, the account templates that you create in the Provisioning Manager are added to provisioning roles, which are then assigned to users.

- Managing accounts using account templates
- Provisioning Server administration

Batch Processing in Identity Manager Implementations

If you used the Universal Feed Option or the PeopleSoft Feed Option for batch loading users in eTrust Admin, consider the following methods, which allow you to batch load and modify user information without feed options:

- Bulk Loader—You can use the Bulk Loader tab in the User Console to upload feeder files that are used to manipulate large numbers of managed objects simultaneously. The advantage of the Bulk Loader method is that you can automate the process of manipulating a large number of managed objects using an information (feeder) file. The Bulk Loader task can also be mapped to a workflow process.

Note: CSV is the default file format for the feeder, but you can create a custom feed for other file formats.

- Service Provisioning Markup Language (SPML)—You can generate SPML requests that are submitted directly to the Provisioning Server without having to convert user data to a CSV or XML input file.
- TEWS—Identity Manager provides Task Execution Web Services (TEWS) which expose Identity Manager tasks as web services. TEWS requests are submitted to Identity Manager as if they were initiated in the Identity Manager User Console.

Identity Manager provides documentation and samples that illustrate how to submit batch changes using TEWS. See the *CA Identity Manager Programming Guide for Java* for details.

- etutil Batch Utility—You can use this command line utility to submit batch changes to the Provisioning Server.

Identity Manager can apply identity policies or workflow processes to any batch feed transaction initiated using one of these methods.

Custom Endpoint Connectors

Existing eTrust Admin implementations may manage custom endpoints. To enable Identity Manager to read custom namespace policies and accounts that are associated with provisioning roles, there are additional procedures required.

Note: For instructions, see the appendix on custom endpoint options in the *Programming Guide for Provisioning*.

Next Steps

After deciding which components and functionality to implement, refer to the following guides for installation and configuration instructions:

- *Installation Guide*
- *Administration Guide*

Appendix A: Third-Party Acknowledgements

This section contains the following topics:

[Apache](#) (see page 97)
[ANTLR](#) (see page 101)
[DOM4J](#) (see page 102)
[Jaxen](#) (see page 103)
[Rhino](#) (see page 104)
[JDOM](#) (see page 113)
[SAXPath](#) (see page 115)

Apache

Portions of this product include software developed by the Apache Software Foundation.

Apache Ant

Apache Axis

Apache Jakarta Commons BeanUtils

Apache Jakarta Commons Codec

Apache Jakarta Commons Collections

Apache Jakarta Commons DBCP

Apache Jakarta Commons Validator

Apache Commons Digester

Apache Commons Discovery

Apache Commons EL

Apache Commons File Upload

Apache Commons Lang

Apache Commons Logging

Apache Commons Pool

Apache Jakarta ORO

Apache Jakarta Slide

Apache HttpClient

Apache MyFaces Tomahawk

Apache JSTL Taglib

Apache Struts

Apache Velocity

The Apache software is distributed in accordance with the following license agreement.

Apache License Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

'License' shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

'Licensor' shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

'Legal Entity' shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, 'control' means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

'You' (or 'Your') shall mean an individual or Legal Entity exercising permissions granted by this License.

'Source' form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

'Object' form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and versions to other media types.

'Work' shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

'Derivative Works' shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

'Contribution' shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, 'submitted' means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as 'Not a Contribution.'

'Contributor' shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a 'NOTICE' text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an 'AS IS' BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

ANTLR

Portions of this product include software developed by the ANTLR.org. The ANTLR software is distributed in accordance with the following license agreement.

ANTLR 3 License

[The BSD License]

Copyright (c) 2005, Terence Parr All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DOM4J

Portions of this product include software developed by the DOM4J Project (<http://dom4j.org/>) and is distributed in accordance with the following license agreement.

BSD Style License

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain copyright statements and notices.

Redistributions must also contain a copy of this document. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.

Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.

Due credit should be given to the DOM4J Project - <http://www.dom4j.org> THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved.

Jaxen

Portions of this product include software developed by the Jaxen Project (<http://www.jaxen.org/>) and is distributed in accordance with the following license agreement.

