Oracle® Access Manager

Deployment Guide

10*g* (10.1.4.3)

E12490-01

May 2009

This guide provides general Oracle Access Manager recommendations and specific capacity planning and sizing, performance tuning, failover, load balancing, caching, and data migration recommendations for administrators who plan and manage deployments.

Oracle Access Manager Deployment Guide, 10*g* (10.1.4.3)

E12490-01

Contributing Author: Gail Flanegin

Contents

2 Capacity Planning

3 Performance Tuning

5 Cloning and Caching

6 Reconfiguring the System

7 Synchronizing System Clocks Across Time Zones

8 About Upgrading

9 Oracle Access Manager Backup and Recovery Strategies

Index

Preface

This Deployment Guide provides information for people who plan and manage the environment in which Oracle Access Manager is to run. This guide covers capacity planning, network topologies and system tuning.

Note: Oracle Access Manager was previously known as Oblix NetPoint.

This Preface covers the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This guide targets the knowledge and skill requirements of system, network, or Oracle Access Manager administrators who are responsible for optimizing the implementation.

This document assumes that you are familiar with your network architecture, your LDAP directory, and firewall and internet security.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Related Documents

For more information, see the following documents in the Oracle Access Manager Release 10*g* (10.1.4) documentation set:

- *Oracle Access Manager Introduction*—Provides an introduction to Oracle Access Manager, a road map to the manuals, and a glossary of terms.
- *Oracle Access Manager Release Notes*—Read these for the latest Oracle Access Manager information.
- *Oracle Access Manager Patchset Notes Release 10.1.4 Patchset 2 (10.1.4.3.0) For All Supported Operating Systems*—Read this document if you want to apply the 10*g* (10.1.4.3) patch set to an existing 10*g* (10.1.4.2.0) deployment. It includes a list of enhancements, bug fixes, and known issues related to the patch set.
- *Oracle Access Manager Installation Guide*—Explains how to prepare for, install, and set up each Oracle Access Manager component.
- *Oracle Access Manager Upgrade Guide*—Explains how to upgrade earlier releases to the latest major Oracle Access Manager release using either the in-place component upgrade method or the zero downtime method.
- *Oracle Access Manager Identity and Common Administration Guide*—Explains how to configure Identity System applications to display information about users, groups, and organizations; how to assign permissions to users to view and modify the data that is displayed in the Identity System applications; and how to configure workflows that link together Identity application functions, for example, adding basic information about a user, providing additional information about the user, and approving the new user entry, into a chain of automatically performed steps. This book also describes administration functions that are common to the Identity and Access Systems, for example, directory profile configuration, password policy configuration, logging, and auditing.
- *Oracle Access Manager Access Administration Guide*—Describes how to protect resources by defining policy domains, authentication schemes, and authorization schemes; how to allow users to access multiple resources with a single login by configuring single- and multi-domain single sign-on; and how to design custom login forms. This book also describes how to set up and administer the Access System.

- *Oracle Access Manager Deployment Guide*—Provides information for people who plan and manage the environment in which Oracle Access Manager runs. This guide covers capacity planning, system tuning, failover, load balancing, caching, and migration planning.
- *Oracle Access Manager Customization Guide*—Explains how to change the appearance of Oracle Access Manager applications and how to control operation by making changes to operating systems, Web servers, directory servers, directory content, or by connecting CGI files or JavaScripts to Oracle Access Manager screens. This guide also describes the Access Manager API and the authorization and authentication plug-in APIs.
- *Oracle Access Manager Developer Guide*—Explains how to access Identity System functionality programmatically using IdentityXML and WSDL, how to create custom WebGates (known as AccessGates), and how to develop plug-ins. This guide also provides information to be aware of when creating CGI files or JavaScripts for Oracle Access Manager.
- *Oracle Access Manager Integration Guide*—Explains how to set up Oracle Access Manager to run with other Oracle and third-party products.
- *Oracle Access Manager Schema Description*—Provides details about the schema.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in Oracle Access Manager?

This section describes new features of the Oracle Access Manager release 10.1.4. This includes details for 10*g* (10.1.4), 10*g* (10.1.4.2.0), and 10*g* (10.1.4.3).

The following sections are included:

- Product and Component Name Changes
- Enhancements Available in 10g (10.1.4.3)
- Deployment Overview
- Access System Performance Enhancements for Large Group Evaluations
- Cache Flush Enhancements
- Capacity Planning
- Failover and Load Balancing
- Migrating Data
- Reconfiguring Oracle Access Manager
- Tuning the Directory
- Tuning the Access Server
- Tuning the Identity System
- Tuning Workflows
- Tuning Your Network
- Tuning Performance for Access System Operations

> **Note:** For a comprehensive list of all new features and functions in Oracle Access Manager 10.1.4, and a description of where each is documented, see the chapter on what's new in the *Oracle Access Manager Introduction*.

Product and Component Name Changes

The original product name, Oblix NetPoint, has changed to Oracle Access Manager. Most component names remain the same. However, there are several important changes that you should know about, as shown in the following table:

Item	Was	Is
Product Name	Oblix NetPoint Oracle COREid	Oracle Access Manager
Product Name	Oblix SHAREid NetPoint SAML Services	Oracle Identity Federation
Product Name	OctetString Virtual Directory Engine (VDE)	Oracle Virtual Directory
Product Name	BEA WebLogic Application Server BEA WebLogic Portal Server	Oracle WebLogic Server Oracle WebLogic Portal
Product Release	Oracle COREid 7.0.4	Also available as part of Oracle Application Server 10g Release 2 (10.1.2).
Directory Name	COREid Data Anywhere	Data Anywhere
Component Name	COREid Server	Identity Server
Component Name	Access Manager	Policy Manager
Console Name	COREid System Console	Identity System Console
Identity System Transport Security Protocol	NetPoint Identity Protocol	Oracle Identity Protocol
Access System Transport Protocol	NetPoint Access Protocol	Oracle Access Protocol
Administrator	NetPoint Administrator COREid Administrator	Master Administrator
Directory Tree	Oblix tree	Configuration tree
Data	Oblix data	Configuration data
Software Developer Kit	Access Server SDK ASDK	Access Manager SDK
API	Access Server API Access API	Access Manager API
API	Access Management API Access Manager API	Policy Manager API
Default Policy Domains	NetPoint Identity Domain COREid Identity Domain	Identity Domain
Default Policy Domains	NetPoint Access Manager COREid Access Manager	Access Domain
Default Authentication Schemes	NetPoint None Authentication COREid None Authentication	Anonymous Authentication
Default Authentication Schemes	NetPoint Basic Over LDAP COREid Basic Over LDAP	Oracle Access and Identity Basic Over LDAP
Default Authentication Schemes	NetPoint Basic Over LDAP for AD Forest COREid Basic Over LDAP for AD Forest	Oracle Access and Identity for AD Forest

Item	Was	Is
Access System Service	AM Service State Policy Manager API Support Mode	Access Management Service **Note**: Policy Manager API Support Mode and Access Management Service are used interchangeably.

All legacy references in the product or documentation should be understood to connote the new names.

Enhancements Available in 10*g* (10.1.4.3)

Included in this release are new enhancements and bug fixes for 10*g* (10.1.4.3) in addition to all fixes and enhancements from 10*g* (10.1.4.2.0) bundle patches through BP07. The following topics describe 10*g* (10.1.4.3) enhancements described in this book:

- Access System Performance Enhancements for Large Group Evaluations
- Asynchronous Cache Flush Operations Between Identity and Access Servers
- Error Handling for Message Channel Initialization During Cache Flush
- Identity System Performance Enhancements for Large Group Evaluations
- Mixed-Mode Communication for Cache Flush Operations
- Multi-Language Deployments and English Only Messages
- Reconfiguring Oracle Access Manager
- Synchronous Cache Flush Between Multiple Access Servers
- Tuning the Internal DBAgent Cache

> **See Also:** *Oracle Access Manager Introduction* for a list of all new features and functions

Access System Performance Enhancements for Large Group Evaluations

The following Access System performance enhancements for large group evaluations are provided with Oracle Access Manager 10*g* (10.1.4.3):

- The Access Server (and Policy Manager when using the Access Tester) evaluates the group for membership as a type, only if that type is enabled. To improve performance during group evaluations when you do not use dynamic groups, or when you have dynamic groups but do not want to evaluate them while processing ObMyGroups, you can turn off dynamic group evaluation using the `TurnOffDynamicGroupEvaluation` parameter in the Access Server (or Policy Manager) globalparams.xml file.

  Access Server v7.0.2 provided the ability to disable nested group evaluation using the `TurnOffNestedGroupEvaluation` parameter in the Access Server globalparams.xml file.

  > **See Also:** "Improving Performance During Group Search When Dynamic Groups Are Not Used" on page 3-38

- In 10*g* (10.1.4.3), a new algorithm can be used during group evaluation involving ObMyGroups: `TurnOffNewAlgorithmForObmyGroups`. This algorithm in the

Access Server globalparams.xml file works equally when you have static, dynamic, and nested groups.

> **See Also:** "Improving Performance of ObMyGroups Evaluations" on page 3-41

- The `NestedQueryLDAPFilterSize` parameter can be used In the Access Server globalparams.xml file, if `TurnOffNewAlgorithmForObmyGroups` is `false`. This improves evaluation performance of ObMyGroups. With this parameter, the LDAP search query is divided and then executed. For more information, see the table on globalparams.xml in the chapter on parameters in the *Oracle Access Manager Customization Guide*.

> **See Also:** "Improving Performance of ObMyGroups Evaluations" on page 3-41

- The `GroupCacheTimeout` parameter enables you to specify the amount of time an element remains valid in the Access Server group cache. The parameter must be added to the Access Server globalparams.xml file (or the Policy Manager file if you are using the Access Tester).

> **See Also:** "Configuring the Access Server Group Cache Timeout and Maximum Elements" on page 3-43

- The `GroupCacheMaxElement` parameter specifies the maximum number of elements that can be stored in the Access Server group cache. The parameter must be added to the Access Server globalparams.xml file (or the Policy Manager file if you are using the Access Tester).

> **See Also:** "Configuring the Access Server Group Cache Timeout and Maximum Elements" on page 3-43

Asynchronous Cache Flush Operations Between Identity and Access Servers

Oracle Access Manager 10*g* (10.1.4.3) provides an asynchronous cache flush option to help streamline performance and avoid delays associated with synchronous cache flush operations on the Access System. With the asynchronous method, the request arrives at the Access Server and a response is sent immediately to the Identity Server without a delay.

> **See Also:** "Configuring Asynchronous Access System Cache Flush" on page 5-31

Error Handling for Message Channel Initialization During Cache Flush

Oracle Access Manager 10*g* (10.1.4.3) enhances the network layer shared by WebGate and Access Server. As a result, errors that might occur as a result of message channel initialization failure (due to a socket with an unlimited time period) are avoided. Today, the message channel stops sending and receiving messages and a WARNING level log message is recorded.

> **See Also:** "Error Handling for Message Channel Initialization During Cache Flush" on page 5-24

Identity System Performance Enhancements for Large Group Evaluations

In the groupdbparams.xml file, `TurnOffDynamicGroupEvaluation` and `TurnOffNestedGroupEvaluation` can be set to `true` to enhance performance during group evaluation by eliminating dynamic or nested groups when these are not used.

> **See Also:** Parameters chapter in the *Oracle Access Manager Customization Guide* and Chapter 3 in this guide

Mixed-Mode Communication for Cache Flush Operations

When installing and configuring Oracle Access Manager, specific transport security guidelines must be observed, as described in previous topics. After installation and setup, you can choose to use mixed-mode communication for cache flush operations.

Oracle Access Manager 10*g* (10.1.4.2.0) provided a method that enabled you to use Open mode communication for cache flush requests between the Identity and Access Server while retaining Simple or Cert mode for all other requests. This type of configuration is known as mixed security mode (or mixed transport security mode) communication. Oracle Access Manager 10*g* (10.1.4.3) provides a streamlined method to implement mixed-mode communication for cache flush requests.

> **See Also:** "Enhancing Performance by Configuring Mixed-Mode Communication for Access Server Cache Flush Operations" on page 5-25

Multi-Language Deployments and English Only Messages

Oracle Access Manager 10*g* (10.1.4.3) provides new Language Pack installers. 10*g* (10.1.4.3) Language Packs are required in any 10*g* (10.1.4.3) deployment, whether it is a fresh installation or an upgraded and patched deployment.