/*

\$Id: LICENSE.txt,v 1.3 2003/06/29 18:22:02 ssanders Exp \$

Copyright 2003 (C) The Werken Company. All Rights Reserved.

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name "jaxen" must not be used to endorse or promote products derived from this Software without prior written permission of The Werken Company. For written permission, please contact bob@werken.com.

4. Products derived from this Software may not be called "jaxen" nor may "jaxen" appear in their names without prior written permission of The Werken Company. "jaxen" is a registered trademark of The Werken Company.

5. Due credit should be given to The Werken Company.
(<http://jaxen.werken.com/>).

THIS SOFTWARE IS PROVIDED BY THE WERKEN COMPANY AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE WERKEN COMPANY OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Rhino

The source code version of Rhino 1.6 Release 4 is licensed under the Mozilla Public License Version 1.1 which can be found at <http://www.mozilla.org/MPL/> and is made available for download from http://opensrcd.ca.com/ips/P02056_4/.

MOZILLA PUBLIC LICENSE Version 1.1 -----

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant. The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims: (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and (b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof). (c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License. (d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant. Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license (a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and (b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination). (c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code. (d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License. The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code. Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications. You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters (a) Third Party Claims. If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained. (b) Contributor APIs. If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file. (c) Representations. Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices. You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions. You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation. If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License. This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions. Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions. Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works. If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY. COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that: (a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above. (b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS. The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE. Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is _____. The Initial Developer of the Original Code is _____. Portions created by _____ are Copyright (C) _____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[_____] License"), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [_____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [_____] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [_____] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

JDOM

This product includes software developed by the JDOM Project (<http://www.jdom.org/>). The JDOM software is distributed in accordance with the following license agreement.

\$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp \$

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact .
4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management . In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the JDOM Project (<http://www.jdom.org/>).\" Alternatively, the acknowledgement may be graphical using the logos available at <http://www.jdom.org/images/logos>.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter and Brett McLaughlin . For more information on the JDOM Project, please see .

SAXPath

This product includes software developed by the SAXPath Project (<http://www.saxpath.org/>). The SAXPath software is distributed in accordance with the following license agreement.

```
/*-- $Id: LICENSE,v 1.1 2002/04/26 17:43:56 jstrachan Exp $
```

Copyright (C) 2000-2002 werken digital. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "SAXPath" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact license@saxpath.org.
4. Products derived from this software may not be called "SAXPath", nor may "SAXPath" appear in their name, without prior written permission from the SAXPath Project Management (pm@saxpath.org). In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the SAXPath Project (<http://www.saxpath.org/>)." Alternatively, the acknowledgment may be graphical using the logos available at <http://www.saxpath.org/>

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE SAXPath AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the SAXPath Project and was originally created by bob mcwhirter and James Strachan . For more information on the SAXPath Project, please see . */

Index

A

- Account Synchronization • 40
- Account Templates • 41
- Additional Components • 33
- Additional Requirements for Provisioning • 47
- Additional Requirements for SiteMinder Integration • 48
- Addressing Business Needs • 19
- Admin Roles for User Account Management • 11
- ANTLR • 101
- Apache • 97
- Applying Custom Business Logic • 23
- Approving Business Changes • 25
- Audit Database • 42
- Audit Settings • 42

B

- Basic Installation • 35
- Batch Processing in Identity Manager Implementations • 95
- Business Changes in Identity Manager • 92
- Business Logic Task Handler Considerations • 24

C

- C++ Connector Server • 31
- CA Audit Considerations • 43
- CA Product References • iii
- CA Security Command Center (SCC) Considerations • 43
- Choose a Method to Import Users • 48
- Combined User Store and Provisioning Directory • 29
- Compliance Reports • 21
- Complying with Business Policies • 20
- Connector Components • 31
- Connector Servers • 31
- Connectors and Agents • 32
- Contact Technical Support • iii
- Custom Endpoint Connectors • 96