Functionality that is new with 10*g* (10.1.4.2.0) and 10*g* (10.1.4.3) can include new messages, which might not be translated and could appear in only English.

> **See Also:** *Oracle Access Manager Installation Guide*.

Native POSIX Thread Library (NPTL) for Linux

Earlier releases of Oracle Access Manager for Linux used the LinuxThreads library only. Using LinuxThreads required that you set the environment variable LD_ ASSUME_KERNEL, which is used by the dynamic linker to decide what implementation of libraries is used. When you set LD_ASSUME_KERNEL to 2.4.19 the libraries in /lib/i686 are used dynamically.

RedHat Linux v5 and later releases support only Native POSIX Thread Library (NPTL), not LinuxThreads. To accommodate this change, Oracle Access Manager 10*g* (10.1.4.3) is compliant with NPTL specifications. However, LinuxThreads is used by default for all except Oracle Access Manager Web components for Oracle HTTP Server 11g.

> **Note:** On Linux, Oracle Access Manager Web components for Oracle HTTP Server 11g use only NPTL; you cannot use the LinuxThreads library. In this case, do not set the environment variable LD_ ASSUME_KERNEL to 2.4.19.

> **See Also:** *Oracle Access Manager Installation Guide*.

Reconfiguring Oracle Access Manager

Updates and additions have been made to this topic:

- You can change basic components that you specified during Oracle Access Manager installation, such as the person object class or the directory server host.

  See Also: "Reconfiguring the System" on page 6-1.

- New examples of updating the LDAP bind password now include a missing required parameter -i *install_dir* and other clarifications.

  See Also: "Updating the LDAP Bind Password" on page 6-3.

Synchronous Cache Flush Between Multiple Access Servers

Oracle Access Manager 10*g* (10.1.4.3) provides a new function that enables you to specify a wait period for sockets during synchronous cache flush requests between multiple Access Servers. In this case, a socket waits for only a specified time for I/O completion. If the expected operation is not completed within the specified time, an error is reported and the request is sent to other Access Servers. With synchronous requests, WebPass and Policy Manager does not hang if one Access Server hangs.

> **See Also:** "Configuring Synchronous Cache Flush Requests between Multiple Access Servers" on page 5-22

Tuning the Internal DBAgent Cache

In the Identity Server globalparams.xml file, you can use the `negativeListForEntityAttributes` parameter to identify specific attributes that are not read or cached during view and modify profile operations.

> **See Also:** "Tuning the Internal DBAgent Cache" on page 3-48

Deployment Overview

A new chapter has been added to discuss deployment types and tiers, deployment scenarios and environments, deployment categories, and deployment guidelines.

> **See Also:** Chapter 1

Access System Performance Enhancements for Large Group Evaluations

The following Access System performance enhancements for large group evaluations are provided with Oracle Access Manager 10*g* (10.1.4.3):

- The Access Server (and Policy Manager when using the Access Tester) evaluates the group for membership as a type, only if that type is enabled. To improve performance during group evaluations when you do not use dynamic groups, or when you have dynamic groups but do not want to evaluate them while processing ObMyGroups, you can turn off dynamic group evaluation using the `TurnOffDynamicGroupEvaluation` parameter in the Access Server (or Policy Manager) globalparams.xml file.

  Access Server v7.0.2 provided the ability to disable nested group evaluation using the `TurnOffNestedGroupEvaluation` parameter in the Access Server globalparams.xml file.

See Also: Parameters chapter in the *Oracle Access Manager Customization Guide* and Chapter 3

- In 10*g* (10.1.4.3), a new algorithm can be used during group evaluation involving ObMyGroups: `TurnOffNewAlgorithmForObmyGroups`. This algorithm in the Access Server globalparams.xml file works equally when you have static, dynamic, and nested groups.

  See Also: "Improving Performance of ObMyGroups Evaluations" on page 3-41

- In the Access Server globalparams.xml file, you can use the `NestedQueryLDAPFilterSize` parameter if `TurnOffNewAlgorithmForObmyGroups` is `false` to improve evaluation performance of ObMyGroups. With this parameter, the LDAP search query is divided and then executed.

  See Also: "Improving Performance of ObMyGroups Evaluations" on page 3-41

- The `GroupCacheTimeout` parameter enables you to specify the amount of time an element remains valid in the Access Server group cache. The parameter is included in the Access Server globalparams.xml file (or the Policy Manager file if you are using the Access Tester).

  See Also: "Configuring the Access Server Group Cache Timeout and Maximum Elements" on page 3-43

- The `GroupCacheMaxElement` parameter specifies the maximum number of elements that can be stored in the Access Server group cache. The parameter is provided in the Access Server globalparams.xml file (or the Policy Manager file if you are using the Access Tester).

  See Also: "Configuring the Access Server Group Cache Timeout and Maximum Elements" on page 3-43

Cache Flush Enhancements

Several cache flush enhancements are available with Oracle Access Manager 10*g* (10.1.4.3), and new information is provided on these as follows:

- Asynchronous cache flush from the Identity System to the Access System

  See Also: "Configuring Asynchronous Access System Cache Flush" on page 5-31.

- Enhancing performance using mixed mode communication for cache flush requests

  See Also: "Enhancing Performance by Configuring Mixed-Mode Communication for Access Server Cache Flush Operations" on page 5-25

- Synchronous cache flush operations between multiple Access Servers using a specified time period for I/O completion

See Also: "Configuring Synchronous Cache Flush Requests between Multiple Access Servers" on page 5-22

- New handling of message channel initialization failures

  See Also: "Error Handling for Message Channel Initialization During Cache Flush" on page 5-24

- Chapter 5 has been reorganized and updated to provide more background information and clarify caching and cache flush operations
- In the Access Server globalparams.xml file, the `UserMgmtNodeEnabled` parameter can be used. This parameter controls the enabling and disabling of a feature that manages WebGate memory growth. For more information, see the chapter on parameters in the *Oracle Access Manager Customization Guide*.. See also, the tip on "Cache Flush Issues with Active Directory" in the *Oracle Access Manager Access Administration Guide*.

  See Also: "Automatically Flushing Access Server Caches" on page 5-17

Capacity Planning

The chapter that describes capacity planning has been updated to provide even more helpful details.

See Also: Chapter 2

Failover and Load Balancing

- Information has been added on load balancing of LDAP data.

  See Also: "About Load Balancing of LDAP Data" on page 4-2.

- A "heartbeat" polling mechanism facilitates immediate failover to a secondary directory server when the number of connections in the connection pool falls below the specified threshold level. Information has been added on setting the polling interval for failover.

  See Also: "Configuring Failover Based on Directory Server Availability" on page 4-19.

- Information on configuring failover for Policy Manager data has been added.

  See Also: "Configuring Directory Failover for Configuration and Policy Data" on page 4-15.

Migrating Data

The Oracle Access Manager Configuration Manager has been deprecated and is no longer available. The overview has been removed from Chapter 8 of this guide.

Reconfiguring Oracle Access Manager

You can change basic components that you specified during Oracle Access Manager installation, such as the person object class or the directory server host.

See Also: "Reconfiguring the System" on page 6-1.

Tuning the Directory

- Several enhancements have been made to directory tuning documentation.

  To optimize performance, you should ensure that your directory performance is optimal. In this release, information on directory tuning has been enhanced.

  New guidelines for configuring the directory connection pool size has been added.

  See Also:

  - "Guidelines for Directory Tuning" on page 3-1
  - "Checking the Performance of the Directory" on page 3-2

- This release provides a new parameter for clearing the LDAP connection cache.

  See Also: "Directory Connection Pool Size" on page 3-2.

- New parameters enable a component to fail over to a secondary directory server if the primary server takes too long to respond or too long to process a request.

  See Also:

  - "Configuring Failover Based on Directory Server Availability" on page 4-19
  - "Configuring Failover Based on Directory Server Response Time" on page 4-20

Tuning the Access Server

- Guidelines have been provided on configuring threads and queues, configuring group searches, and tuning Policy Manager LDAP searches.

  See Also:

  - "Changing the Number of Request Queues and Threads" on page 3-34
  - "Tuning the Handling of Groups in the Access System" on page 3-37
  - "Tuning the LDAP Search Filter in the Policy Manager" on page 3-45

Tuning the Identity System

- Guidelines have been provided for optimizing directory searches that users perform with the Identity System applications.

  See Also: "Tuning Identity System Searches" on page 3-21.

- Guidelines are provided to improve performance during group evaluations when you do not use dynamic groups or nested groups.

  See Also: "General Recommendations for Tuning Groups in the Identity System" on page 3-25

- Guidelines are provided to optimize performance of the Group Manager application in the Identity System.

  See Also: "Tuning the Group Manager Application" on page 3-30

Tuning Workflows

- There are best practices for optimizing workflow performance.

  To minimize the impact that workflows have on server performance, you can tune various parameters in workflowdbparams.xml. You can also tune various workflow search parameters to enhance performance.

  See Also: "Tuning Workflows" on page 3-32.

Tuning Your Network

- There are best practices for optimizing network and Oracle Access Manager performance.

  See Also: "Tuning Your Network" on page 3-50.

Tuning Performance for Access System Operations

- If you do not use nested groups in your directory, you can improve group membership searches by turning off nested group evaluation.

  See Also: "Use Nested Groups with Caution" on page 3-26.

1

Oracle Access Manager Deployment Overview

This book provides methods, procedures, and guidelines to help you and your team successfully deploy Oracle Access Manager. This chapter provides an overview of Oracle Access Manager deployments and includes the following topics:

For a general overview of Oracle Access Manager, see the *Oracle Access Manager Introduction*.

About Oracle Access Manager Deployment Types and Tiers

Oracle Access Manager deployment types can be described as either an Identity System *only* type deployment, or as a *joint* Identity and Access System type deployment.

The Oracle Access Manager Identity System is comprised of the following three tiers:

- **Presentation Tier**: WebPass
- **Application Tier**: Identity Server
- **Data Tier**: A back-end Lightweight Directory Access Protocol (LDAP) directory

In general, the Oracle Access Manager Identity System is a *CPU intensive* system. Identity System performance increases significantly with increased CPU power.

The Oracle Access Manager Access System *cannot* be deployed without the Identity System. The Access System is comprised of the following three tiers:

- **Presentation Tier**: WebGates and custom AccessGates
- **Application Tier**: Policy Manager and Access Server
- **Data Tier**: A back-end LDAP directory

Generally speaking, the Oracle Access Manager Access System is a *memory intensive* system. Access System performance increases significantly with increased system memory.

Whether you choose to deploy the Identity System only, or a joint Identity and Access System, your enterprise may have more than one deployment scenarios. For more information, see "Deployment Scenarios and Environments".

Deployment Scenarios and Environments

Many companies provide more than one deployment, each with a specific audience and purpose. Providing multiple deployments helps minimize service disruptions. For example, your company may have one or more of the following independent deployments:

- Development and Test Deployment

  This deployment is where you configure and test a fully configured and operational Oracle Access Manager system. On a development or test server, a smaller amount of RAM may be required. Assuming that only a small development team is using these systems (not the whole user population), you can run the Web server on the same host computer as the Identity Server.

- Staging Deployment

  This deployment is used for new application rollouts, software upgrades, and performance benchmarking can be performed without affecting your production and development environments. This deployment may closely resemble the production environment. However, the staging deployment may be a scaled down image of the production deployment. If so, a smaller amount of RAM may be used.

- Production Deployment

  This is fully deployed system that your end users can access. Oracle recommends you use a replicated directory and the native load-balancing features of Oracle Access Manager. You may also want to configure the Oracle Access Manager Servers for failover and load balancing and have separate computers for your Web servers.

These are a few examples. Your company may include other deployment scenarios that target the needs of the QA team or integration team.

Whether you have one or several deployment scenarios, each falls into one of two categories. For more information, see "Deployment Categories".

Deployment Categories

Oracle Access Manager deployments can be classified into two primary categories: Extranet (B2B,G2C, B2C) and Intranet (B2E, G2E) deployments. While these are, generic categories they do provide some relevant patterns about deployment demographics.

For more information, see the topics:

- Extranet Deployment Category
- Intranet Deployment Category

Extranet Deployment Category

Extranet deployments are those where you have:

- A relatively large user population (over 20 thousand users)
- The user population is being served through a relatively small number of applications (less than 20)
- The applications are integrated with NetPoint (Oracle Access Manager), and are typically consolidated in a portal

The most typical characteristics for extranet deployments include:

- A higher complexity on the Identity System deployment relative to the Access System
- A large number of workflows (self-registration, self-service, delegated administration) typically involving Identity Event plug-ins (customizations)
- Sophisticated delegated administration requirements, often involving various user types (at a minimum four levels of administrative roles/access) and reliance on ACLs, groups, and other objects.
- User interface customizations (accomplished using XSL stylesheets, PresentationXML, and IdentityXML) because the majority of the requirements center on identity administration of a large number of users and ease of use is a paramount driver. The majority of implementations exhibit front-end user interfaces built on top of IdentityXML.
- Features such as lost password management are very commonly configured.
- A relatively small software footprint (for example, only a handful of servers—2 to 4 servers at each tier—often distributed between a few data centers), and a very low tolerance for downtime because the applications that rely on Oracle Access Manager are often business critical.
- Commonly the directory environment is dedicated to Oracle Access Manager and the applications it supports. Therefore, there is a bit more control over the directory service in conjunction with Oracle Access Manager from an operational perspective. There are a relatively small number of stakeholders from the application side (typically belonging to a common line of business.)

Performing the upgrade to 10*g* (10.1.4) with minimal service disruption in such a highly complex environment can be challenging.

Intranet Deployment Category

Intranet deployment environments are typically:

- Internal facing portals with a relatively small user population (less than 20 thousand users)
- The user population is being served through a relatively large number of applications (more than 20) integrated with NetPoint (also known as Oracle Access Manager)

The most typical characteristics for intranet deployments include:

- A greater prevalence of the Access System customizations, if any, are typically:
  - On the front-end at the login page (or login front-end)
  - Or using custom built AccessGates

  - Or on the back-end using customized authentication or authorization plug-ins developed with the APIs

- A relatively large number of applications (over 20) being protected where the emphasis is primarily on authentication and single-sign on (SSO), with a significant number of application-level integrations.
- A high number of Oracle WebLogic and IBM WebSphere Application Server integrations using Oracle Access Manager connectors for these servers.
- Often the Identity System is either not widely deployed, or deployed only to an administrator user community (for example, the help desk, IT department, or system administrators).
- Password management features are not typically configured or used, because Oracle Access Manager often relies on the same back end store as the NOS (AD), and it is rare to see self-registration workflows.
- These environments tend to have a broad footprint, especially at the WebGate/AccessGate tier, with a high number of Web servers and Application servers with WebGate to Access Server ratios in the range of 10:1.
- On the Access Server tier, intranet deployments tend to be global and geographically distributed, with a handful of servers deployed in each location.
- The directory environment is often shared, because it is the employee directory or even the NOS directory (AD). Therefore, the number of dependencies associated to the directory is high (meta-directories, provisioning solutions, NOS logon, white pages, and the like). As a result, changes and operational impact to the directory is very rigorously managed. Many stakeholders must be coordinated with in a change-control process, and tight operational windows are allowed. On the application front, there tends to be more flexibility on server availability, and applications tend to be "clustered" by line of business, geography, or security requirements. Therefore, the impact can be segregated.

General Recommendations

Any deployment may be installed at a single site or across multiple sites whether Identity System only or a joint Identity and Access System, whether intranet or extranet.

Oracle Access Manager supports a variety of operating systems, directory servers, Web servers, compilers, and browsers, as well as integration with a number of application servers, portal servers, system management products, and packaged applications. For the latest support information, see details on Oracle Technology Network (OTN):

http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html

To locate the latest certification details

1. Go to Oracle Technology Network:

   http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html

2. Locate and click the link for Oracle Access Manager Certification.

The following topics provide:

- Security Recommendations

- Standardization Recommendations
- Oracle Access Manager Server Recommendations
- Web Server Recommendations
- LDAP Directory and Data Recommendations
- Audit Data Usability Recommendations
- Configuring a Single Idle Timeout for the Entire Deployment
- Customization Recommendations
- Testing and Performance Recommendations

See Also:

- "Identity System Recommendations" on page 1-9
- "Access System Recommendations" on page 1-10
- Capacity and sizing recommendations in Chapter 2
- Performance tuning recommendations in Chapter 3

Security Recommendations

In any production environment, Oracle recommends that you secure all transport between Oracle Access Manager components using either Simple or Certificate Mode. All intranet and extranet communications should be encrypted.

When enabling SSL with Simple or Certificate mode, ensure that the clocks of the servers hosting the WebGates, AccessGates, and WebPass as well as those hosting the Access or Identity Servers are within 1 minute of absolute time difference. This includes ensuring that daylight savings and time zones have no more than a 1-minute difference between the clocks in GMT. Oracle recommends that you use the Network Time Protocol (NTP) for ensuring server clocks are synchronized.

If Oracle Access Manager is running in Simple mode, once a year by default both the Identity and Access Servers must refresh the self-signed certificate that Oracle Access Manager generates to provide SSL encryption. The frequency of updates can be configured. In this case, however, be sure to:

- Update the Simple certificates for a Web server running the WebPass.
- Update the certificate required for Policy Manager to work in Simple mode.
- Be sure the obSSOCookie is not shared between computers in the two different environments.

  This would allow a user who exists in both environments to use single sign-on from the source environment. Updating the obSSOCookie is an important security precaution when migrating across environments.

Important: Other ways of breaking SSO between different environments include changing domains, for instance, changing from dev.company.com to staging.company.com and updating the shared secret among different environments.

For complete installation and setup details, see the *Oracle Access Manager Installation Guide*. For information about changing the transport security modes after installation, see the *Oracle Access Manager Identity and Common Administration Guide*.

Standardization Recommendations

Each component should be running on a stable and appropriately installed platform. Oracle recommends that you standardize on directory server versions and Web server versions, both within and across all deployments. Also, Oracle recommends that you document the host computers, IP addresses, transport security modes, searchbases and directory profiles in your deployment. For more information, see the *Oracle Access Manager Installation Guide*.

Oracle recommends that you have a dedicated computer for the LDAP directory server within each Oracle Access Manager deployment and that you standardize the layout of the file system. For more information, see "LDAP Directory and Data Recommendations" on page 1-7.

Oracle recommends that you create a /customizations directory and document all changes to your deployment. For more information, see "Customization Recommendations" on page 1-8.

Oracle Access Manager Server Recommendations

Oracle Access Manager servers is a generic phrase that refers to the Oracle Access Manager Identity Server and Oracle Access Manager Access Server components.

To improve reliability, Oracle recommends that you install and operate Identity Servers and Access Servers on independent, dedicated hardware. This means that you need a dedicated computer for each Identity or Access Server. As an example, consider a small Oracle Access Manager deployment for up to 15,000 users. You could use four server-class computers: two installed with Identity Servers and two installed with Access Servers, with IP switching technology in front of each pair of Web servers to provide load balancing and failover.

When your deployment has light load conditions, where the primary Identity Server and Access Server have a combined utilization of 85% or less, you may conserve hardware by using a cross-over deployment. In this case, you may install a primary Identity Server and a secondary Access Server on one computer and a secondary Identity Server with a Primary Access Server installed on a different computer. Failover is configured.

Note: A cross-over deployment is only recommended when the combined utilization of the two primary Identity Server and Access Server equals 85%. Otherwise, Oracle recommends a dedicated computer for each Oracle Access Manager server.

Web Server Recommendations

Oracle recommends that you dedicate one or two internal-facing, highly-secured Web servers to host the administrative interfaces for Oracle Access Manager: Identity System Console and the Access System Console. This helps protect the authentication and authorization system in the event the application Web servers are compromised.

Oracle recommends that you install WebPass instances on a dedicated set of Web servers, especially if you expect that they will serve IdentityXML (SOAP) calls.

When you have a joint Identity and Access System deployed, Oracle recommends that you install both WebPass and WebGate against the same Web server instance This allows WebGate to provide authentication and single sign-on (SSO) to actions that go through WebPass. For example, when a user performs self-registration or an identity administrator accesses the Identity System Console.

Oracle also recommends that you standardize on Web server versions within deployments, as discussed in "Standardization Recommendations" on page 1-6.

For capacity planning and sizing guidelines, see Chapter 2. For performance tuning recommendations, see Chapter 3.

LDAP Directory and Data Recommendations

In general, the following factors impact the overall performance of the LDAP directory server and of Oracle Access Manager itself. However, your own business requirements will drive the decisions that you make regarding the following:

- DIT structure: Flat versus deep will have an impact on search operation performance
- Replication: Frequency, multi-mastering, network latency
- Disk size and I/O response time
- Disk I/O contention for attribute operations (searching and indexing) versus LDAP access and error logging
- Attribute indexing
- Cache size and life span
- Connection overhead: Long- lived connections are preferred, especially if LDAPS is used. Be aware that many current network routers and switches contain connection state tables (like traditional firewalls) and can sever connections between systems, which can result in substantial overhead for transactions (timeout plus connection recovery plus operation response time).

Oracle recommends that you have a dedicated computer for the LDAP directory server within each Oracle Access Manager deployment and that you standardize the layout of the file system. For example, consider locating all Oracle Access Manager-specific files in one particular directory path in each deployment. Oracle recommends that you use the same Web server and directory server versions across all deployments.

Also, consider storing Oracle Access Manager configuration and policy data in a separate directory from user and group data. This allows greater flexibility when upgrading to a later release and minimizes the impact on the user and group directory (typically a shared, enterprise directory). Separating the data is particularly beneficial for workflow-driven processes that generate a significant load on the directory. Configuring separate logical directory instances also ensures that each directory can be tuned and managed independently improve overall performance.

Note: If directories cannot be separated due to hardware or topology related issues, consider creating a dedicated suffix to hold the Oracle Access Manager configuration and policy data.

Oracle also recommends that you standardize on directory server versions within deployments, as discussed in "Standardization Recommendations" on page 1-6.

For more information, see "Considerations for the LDAP Directory Server" on page 2-16. For capacity planning and sizing guidelines, see Chapter 2. For performance tuning recommendations, see Chapter 3. For specific directory server requirements, see the *Oracle Access Manager Installation Guide*.

Audit Data Usability Recommendations

Whenever possible, Oracle recommends that you use a database to record and store all Oracle Access Manager audit data. This protects the audit information and makes it easier to generate audit trails and reports.

For more information about logging and auditing, see the *Oracle Access Manager Identity and Common Administration Guide*.

Configuring a Single Idle Timeout for the Entire Deployment

In general, Oracle Access Manager timeout values should be configured to be the lowest of all application timeouts. Timeouts are enforced by WebGate or the Web server, and not by application. The goal is to avoid having applications time out before Oracle Access Manager times out. Otherwise session issues within the applications may arise producing potential discrepancies in behavior

There are a number of considerations in selecting timeout values. For example, applications which are able to regenerate a session from an existing Oracle Access Manager session (or header variable) can timeout earlier than Oracle Access Manager. The Identity System, in fact, is a good example of an application where having a shorter session timeouts than WebGates is recommended.

Generally speaking. however, these timeout values should be close to each other. One exception to this rule is for AccessGates, which are typically deployed downstream from a WebGate, for example supporting a BEA WebLogic implementation. In this case it is recommended that the AccessGate have a greater idle timeout than the WebGate to avoid the problem of a fresh browser session being rejected by the downstream AccessGate. For AccessGates, Oracle recommends configuring the idle and maximum timeouts to be the same.

For more information, see the chapter on configuring WebGates and Access Servers in the *Oracle Access Manager Access Administration Guide*.

Customization Recommendations

You can tailor Oracle Access Manager for your deployment and users. For example, you can create front-end customizations using IdentityXML, PresentationXML, and the Access Manager API. You may create back-end customizations with the Identity Event API, Authentication API, Authorization API (including custom AccessGates and plug-ins).

Oracle recommends that you create a /customizations directory to ensure that any customized information resides in a directory that is outside of any Oracle Access Manager component installation directory. This is important when re-installing or upgrading Oracle Access Manager because all sub directories are deleted during these processes.