D

- Databases • 30

- Decide Hardware Requirements • 46
- Decide User Store Requirements • 43
- Decide What to Manage • 39
- Deploy Delegated Administration for Roles • 57
- Deploy Delegated Administration for Users, Groups and Organizations • 57
- Deploy Identity Policies • 54
- Deploy Self-Service and Password Management • 53
- Deploy Workflow Approvals • 56
- Design Efficient Identity Policies • 81
- Determine Audit Requirements • 41
- Develop a Deployment Plan • 53
- Develop a Transition Plan • 91
- DOM4J • 102

E

- Endpoint Management • 40
- Environment and Task Level Workflow Approvals • 56
- eTrust Admin Management Interfaces • 93
- Execute Identity Policies on Imported Users • 50
- Explore and Correlate Functionality • 41

G

- Guidelines for Group Member\Administrator Optimizations • 77
- Guidelines for Optimizing Tasks • 76
- Guidelines for Policy Rule Creation • 68

H

- How Identity Manager Renders Relationship Tabs • 73
- How JMS Messages Drive Event Transactions • 86
- How Role Evaluation Affects Performance at Login • 64
- How to Choose a User Store Solution • 44
- How to Import Users into a New User Store • 48
- How Users and Identity Policies Are Synchronized • 79

I

- IA Manager Considerations • 94
- IAM Report Server • 34
- Identity Manager Architecture • 27
- Identity Manager Auditing Considerations • 42
- Identity Manager Components • 27
- Identity Manager Customization and Extensibility • 16
- Identity Manager Performance • 63
- Identity Policy Optimizations • 79
- Import Users Through Identity Manager • 49
- Import Users Through the Provisioning Server • 51
- Installation with Provisioning Components • 36
- Installation with SiteMinder Policy Server • 37
- Integrating with SiteMinder • 59
- iRecorder for Auditing • 34

J

- Java Connector Server • 32
- Jaxen • 103
- JDOM • 113
- JMS Messages and Performance • 87
- JMS Settings • 86

L

- Limit Policy Objects and User Store Searches • 69
- Limit the Tasks that Trigger User Synchronization • 82
- Logical Attribute Handlers • 23

M

- Managing Identities and Access • 9
- Managing Multiple User Stores • 44

N

- Next Steps • 96

O

- Object Store Performance • 67
- One-Step Approach • 50
- Optimize Identity Policy Rule Evaluation • 82
- Optimize Role Policy Evaluation • 67
- Optimizing Identity Manager • 63

P

- Password Management • 15
- Password Management Considerations • 94
- Password Policies with SiteMinder • 61
- Planning Your Implementation • 39
- Processing Business Changes • 19
- Profile Management at the Attribute Level • 12
- Provisioning Accounts from Other Applications • 40
- Provisioning Manager • 34
- Provisioning Manager Considerations • 94
- Provisioning Roles for Additional Accounts • 14
- Provisioning Terminology Changes • 93

R

- Relationship Tabs and Performance • 75
- Requirements for Basic Installations • 47
- Rhino • 104
- Role Objects and Performance • 65
- Role Optimizations • 64
- Role-Based Entitlements • 10
- Runtime Components Tuning • 85

S

- Sample Identity Manager Installations • 35
- SAWI/DAWI Considerations • 94
- SAXPath • 115
- Select Components to Install • 45
- Select Scalable Policy Rule Types • 71
- Self Service Options for Users • 16
- Separate User Store and Provisioning Directories • 28
- Servers • 27
- SiteMinder Authentication • 60
- SiteMinder Integration • 59
- Synchronize Global Users with the Identity Manager User Store • 52

T

- Task Optimizations • 72
- Task Scope Evaluation and Performance • 73
- Third-Party Acknowledgements • 97
- Transforming Data in the User Store • 23
- Transitioning From eTrust Admin to Identity Manager • 91
- Tuning for Provisioning Components • 84
- Tuning Identity Manager Databases • 85
- Tuning JMS Settings • 88

Two-Step Approach • 50

U

User Entitlement Certification • 20

User Identities • 39

User Management and Application Access • 9

User Store and Provisioning Directory • 28

User Store Tuning • 83

W

Workflow Approval of Admin Tasks • 13

Workflow Process Considerations • 25

WorkPoint Workflow • 33