Oracle recommends that you document any changes or customizations made within any deployment. Also, develop test scripts that verify the behavior of your customizations to help ensure that these work as expected. Scripts help simplify the task of redeploying the customizations to a larger deployment, or upgrading to a later Oracle Access Manager release.

Task overview: Creating and testing customizations and plug-ins

1. Review considerations here and in "Customizing the Look and Feel of Embeddable User Interface Elements" on page 1-9.

2. Install and setup Oracle Access Manager in a small test or development deployment (ideally a *sandbox*-type setting) where a dependency on the overall Oracle Access Manager deployment is minimal.

   For details, see the *Oracle Access Manager Installation Guide*.

3. Develop deterministic test scripts to run both before and after creating your customizations to exercise a full end-to-end transaction and ensure that everything works as expected.

   Your test scripts will depend on the specific customization being exercised. For example, your script could request a single page that requires authentication and authorization and a workflow request (all triggered by a single page request).

4. Compile the code or deploy the customization, and develop a set of instructions that explain how to configure the customization in a given deployment.

   For details about using the Oracle Access Manager Software Developer Kit and APIs, see the *Oracle Access Manager Developer Guide*.

5. Test any customization (styles, AccessGates, or plug-ins, for example) to ensure things are working as expected.

6. When the test is successful, redeploy any compiled binaries and customizations in a larger deployment for further testing before migrating this information to a production environment.

Testing and Performance Recommendations

Before deploying Oracle Access Manager into production, Oracle recommends that you run a thorough and extended load test and benchmark analysis. This enables you to fine tune and predict the behavior of the overall system. Based on the performance figures, you can tune performance, for example, by altering cache settings, timeout values, the number of directory connections, and increasing the number of threads on the Identity or Access Servers.

For performance tuning recommendations, see Chapter 3.

Identity System Recommendations

This section offers general recommendations for any Identity System deployment, including those in a joint Identity and Access System deployment:

- Customizing the Look and Feel of Embeddable User Interface Elements
- Recycling an Identity Server Instance Name

For capacity planning details, see Chapter 2. For specific performance tuning details, see Chapter 3.

Customizing the Look and Feel of Embeddable User Interface Elements

Oracle Access Manager Identity combines Extensible Style Language (XSL) stylesheets and Extensible Markup Language (XML) data to dynamically create almost all of the pages presented to its users. This capability, known as PresentationXML, provides developers with design flexibility and avoids the need for static HTML content.

PresentationXML is the recommended approach if your intent is to deal with front-end, user interface issues, for example, look and feel, layout of the tags, enhancing the navigation, and so on. It is *not* recommended for back-end logic, for example, pre-filling values based on data on a database, computing values based on other input values, communicating with external systems, and so on.

Oracle recommends:

- Before modifying or using a stylesheet, create a new style based on the default (style0) style of Oracle Access Manager. Replicate all related graphics, stylesheets and JavaScripts in the Identity Server and WebPass so that the default style remains unchanged.

  For details about creating a new style, see the *Oracle Access Manager Identity and Common Administration Guide*. For details about customizing Identity System pages, copying styles, testing, and propagating styles throughout the deployment, see the *Oracle Access Manager Customization Guide*.

- To expedite development and testing when developing PresentationXML, use a powerful XML or XSL editor, for example, XMLSpy. These editors provide an integrated development environment (IDE) to simplify and speed up the process of XSL programming.

  For more information about testing, and propagating styles throughout the deployment, see the *Oracle Access Manager Customization Guide*.

- Use caution when implementing Javascript code in PresentationXML

  When the need arises to insert JavaScript code to a front-end page through PresentationXML, the best practice is to encapsulate all of the JavaScript code into a file, then include the file in the XSL file. At deployment time, you must deploy this file on the appropriate WebPass installation directory.

  When including JavaScript code in PresentationXML, do not modify the main misc.js file in the WebPass installation directory. This file is used for client-side processing and is common to all Oracle Access Manager components. Any modification can adversely affect all components.

  For more information about customizing interfaces with PresentationXML, see the *Oracle Access Manager Customization Guide*.

Recycling an Identity Server Instance Name

If you must remove an Identity Server instance on one computer and reinstall it on another computer, you may re-use the original Identity Server instance name. However, this requires that you take specific steps to ensure that Oracle Access Manager recognizes the new instance and does not look for the original instance.

If you do not delete the Identity Server name from the System Console, a login following setup may result in the message "Application has not be set up". For more information about recycling an Identity Sever instance name after uninstalling the instance, see the *Oracle Access Manager Installation Guide*.

Access System Recommendations

This section discusses the following general recommendations for any deployment that includes the Access System:

- Using IP Validation, HTTPS, and Secure Cookies to Mitigate The Risk of a Cookie Reply Attack
- Configuring Dynamic Groups Rather than Authorization Filters to Simplify Authorization Administration

- Deploying WebGates On Reverse Proxies to Simplify Management
- Developing Document Protection Policies to Minimize WebGate Calls to the Access Server
- Configuring Form-Based Authentication to Avoid Login Errors

For capacity planning details, see Chapter 2. For specific performance tuning details, see Chapter 3.

Using IP Validation, HTTPS, and Secure Cookies to Mitigate The Risk of a Cookie Reply Attack

Oracle recommends that you always enable IP validation to mitigate the risk of a cookie reply attack. If exceptions are required, for example, when deploying using a reverse proxy topology, ensure that only allowed IP addresses are included in the exception list. Avoid ever turning off IP validation.

To avoid the risk of cookie reply attacks, you can also deploy content securely over HTTPS. This prevents unauthorized clients from eavesdropping on the ObSSOCookie. Also, specify ssoCookie for the Challenge Parameter for a Form, Basic, or External challenge method to ensure that the ObSSOCookie set during authentication is sent only through SSL. This prevents the ObSSOCookie from being sent back to a non-secure Web server. This parameter setting requires configuring all protected Web servers for SSL. An SSL Web server will not perform single sign-on with a non-SSL Web server. A browser will not return a secure cookie obtained from an SSL Web server to a non-SSL Web server in the same domain.

For more information, see the chapter on configuring user authentication in the *Oracle Access Manager Access Administration Guide*.

Configuring Dynamic Groups Rather than Authorization Filters to Simplify Authorization Administration

Generally, Oracle recommends using dynamic groups instead of authorization filters to specify authorization rules. Dynamic groups allow you to separate the management of the filter ("Who is a virtual member of the group?") from the management of the authorization rule. This enables you to delegate the management of the authorized role to a class of administrators that is separate from those who configure the access policies. Additionally, group management allows tracking of changes and approvals for changes, whereas authorization rule filters do not.

For more information, see the chapter on configuring user authentication in the Oracle Access Manager Access Administration Guide.

Deploying WebGates On Reverse Proxies to Simplify Management

There are a number of benefits to deploying WebGates on reverse proxies. These include:

- You can protect all Web content from a single logical component by directing all requests through the proxy.

  This is true even for platforms that are not supported by Oracle Access Manager. If you have different types of Web servers, for example, iPlanet, Apache, and so on, on different platforms, for example, MacOS, Solaris x86, mainframe and so on, all content on these servers can be protected. A reverse proxy can be a workaround for unsupported Web servers, eliminating the need to write custom AccessGates for unsupported Web servers or for platforms where there is no AccessGate support.

- You can install a WebGate on only the reverse proxy, rather than on every Web server.

  This creates a single management point. You can manage the security of all of the Web servers through the reverse proxy without establishing a footprint on the other Web servers.

- A reverse proxy provides architectural flexibility and can enable you to expose the same application on the intranet and the extranet without requiring any changes to the application already deployed.

The main pitfall of using a proxy is the extra work involved in setup. If you deploy the WebGate on a Web server that resides behind a reverse proxy, the following are required:

- Ensure that any Web server that uses the reverse proxy for authentication only accept requests from the reverse proxies.

  You must configure the WebGate deployed on this Web server to not enforce IP validation on requests coming from the reverse proxy server or servers acting as its front end. You must configure the IP addresses of the reverse proxy server or servers in the IP Validation list for the WebGate. Oracle does not recommend turning IP validation off for the WebGate because it can expose a security risk.

- Update the virtual hosts that are configured in the Policy Manager so that the Access System intercepts requests that are sent to the reverse proxy.

- Prevent people from circumventing the proxy by entering URLs that point directly to the back-end system. You can add Access Control List (ACL) statements in the server to prevent users from bypassing the reverse proxy and directly accessing restricted content. Or, you can configure firewall filters.

- Since the proxy processes all user requests, you must deploy enough proxy servers to enable the system to handle the load.

- Redirect all existing URLs to the host name and port number of the reverse proxy server.

  This often requires configuring the reverse proxy to inspect content and rewrite URLs, for example, to prevent any absolute HTML links from resulting in a broken link. This is available in most reverse proxies, and it is functionality that is independent of the Access System. It is a best practice that you configure URL links exposed to the front-ended applications to contain only relative URLs (`../../sub-path/resource`) rather than absolute URLs (`http://hostname.domain:port/path/resource`) or pseudo-relative URLs (that is, /path/resource). Absolute URLs can break links on the end user's browser when deployed behind a reverse proxy.

For more information, see the chapter on configuring WebGates and Access Servers in the *Oracle Access Manager Customization Guide*.

Developing Document Protection Policies to Minimize WebGate Calls to the Access Server

When specifying policies to protect all the documents on a Web server, there are two approaches that work:

- Protecting all the documents from the root of the document tree, and specifically allowing access to specified documents
- Setting the `DenyOnNotProtected` flag, and specifically allowing access to specified documents.

In general, the second approach provides better performance. When protecting Web documents from the root, the WebGate must always contact the Access Server for each request to check if the user is authorized to access the resource. This places additional load on the Access Server. When using the `DenyOnNotProtected` flag, the WebGate caches information from Access Server on whether a particular URL is protected by the Access System. As a result, it can simply deny access to subsequent requests for unprotected resources without contacting the Access Server thereby reducing server overhead.

For more information, see the chapter on configuring WebGates and Access Servers in the *Oracle Access Manager Access Administration Guide*.

Configuring Form-Based Authentication to Avoid Login Errors

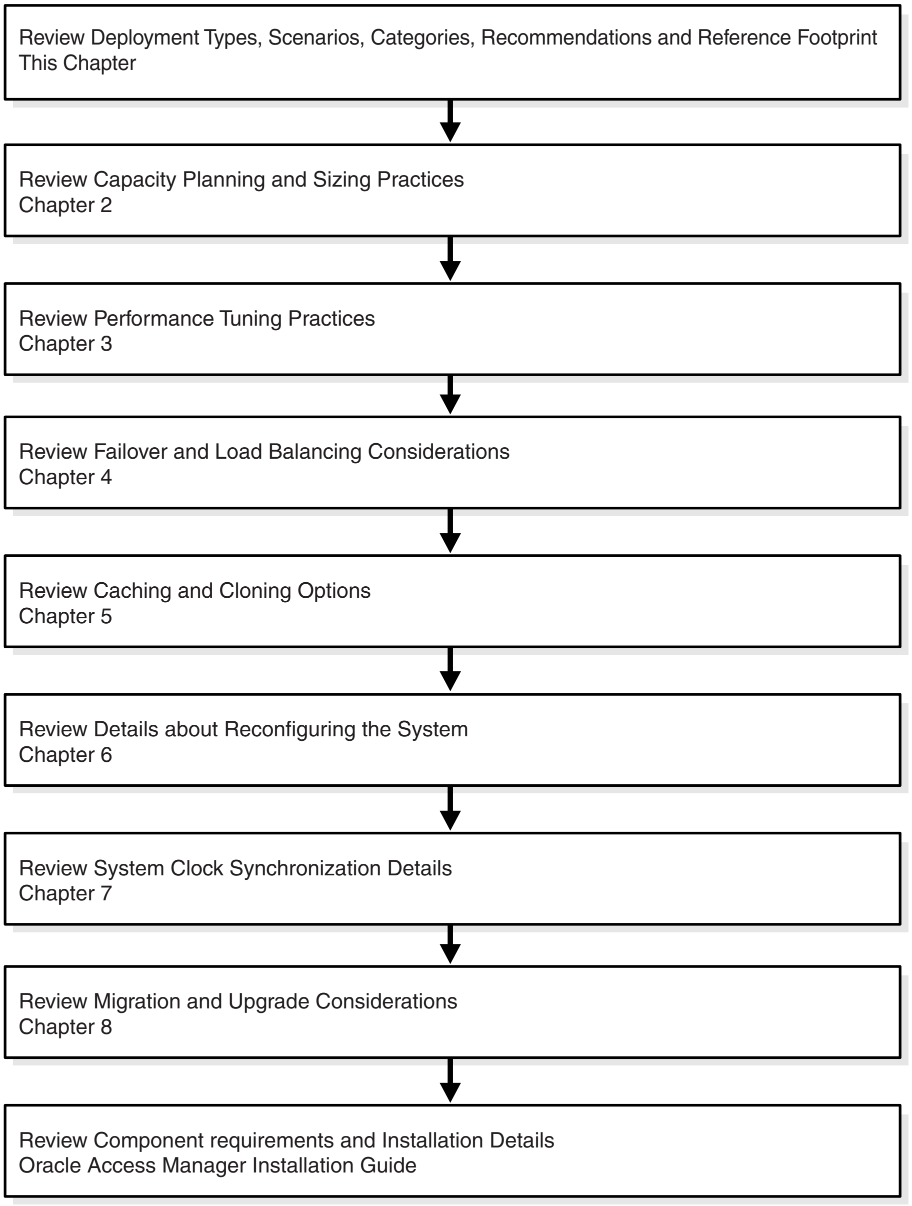
When implementing form-based authentication with Oracle Access Manager, develop code in such a way as to avoid login errors. This includes embedding code to validate input fields in the form to avoid posting the wrong credentials. For example, you can check that user name and password fields are not blank. In addition, use HTML code that prevents content caching, for example: `<meta http-equiv="pragma" content="no-cache">`.

For more information, see the chapter on form-based authentication in the *Oracle Access Manager Access Administration Guide*.

Oracle Access Manager Deployment Planning

Oracle strongly recommends that before starting any deployment task, you and your team become familiar with all topics suggested in Figure 1–1, and the overview that follows the figure.

Figure 1–1 Deployment Planning Overview


Review Deployment Types, Scenarios, Categories, Recommendations and Reference Footprint
This Chapter
Review Capacity Planning and Sizing Practices
Chapter 2
Review Performance Tuning Practices
Chapter 3
Review Failover and Load Balancing Considerations
Chapter 4
Review Caching and Cloning Options
Chapter 5
Review Details about Reconfiguring the System
Chapter 6
Review System Clock Synchronization Details
Chapter 7
Review Migration and Upgrade Considerations
Chapter 8
Review Component requirements and Installation Details
Oracle Access Manager Installation Guide

Task overview: Planning for your deployment

1. Review the information in this chapter to get a high-level overview and general recommendations about deployments. For more information, see the following topics:

   - About Oracle Access Manager Deployment Types and Tiers
   - Deployment Scenarios and Environments
   - Deployment Categories
   - General Recommendations
   - Identity System Recommendations
   - Access System Recommendations

- About Deployment Best Practices

2. Review "Planning Deliverables" on page 1-15 for details about the planning document you need to produce for each deployment.
3. Review Chapter 2 for the methods and strategies you use to determine capacity and sizing requirements for each deployment, for an Oracle Access Manager Reference Server Footprint, and for a Sample Medium-to-Large-Scale Deployment.
4. Review Chapter 3 for performance tuning recommendations, as well as tools and methods.
5. Review Chapter 4 for details about configuring failover and load balancing between Oracle Access Manager servers and Web components, as well as between Oracle Access Manager servers and directories.
6. Review Chapter 5 for details about cloned and synchronized components and caching system configuration information.
7. Review Chapter 6 for details about what can be reconfigured and how to do it.
8. Review Chapter 7 for details about synchronizing clocks across time zones.
9. Review Chapter 8 for details about migrating data and upgrading to a later Oracle Access Manager release.

Planning Deliverables

Planning activities include preparing a document where you define and record a detailed plan for each deployment.

Task overview: Developing your planning deliverables

1. **Decide Deployment Details**: Define a plan that identifies the following information for each deployment:
   - **Deployment Type and Tiers**: Decide and record the deployment type (Identity System only versus a joint Identity and Access System), as described in "About Oracle Access Manager Deployment Types and Tiers" on page 1-1.
   - **Deployment Scenario**: Decide and record each deployment scenario, as described in "Deployment Scenarios and Environments" on page 1-2.
   - **Deployment Category**: Decide if your deployment is to be an intranet versus extranet deployment, which have individual characteristics and dependencies as described in "Deployment Categories" on page 1-2.
   - **Deployment Size and Distribution of Components**: Decide and record the number and location of installed components, whether on one site or many. For capacity planning details, see Chapter 2.
   - **Standardization**: Each component should be running on stable and appropriately installed platforms and follow:

     General Recommendations
     Identity System Recommendations
     Access System Recommendations

   - **Administrative Access**: Schema and other operations require administrative access with write permissions to the directory and Oracle Access Manager files. Contact the individuals selected to be administrators for each deployment.

- **Customizations**: Customized configurations can be created to tailor Oracle Access Manager for your environment and users, as described in "Customization Recommendations" on page 1-8. Research the types of customizations that may be needed in your deployment.

2. **Create a Planning Document**: Record deployment decisions in a document using the details in Table 1–1 as a guide.

Table 1–1 General Deployment Details

Deployment Details	Description
Type	Identity System on versus Joint Identity and Access System
Scenario	Test or QA or Staging or Production or other
Category	Intranet versus Extranet
Administrators	Oracle Access Manager Master Administrator Master Identity Administrator Master Access Administrator Directory bind credentials used by Oracle Access Manager
Directory Profile	Transport Security: Open versus SSL-enabled Master/replica configuration Failover configuration For more information, see: ■ Chapter 2 for capacity planning and sizing guidelines ■ Chapter 4 for details about failover and load balancing
Oracle Access Manager Security	Transport Security Mode: Simple, Cert, or Open
Identity System	Identity Server instances WebPass instances Failover configuration For more information, see: ■ Chapter 2 for capacity planning and sizing guidelines ■ Chapter 4 for details about failover and load balancing
Access System	Policy Manager instances Access Server instances WebGate instances Failover configuration For more information, see: ■ Chapter 2 for capacity planning and sizing guidelines ■ Chapter 4 for details about failover and load balancing
3rd Party Integration Applications	See the *Oracle Access Manager Integration Guide* for implementation details for supported third-party applications.

Table 1–1 (Cont.) General Deployment Details

Deployment Details	Description
Customizations	Front-end customizations created using IdentityXML, PresentationXML, and the Access Manager API. Back-end customizations with the Identity Event API, Authentication API, Authorization API Custom AccessGates and plug-ins created using the Oracle Access Manager Software Developer Kit. For more information, see: ■ The *Oracle Access Manager Customization Guide*, which explains how to change the appearance of Oracle Access Manager applications and how to control operation by making changes to operating systems, Web servers, directory servers, directory content, or by connecting CGI files or JavaScripts to Oracle Access Manager screens ■ The *Oracle Access Manager Developer Guide*, which explains how to access Identity System functionality programmatically using IdentityXML and WSDL, how to create custom WebGates (known as AccessGates), and how to develop plug-ins.

3. **Fill in Installation Preparation Worksheets**: Review and record installation details in the preparation worksheets available in the *Oracle Access Manager Installation Guide*.

4. **Record Any Changes to the Deployment**: Be sure to keep a record of any changes within the deployment, including:

   - Patch set or bundle patch release numbers that are applied
   - Identity Servers (and Access Servers) configured for auditing to files or a database
   - Identify any Identity Event plug-ins
   - PresentationXML and XSL stylesheet customizations
   - File-based changes, for example to globalparams.xml or .lst files
   - Customized authentication or authorization plug-ins for the Access Server
   - Status of the Access Management flag
   - Details for each AccessGate, WebGate, and Policy Manager, such as the HTTP Cookie domain, preferred host name, cache timeout and size, failover threshold; custom IdentityXML clients; any virtual IP and DNS aliases used to reference the WebPass or Web server farm protected with WebGate

About Deployment Best Practices

Before starting your deployment, there are several broad guidelines to follow:

- **Think Strategically, Act Tactically**: Plan and design the solution from a strategic perspective, with a long-term view and a road map that is not necessarily tied to nor limited by tactical deadlines. Divide the plan into measurable and attainable phases, each of which contributes to the strategic goal while providing some return-on-investment (ROI) with every milestone.
- **Seek the Advise of Experts**: During planning, requirements gathering, and design, there is tremendous value in engaging experts. Consult with individuals who have years of industry experience and who have completed projects in

various vertical markets. The time and cost involved in engaging experts is offset by mitigating risks, adopting industry best practices, and defining a validated strategy for achieving success. Industry expertise is typically two fold:

  - **Industry Expertise**: Refers to knowledge of established policies, procedures, and standards; familiarity with compliance with governmental regulations and guidelines; information security best practices; security management and operations; technical infrastructure.
  - **Technical Expertise**: The best use of the technology in addressing business requirements; technical infrastructure; Oracle technology and application security; identity management; access management, Web services security, information assurance, and privacy management.

- **Invest in Knowledge**: Ensure that your team has the appropriate technology and product knowledge to support the deployment before you start any design. Oracle strongly recommends providing specific training to capture and transfer knowledge that enables individuals to independently support and maintain an evolving infrastructure. This investment is key to a successful deployment.

In addition to the recommendations throughout this guide, see the *Oracle Application Server Best Practices Guide*. It includes recommendations that may involve a combination of tools and manual processes to achieve a desired result that fall outside the scope of this manual.

The *Oracle Application Server Best Practices Guide* is updated on a quarterly basis. It focuses on the Oracle Identity and Access Management Suite, which covers the following technology areas:

- Oracle Access Manager: All recommendations specific to Oracle Access Manager are repeated in this guide.
- Oracle Internet Directory
- Oracle Virtual Directory
- Security
- Oracle Application Server High Availability

2
Capacity Planning

Capacity planning is the process of determining which server hardware best supports an Oracle Access Manager deployment based on anticipated usage. The information in this chapter provides a basis for capacity planning that *helps* ensure that the server hardware in an Oracle Access Manager deployment is adequate for handling peak loads. This chapter includes the following topics:

For a general overview of Oracle Access Manager deployments, see Chapter 1.

About Capacity Planning

The goal of capacity planning for an Oracle Access Manager deployment is to maintain an acceptable level of system performance while taking into account a number of different factors:

- The time frame in which the users are expected to interact with the system
- The user population that is expected to access the system in the given time frame
- The average duration of the user session, the duration of the set of transactions
- The average number of pages that make up a user session

To access the appropriate hardware for your environment, it is important to understand Oracle Access Manager sizing and scaleability characteristics. Oracle Access Manager components perform well on standard hardware. However, it is more cost effective to have additional capacity than to try to make do with inadequate hardware. The cost of hardware is low compared to the effort required to maintain an under-powered system.

Capacity planning includes:

- Estimating the expected system load

- Making decisions based on the peak load while factoring in performance and scaleability characteristics

The methods and procedures in this chapter can be applied to help you determine the capacity and sizing needs of the deployment, whether you have a single deployment on a single computer or multiple deployments spanning different sites.

Note: This chapter provides guidelines and examples intended for demonstration purposes only. This chapter may not be specific to the hardware on which you are using the software and does not always provide actual data. Please be advised that Oracle Corporation is not be responsible for any loss, costs, or damages incurred due to the use of the information in this chapter.

Estimating the Anticipated Peak System Load for Server Sizing

Appropriate server sizing should ensure that your server hardware can handle the maximum number of operations that can be expected in a particular time interval. Put another way, the server hardware in your Oracle Access Manager deployment should accommodate all users during times of peak load.

Information about the peak load for a given time interval can usually be obtained from:

- Measurements from live systems in use or historical data
- Calculations and projections

Oracle Access Manager is a stateless system. Therefore, the estimated maximum transaction throughput and network traffic are critical factors in capacity planning.

This section includes the following topics:

- Measuring the Load
- Projecting System Usage

Measuring the Load

This discussion describes two methods that you can use to measure the load during peak hours in an Oracle Access Manager deployment. From this information, you can estimate your overall system-capacity requirements.

You can compare your load estimates (transactions-per-user-per-second) with your equipment manufacturer's specifications for server hardware. Based on these comparisons, you can determine if the computers you already have are adequate for supporting the estimated load. If existing computers are not adequate, you can base your equipment choices in part on your own throughput requirements.

There are numerous network traffic and Web site usage monitoring tools available for use with the methods described here. However, use of third-party tools is outside the scope of this book.

This discussion includes the following topics:

- Measuring the Load in a Deployment
- Measuring the Active User Sessions in a Multi-Site Deployment

Note: Even these methods of estimation may be more rigorous than is required for a deployment of fewer than 20,000 users. Standard server class hardware is adequate for most deployments, as discussed in "Sample Medium-to-Large-Scale Deployment" on page 2-17.

Measuring the Load in a Deployment

Measuring the load includes establishing the highest number of pages and requests per second over a given time interval. This provides you with a good idea about your overall system-capacity requirements.

While you can measure usage over as little as a 24-hour period, Oracle recommends that you measure usage over a period of several weeks. If usage tends to spike during particular weeks of the year, try to obtain measurements from the busiest weeks. From this, you can better extract system-capacity requirements that hold true even in the busiest period.

To estimate a typical busy load, you multiply the value of an average heavy load by a small integer such as 2 or 3. This allows for usage patterns that are two or three standard deviations higher than an average heavy load, assuming a Gaussian distribution (bell curve) of loads.

To base your estimate on the peak load for the deployment

1. Measure usage over a significant time period to obtain measurements from the busiest period.
2. Choose the highest value seen in a production deployment to use during the next step.
3. Estimate the parameters of a typical busy load by multiplying the value of an average heavy load by a small integer such as 2 or 3.

Another method that you can use is to measure the active user sessions in a multi-site deployment, as described next.

Measuring the Active User Sessions in a Multi-Site Deployment

If you have a multi-site deployment, Oracle recommends that you create a chart of peak usage for all sites, and then estimate your peak load based on the total estimated usage across all sites. One way to do this is to record the number of logged-in users at each site during different times of the day.

Note: To ensure accuracy using this method, the actual user request rate during peak hours should come from either monitoring the live system in use, or from historical data.

A table such as Table 2–1 allows you to estimate the times when the majority of the *users* on each site are busiest (the shaded area). Each column reflects an hour of the day (local time) that is recorded based on Greenwich Mean Time (GMT). Each row represents the number of logged-in users that were monitored at that hour. According to the example, usage in Mexico City typically starts at GMT 12 and continues through GMT 24. The peak in Mexico City occurs between GMT 16 through 19.

Note: Mexico City, Mexico is 6 hours behind GMT. Therefore, when it is 6:00:00 PM on Tuesday, February 6, 2007 in Mexico City, GMT is 00:00:00 on Wednesday, February 7, 2007. Greenwich Mean Time (GMT) or World Time is also known as Universal Time Coordinated (UTC). GMT, World, and UTC time reflects the mean solar time along the Earth's prime meridian. The prime meridian is arbitrarily based on the meridian that runs through the Greenwich Observatory outside of London, England, where the present system originated. UTC is also known as Coordinated Universal Time and sometimes as Universal Coordinated Time; all are abbreviated as UTC and refer to the standard time common to every place in the world (formerly and still widely referred to as Greenwich Mean Time or World Time.

Table 2–1 Peak Load Based on Estimated Usage Across Sites

Peak hours GMT	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Mexico	0	0	0	3	75	150	175	225	225	225	225	225	225	225	225	175
Spain	35	50	75	50	25	35	50	75	75	75	75	35	20	0	0	0
Egypt	45	45	50	50	50	50	50	55	55	45	30	10	0	0	0	0
U.S.	0	2	5	10	30	80	95	95	95	95	86	80	80	90	75	60
Columbia	0	2	7	15	30	45	45	50	50	50	55	55	55	55	45	35
Costa Rica	0	0	0	0	2	5	10	10	10	12	12	12	12	12	12	12
Indonesia	60	45	30	10	4	2	0	0	0	0	0	0	0	0	3	5
Taiwan	125	115	100	60	30	5	0	0	0	0	0	0	0	10	25	75
Total	265	269	267	198	372	425	425	510	510	502	483	417	392	392	385	362

When estimating the peak load based on the number of logged-in users, Oracle recommends that you assume:

- At peak hours all logged-in users are active
- Each active user makes an average of 4 requests per minute

Based on the maximum active users (510 users in Table 2–1), you estimate the peak number of requests per second as follows:

Maximum users * Requests per minute = Total requests per minute
/ 60 seconds = Total requests per second

For example:

510 users * 4 requests per minute = 2040 requests per minute
/ 60 seconds = 34 requests per second

To factor in a standard deviation that is higher than an average heavy load, assuming a Gaussian distribution (bell curve) of loads, you can multiply the estimated requests per second by a small integer: 2 or 3, for example. In this case, you have an estimated peak usage value of:

Requests per second * Standard deviation = Estimated peak requests per second

For example:

34 requests per second * a standard deviation of 2 = 68 requests per second

34 requests per second * a standard deviation of 3 = 102 requests per second

To estimate the peak load based on the number of logged-in users

1. Create a table that includes all sites in your deployment using Table 2–1 as a guide.
2. Determine the number of logged-in users for each hour of the day based on either monitoring each site or historical data.
3. Multiply the maximum number of users by the estimated number of requests per minute (for example, 4 unless you have more accurate data) to determine the total requests per minute.
4. Divide the total requests per minute by 60 to establish the number of requests per second.
5. Multiply the requests per second by a small integer (2 or 3) to complete your estimate for a higher than average heavy load.

Projecting System Usage

When there is no simple way to measure usage, or when there is no historical data available, you can use the method described here to project the system usage in an Oracle Access Manager deployment.

Projected Number of Users: Obtain data on the number of users per office from the Human Resources department, or some other authoritative source in your enterprise. From this you can estimate the total number of users accessing resources during peak hours. Be sure to include full-time employees, part-time employees, and contractors in your estimate.

Projected Time Intervals in a Geographically Distributed Deployment: Rather than trying to gather statistics on who is logged in at each site, it may be more practical to identify peak usage time intervals in a geographically distributed deployment and estimate the number of users who are active during those intervals.

For instance, suppose your company has offices in several countries worldwide. You can create a chart of regular office hours plotted against Greenwich Mean Time (GMT). A table such as Table 2–2 allows you to estimate the times when the majority of the offices are busiest (the shaded area), which is a good indication of peak load hours. In Table 2–2, the number 1 within each row indicates 12:00 00AM local time as well as its relationship to GMT. For example, with daylight savings time in effect 12:00AM in Mexico City occurs at 07:00 AM GMT.

Table 2–2 Chart of Office Hours Plotted Against GMT with Daylight Savings Time

GMT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Mexico	19	20	21	22	23	24	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Spain	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1
Egypt	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2
U.S.	20	21	22	23	24	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Columbia	20	21	22	23	24	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Costa Rica	19	20	21	22	23	24	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Indonesia	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7	8
Taiwan	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7	8

A table such as Table 2–3 provides user community statistics per office, from which you can estimate the total number of users accessing resources protected by Oracle Access Manager during peak hours.

Table 2–3 Employee and Contractor Data from an Authoritative Source in Your Company

Country	Full time	Part time	Contract	Total
Mexico	3021	496	35	3552
Spain	755	356	5	1116
Egypt	329	275	0	604
U.S.	134	25	55	214
Columbia	1290	245	11	1546
Costa Rica	175	130	0	305
Indonesia	250	97	18	365
Taiwan	286	88	44	418

You can create a simple throughput projection based on:

- An estimate of the total maximum number of users that you expect to be logged in at the same time
- The estimated number of transactions per user for a given time period
- The estimated transactions-per-second that need to be supported

You can compare your transactions-per-second estimates with claims made by your hardware vendors.

For example, using information in the Table 2–2 and Table 2–3 as a sample, you can project the total users and peak load as follows:

- **Total Maximum Users**: Add together the total number of users on each site. Using the data in Table 2–3 as a sample: 3552+1116+604+214+1546+305+365+418=8120 total users.
- **Peak Load**: This calculation is similar to the one described in "Measuring the Active User Sessions in a Multi-Site Deployment" on page 2-3 where you multiply the total users by the estimated requests per minute (4 is a good number unless you have better data) then divide the requests per minute by 60 seconds to establish the requests per second.

  > Total maximum users * Requests per minute / 60 seconds = Total requests per second

  Using the data in Table 2–3 as a sample:, you can project:

  > 8120 total users * 4 requests per minute = 32,480 / 60 seconds = 541 requests per second

Based on your human resources data, this is the maximum possible load. Therefore, you do *not* need to multiply the peak load to take account into the Gaussian distribution of loads. However, you may want consider some safety factors that project the future growth in the system usage.

To project system usage

1. Create a chart of hours plotted against GMT for your deployment, using the sample Table 2–2 as a guide.

   a. List each GMT hour in a separate column across the top of your table.

   b. Create a row for each office site in your deployment in the left column of the table.

   c. For each site, add a number from 1-24 to indicate local time as it corresponds to the site's offset from GMT using Table 2–2 as a guide.

   d. Determine the hours of peak usage for each site in the table and highlight those.

2. Create a table template for employee and contractor data for each site using the sample Table 2–3 as a guide.

   a. Obtain the number of users per site from your Human Resources department or another authoritative source in your enterprise.

   b. For each site in your table, enter the number of:

      Full time employees
      Part-time employees
      Contractors

   c. Calculate and enter the total number of users for each site.

3. Add the total for each site to estimate of the maximum number of users that could be logged in at the same time.

4. Estimate of the maximum load using the calculation below:

   Maximum total users * Requests per minute / 60 seconds = Total requests per second

5. Consider adding in some safety factors to project future growth in system usage.

Component-Specific Capacity Planning and Sizing Considerations

The "Oracle Access Manager Deployment Overview" in Chapter 1 provided several general recommendations for Oracle Access Manager installations. In addition, the following component-specific considerations should be taken into account:

- Identity and Access Server Recommendations
- WebPass Considerations and Recommendations
- Access System Considerations and Recommendations

For performance-tuning guidelines, see Chapter 3.

Identity and Access Server Recommendations

As discussed in Chapter 1, Oracle recommends installing only one Identity or Access server instance per dedicated computer because 85% utilization is considered a fairly standard load. If you anticipate that the combined utilization for a pair of primary Identity and Access Servers is below 85%, you could consider the following option to help reduce the required hardware estimate:

- One host is installed with a primary Identity Server and a secondary Access Server

- A different host is installed with both a primary Access Server and a secondary Identity Server

The primary server handles the load unless there is a failover situation. During failover, the secondary server handles the load. If the secondary server becomes activate the utilization of both servers should add up within 85%. If the combined load is projected to be much greater, Oracle recommends that you install the secondary Identity or Access Server on a different computer.

WebPass Considerations and Recommendations

See Chapter 1, "Web Server Recommendations" on page 1-6.

Access System Considerations and Recommendations

In addition to general Access System recommendations in Chapter 1, Oracle recommends that you take items described in the following discussions into account for Access System capacity planning:

- Access Server Recommendations
- Access Server to WebGate Ratios
- WebGate Impact on Web Server Performance

Access Server Recommendations

Depending on the load, Access Servers can reach both high CPU and memory utilization. This is critical when planning for the number of Access Servers in your deployment.

System configuration and deployment scenarios also affect performance significantly and are critical factors in capacity planning. For example, the following items impact your capacity planning and sizing calculations, as well as your performance tuning decisions. Whether you have the following functions configured to be on versus off depends on the requirements of your enterprise:

- The transport security mode between each tier (whether open or SSL-enabled)
- Auditing on
- User and group caches
- The policy cache
- User credential caching
- Password policy

The *Oracle Access Manager Installation Guide* describes transport security modes. For performance tuning recommendations, see Chapter 3. For more information about the features above, see *Oracle Access Manager Identity and Common Administration Guide* and *Oracle Access Manager Access Administration Guide*.

Access Server to WebGate Ratios

A commonly used WebGate to Access Server ratio is 10:1. A production deployment may require a higher ratio than a development or staging deployment. The actual ratio in your deployment depends on the actual peak load. By carefully monitoring the situation, you may scale up to higher ratios.

WebGate Impact on Web Server Performance

A WebGate may be installed against an existing Web server instance or a new Web server instance.

The number Web server/WebGate pairs depends on usage patterns within the deployment. There is some degradation to baseline Web server performance (about 10% to 20%) when hosting a WebGate. Based on this estimate, Oracle recommends that you add one additional Web server/WebGate pair for every five Web server/WebGate pairs in the deployment. Put another way, to provide the same Web server performance, you should add a sixth Web server/WebGate pair for every five Web server/WebGate pairs.

For details about Web server requirements, see the *Oracle Access Manager Installation Guide*.

Oracle Access Manager Performance and Scaleability Characteristics

The following topics provide an overview of Oracle Access Manager performance and scaleability characteristics based on tests conducted by Oracle:

- Scale-Up Characteristics
- Scale-Out Characteristics
- Deployment and Configuration Impact on Performance
- Baseline Performance for Identity and Access Servers

Scale-Up Characteristics

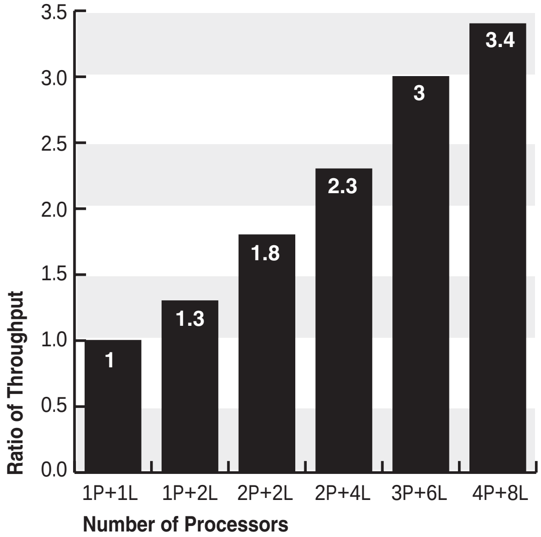
Scaling up refers to adding additional processors to your deployment to increase capacity. The results of this test show a fairly nice linear scale up of throughput versus the number of processors.

Figure 2–1 shows the Oracle Access Manager scale-up characteristics in a 4-CPU Wintel[1] system. Oracle Access Manager benefits from Hyper-Threading (simultaneous multithreading on the Pentium 4 microarchitecture). With Hyper-Threading enabled, a Pentium 4 processor is treated by the operating system as two processors instead of one.[2] In Figure 2–1, you can plot the throughput ratio of a scale-up test in a 4 CPU system.

[1] Wintel is an industry term for personal computers that are based on the Intel microprocessor with one of the Microsoft Windows operating systems.

[2] Intel's trademark for their implementation of this technology is officially called Hyper-Threading Technology (HTT).

Figure 2–1 Oracle Access Manager Scale Up (4 x 2.2 Ghz CPU)



In the bar chart shown in Figure 2–1, throughput-ratio data values are shown along the vertical y-axis. The horizontal x-axis provides the following category data:

- P represents the number of processors.
- L represents the number of logical processors for the Hyper-Threading equipped processors.

  For the same Hyper-Threading equipped processors, with Hyper-Threading turned on the system shows 2 logical processors; if Hyper-Threading is turned off, the system shows 1 logical processor.
- 1P+1L and 2P+2L data show the performance with Hyper-Threading turned off.
- 1P+2L, 2P+4L, 3P+6L, and 4P+8L data show the performance with Hyper-Threading turned on.

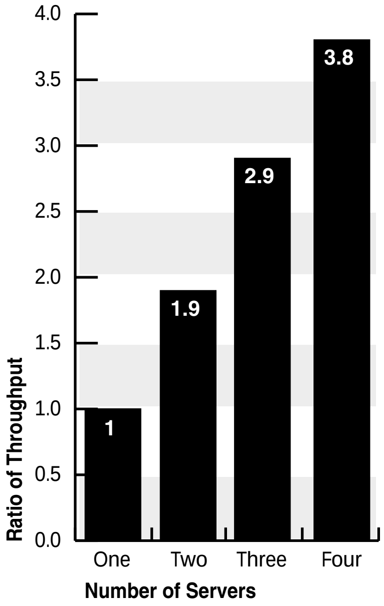
The throughput ratio of from 1 to 3.4 is calculated by dividing the throughput of each test by the throughput of the 1P+1L results. Therefore, the 1P+1L data is 1.

Scale-Out Characteristics

Scaling out, also known as scaling out horizontally, refers to adding more physical servers to increase the capacity in your deployment.

Figure 2–2 shows Oracle Access Manager scale-out characteristics based on a series of tests that run with from one to four identical Wintel servers. In this series of tests each Oracle Access Manager server was driven to 100% CPU, and then the throughput was measured. You can see an almost straight linear increase ion throughput as the number of servers increase.

Figure 2–2 Oracle Access Manager Scale Out



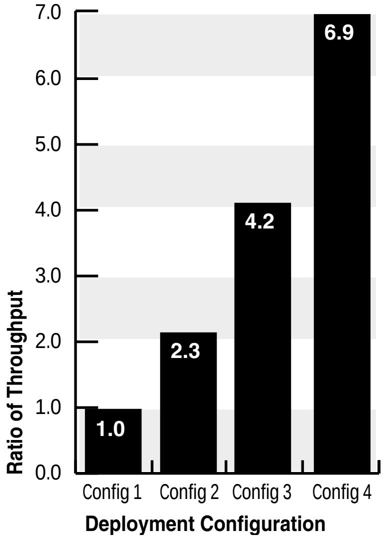
In the bar chart shown in Figure 2–2, throughput-ratio data values are shown along the vertical y-axis. The horizontal x-axis provides category data for individual configurations that include either 1 or 2 or 3 or 4 identical Wintel servers hosting Oracle Access Manager. Data for a single server was used as the basis for the comparison and given a ratio of 1.0.

The results of this test show how Oracle Access Manager can be scaled out horizontally.

Deployment and Configuration Impact on Performance

Figure 2–3 shows the impact of various deployment configurations on Oracle Access Manager performance. In this test, the throughput ratio is calculated against Config 1 both because this is the heaviest load option and because this is the most typical deployment.

Figure 2–3 Configuration Impact on Performance



In the bar chart shown in Figure 2–3, throughput-ratio data values are shown along the vertical y-axis. The horizontal x-axis provides category data for individual deployment configurations.

Table 2–4 provides details about each of the four deployment configurations that were used in the throughput ratio tests depicted in Figure 2–3.

Table 2–4 Configuration Settings for Performance Tests

Config #	Transport Security: Web Components to Identity and Access Servers	Identity Server: Password and Lost Password Policy	Auditing of Identity and Access Servers	User Login	User Credential Caching on Access Server
Config 1	Simple or Third-Party Certificate	Enabled	On	Form-based	On
Config 2	Simple or Third-Party Certificate	Enabled	Off	Basic-Over-LDAP	On
Config 3	Simple or Third-Party Certificate	Disabled	Off	Basic-Over-LDAP	On
Config 4	Open	Disabled	Off	Basic-Over-LDAP	On

Baseline Performance for Identity and Access Servers

This discussion illustrates the results of a baseline performance test. You can use the baseline ratios here in combination with details in "Scale-Up Characteristics" on page 2-9 and "Scale-Out Characteristics" on page 2-10 to estimate possible server (or server cluster) throughput and capacity.

For this test, 1 million users were loaded into a Sun Java System directory server v 5.2 in a joint Oracle Access Manager Identity and Access System deployment. During this test there were 100,000 active user sessions on the Identity and Access Servers. This deployment configuration includes:

- Either Simple or third party certificate to secure communication between Web server components and Identity and Access Servers.
- On the Identity Server, password and lost password policy are configured and enabled as follows:

  - Password policy requires uppercase, number, and lowercase with account lock out after three wrong tries.
  - Lost password challenge is set, password histories are stored for three tries.
- Auditing to file is turned on for the Identity and Access Servers.
- Form-based login is configured; required credentials are Username and password.
- User credential caching in the Access Server is turned on.

Baseline throughput numbers were obtained from the test setup described above. The results of this test are shown in Table 2–5. The numbers are normalized, and equivalent to the 1P+2L server data shown in Figure 2–1 on page 2-10.

Table 2–5 Sample Throughput for Config 1

Operation	Throughput (Request-per-second-per-CPU)	Comments
Self Registration	32	Users perform only self registration
Change Password	86	Users perform only change password
Lost Password	69	Users perform only recover lost password
Login	116	Users perform only login
LoginNavi	366	Users perform login and page navigation
AllMix	341	Users perform all above actions in mixed ratio: ■ 238 total virtual users (stimulated through Mercury LoadRunner) ■ 10% (~24) perform change password ■ ~80% (195) perform LoginNavi scenario ■ ~4 virtual users performed locked out scenario ■ ~5% (12) perform the lost password scenario ■ 3 perform self registration with no approval

As shown in Table 2–5, the LoginNavi operation provides the greatest throughput ratio for requests-per-second-per-CPU; the AllMix operation provides the second greatest throughput ratio. The joint Identity and Access System deployment that produced the test results in Table 2–5 includes Identity and Access Servers, each installed on a single dedicated computer. For more information about the baseline deployment and test cases, see "Sample Medium-to-Large-Scale Deployment" on page 2-17 and "Test Cases for Baseline Performance Data" on page 2-19.

You can use the sample information in Table 2–5 along with other information in this chapter to determine the load and sizing for the Access Server, and then to extrapolate load and sizing details for the Identity Server.

The sample calculations here are derived from the AllMix throughput data in Table 2–5. When you have a joint Identity and Access System deployed, using the AllMix operation for calculations is fairly realistic. You can also use the LoginNavi throughput for your calculations. For example, based on an AllMix 1P+2L baseline of 341, a scale-up ratio for 1P+2L =1.3 and scale-up ratio for 2P+4L = 2.3 as shown in Figure 2–1 on page 2-10, you find that:

- 2-CPU servers can handle 603 requests-per-second

  341:1.3 = throughput:2.3; throughput=341*2.3/1.3 = 603

- 4-CPU servers should be able to handle 892 requests-per-second (341*3.4/1.3)
- 2-server clusters with 2-CPUs in each box should be able to handle 1146 requests-per-second (603*1.9)

If your estimated load is 800 requests-per-second, you could choose one of the following hardware configurations for your Access System deployment:

- Either a 4-CPU server
- Or a 2-sever cluster with 2-CPUs on each server

After sizing the Access Server, you can establish preliminary Identity Server sizing. For Identity Server sizing, you use as a starting point a number that represents *half* the Access Server firing power. You then refine your calibrations using the estimated transaction throughput data shown in Table 2–5 for each of the following:

- Self Registration
- Change Password
- Lost Password

Using the example of 800 requests-per-second for the Access Server as a basis (with a 2-server cluster deployment for the Access Server), half of this capacity for the Identity Server results in one server with 2-CPUs. However, you must again use the Self Registration, Change Password, and Lost Password details in Table 2–5 to determine whether one Identity Server with 2 CPUs is adequate.

The following procedure provides an example of how you can use the information in in Table 2–5 and the rest of this chapter to determine the load and sizing for the Access Server and the Identity Server.

To determine the load and sizing for the Access Server and the Identity Server

1. Start with the maximum baseline of requests-per-second-per-CPU using details in Table 2–5 as a guide.
2. Determine the scale-up ratio using data in Figure 2–1 on page 2-10, and ratio for performance versus configuration details in Figure 2–3 on page 2-12.
3. Use a number that represents half of the required Access Server power as a starting point for Identity Server sizing.
4. Use the Identity System transaction data (Self Registration, Change Password, and Lost Password) in Table 2–5 to determine whether one Identity Server with 2 CPUs is adequate.

Oracle Access Manager Reference Server Footprint

This discussion provides details based on use cases created by Oracle to establish a point of reference for Oracle Access Manager components in live deployments of varying size.

A small scale deployment can be estimated as up to 20,000 users. A small-to-medium-sized deployment typically includes up to 100,000 users. Large scale deployments include from 100,000 to 2,000,000 users.

Oracle Access Manager components perform well on standard hardware. Based on tests with various configurations (Config4 shows the highest memory consumption), you may find that:

- 100,000 active user sessions may consume from 600 MB to 1 GB of memory on an Access Server

- With 32-bit systems, the Operating System can usually support up to 2 GB of memory per process

The following topics provide more information about the hardware for Oracle Access Manager deployment sizes:

- Hardware for Small-to-Medium Deployments
- Hardware for Large Deployments

> **See Also:** "Scale-Up Characteristics" on page 2-9 and "Scale-Out Characteristics" on page 2-10

Hardware for Small-to-Medium Deployments

For small to medium-sized deployments with between 20,000 and 100,000 users, any supported server-class computer and operating system should be adequate for the Identity and Access Server. The following items should be taken into account:

- Failover requirements double the number of computers needed. For example, expect to use a minimum of two Identity Servers and two Access Servers for redundancy. When the load is light, these servers may be deployed in a cross-over configuration to optimize hardware utilization. For more information, see "Identity and Access Server Recommendations" on page 2-7.
- Sever Sizing
  - A minimum of 2 GB of memory is needed for each Identity and Access System process.
  - A minimum of a 2 x 2 GHz CPU system with 4 to 6 GB of memory for each server.
- A single dedicated Web server is required for the System Console. This may be deployed on either of the two main Access and Identity Servers to eliminate additional hardware requirements.

For more information about Oracle Access Manager requirements, Web server requirements, and how to prepare for and install components, see the *Oracle Access Manager Installation Guide*.

Hardware for Large Deployments

For deployments of 100,000-2,000,000 users, Oracle recommends the following hardware for Identity and Access Servers:

- Each server should be a 4 x 2 GHz CPU system, or a 2 x 2 GHz system, with 6-8 GB of memory and 2 x 17 GB of mirrored storage. A cluster setup is preferred.
- The Identity Server should be deployed in a Delegated Administration model as described in the *Oracle Access Manager Identity and Common Administration Guide*

For a multi-million user deployment, Oracle recommends that you scale-out with multi-servers using guidelines"Scale-Out Characteristics" on page 2-10. Use "Baseline Performance for Identity and Access Servers" on page 2-12 to establish your sizing requirements.

For more information about Oracle Access Manager requirements, Web server requirements, and how to prepare for and install components, see the *Oracle Access Manager Installation Guide*.

Considerations for the LDAP Directory Server

In addition to the Lightweight Directory Access Protocol (LDAP) directory server considerations discussed in Chapter 1, Oracle recommends that you consider using multiple LDAP directory servers to balance the load if there are significant update operations that involve the Identity Server (or Identity and Access Servers in a joint deployment). This is especially true for password policy-related operations. In such cases, a multi-master and replica for the LDAP server allows you to configure load balancing between the Identity Server and the LDAP replica (as well as between the Access Server and LDAP replica in a joint Identity and Access System deployment).

When authenticating against the LDAP directory, consider enabling Access Server's user credential caching. This can significantly lower the load on the LDAP directory server during authentication, and improve throughput. For more information, see "Configuring Password Validation by the Access Server" on page 3-33.

Disk Space Sizing: A general rule of thumb is to multiply your LDIF file size by 5 to establish the actual LDAP disk space needed for entries and the index. During LDAP initialization, the host computer needs at least the same amount temporary disk space as the LDIF file size multiplied by 5. Each backup copy of your LDAP content requires the same amount of disk space as that of the live LDAP server. For example, an LDIF file for 1 million users is approximately 2 GB in size. When loaded, a 2 GB LDIF file expands to approximately 10 GB. When you take into account the temporary space during initialization, and space required for three full backup copies, you need at least 2 x 5 x 5 = 50 GB disk space:

LDIF size * 5 = Expanded LDIF size
Total LDIF * 2 = LDIF and Temporary space
Total LDIF * 3 = Backup space
Expanded LDIF * 2 + Backup space = Total Disk Space

For example:
2 GB * 5 = 10 GB * 2 = 20 GB + 30 = 50 GB

Memory Sizing: This depends on the LDAP server and the deployment configuration you choose. Oracle recommends that you review the LDAP documentation for your specific directory server and Operating System.

For 32-bit systems, the Operating System may impose a 2 GB limitation per process. For large-scale deployment, a 64-bit system and Operating System are preferred. For more information, see:

- LDAP Server Requirements For Small to Medium Deployments
- LDAP Server Requirements For Large Deployments

> **Note:** For LDAP directory performance tuning recommendations, see Chapter 2.

For performance tuning recommendations, see Chapter 3.

LDAP Server Requirements For Small to Medium Deployments

Most currently available directory servers with 2 GB of memory may be adequate for small to medium sized deployments of up to 100,000 users. In addition, Oracle recommends that you:

- Use disks that are RAID 01 (striped and mirrored) for the directory server.

- Use two replicated directory servers for failover.
- Have enough memory to hold two times the LDAP data in memory.
- Have an LDAP cache that is large enough to hold the entire database.
- Have available 5 GB of disk storage per directory server

LDAP Server Requirements For Large Deployments

For deployments of 100,000 and more users Oracle makes the following recommendations:

- Allow 100 GB of disk storage for 2 million users
- Segregate physical drives for directory server access and error logs from the attribute data (indexed or not) volume drives

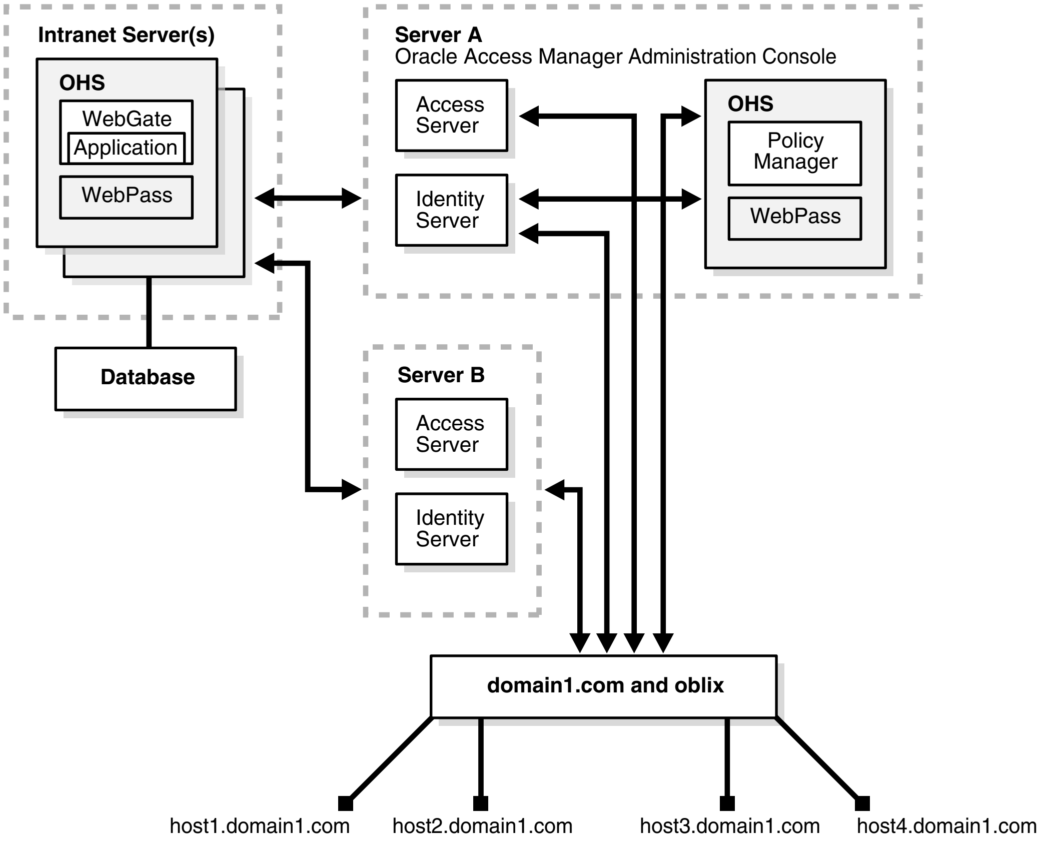
  For example, Oracle Internet Directory I/O Subsystem Requirements include arrays of disk drives controlled by disk controllers. It is important to consider performance requirements when you size the I/O subsystem rather than using sizing estimates based only on storage requirements. For more information, see the *Oracle Application Server Best Practices Guide*.
- Allow CPUs 2 or 4 x 2 GHz: If Access Server user credential caching is configured and on, you can significantly lower the LDAP server load and boost performance.
- Allocate at least 2 GB of RAM for a 100,000 user directory; at least 4 GB for a very large directory.

Sample Medium-to-Large-Scale Deployment

Figure 2–4 illustrates hardware and software choices in a lab test environment for a medium to large scale deployment that includes 1 million users in the LDAP directory and 100,000 active user sessions on the Access Server. The baseline throughput numbers were obtained from this deployment and used in"Baseline Performance for Identity and Access Servers" on page 2-12.

This deployment is depicted in Figure 2–4 and is installed as described in Table 2–6.

Figure 2–4 Sample Deployment



This deployment includes Intel Xenon server-class computers with 2 CPUs, each with 2.8 GHz and 28 SPECInt_rate2000, as shown in Table 2–6.

Table 2–6 Server Installation Details

Number of Servers	Installed With
5	Oracle HTTP Server/WebPass/WebGate
2	Access & Identity Server
2	LDAP Server
2	LR Test Load Drivers

For the baseline test deployment quoted in this chapter:

- The servers (Sun Java System Directory Server, Oracle HTTP Server and Oracle Access Manager), are running on a Windows 2003 Server (SP1).
- Each server has 2x2.8 GHz CPUs and 6 GB of RAM, with 140 GB of disk space.
- Access Server and Identity Servers are co-located and configured as a primary-secondary setup.
- One LDAP directory server is used
- WebPass and WebGate are installed on each of five Oracle HTTP Servers
- The Policy Manager is installed on one of the Oracle HTTP Server systems

You can scale this deployment as follows:

- Increase the number of Oracle Access Manager server hosts to spread the load and offer higher redundancy.
- Increase capacity by increasing the amount of CPUs, memory, and disk space to improve server performance
- The LDAP directory server supports the addition of new servers for cases where this is desired to improve performance and reliability.

For more information on scaling a deployment, see "Scale-Up Characteristics" on page 2-9 and "Scale-Out Characteristics" on page 2-10.

Test Cases for Baseline Performance Data

The baseline performance data in this chapter is obtained from tests run with the Oracle Access Manager Benchmark Suite Test Application, which contains 100 static HTML pages, each of 14k in size. Data for 1 million users resides in the deployment's LDAP directory server.

The following topics describe scenarios for the Oracle Access Manager Benchmark Suite for Access Server, Identity Server, and an integrated test case for both servers:

- Identity Server Baseline Performance Test Case
- Access Server Baseline Performance Test Case
- Integrated Baseline Performance Test Case

Identity Server Baseline Performance Test Case

The following topics describe baseline performance test cases for the Oracle Access Manager Identity Server:

- Self Registration Test Case
- Lost Password Test Case
- Change Password Test Case
- Account Lockout Test Case

Self Registration Test Case

About 5% of the users perform self-registrations. Oracle created a self-registration workflow that takes a set of user attributes and enables the user immediately. This is a two step workflow that includes both Self Registration and Enable Password Policy.

Lost Password Test Case

For this test case, Oracle configured a password policy that accounts for the syntax and history of the password. 15% of logged in users try to change their password. Changing a password should succeed approximately 80% of the time. In the remaining 20% of cases, the password may be either too short, too weak, or repeated.

For example after a user incorrectly enters his or her password when asked for his or her challenges, and then they go into the password change screen. For the lost password screen, 10% of wrong user ID entered errors, and 10% of wrong challenges/response, and then 80% of the cases could make it to the next screen (change password).

Change Password Test Case

15% of the logged in users attempt to change their password. For this test case, Oracle configured a password policy that accounts for the syntax and history of the password. 15% of logged in users try to change their password. Changing a password should succeed approximately 80% of the time. In the remaining 20% of cases, the password may be either too short, too weak, or repeated.

Account Lockout Test Case

About 15% of logins result in unsuccessful authentications. Of these, 2% result in account lockouts. For example, if a user tries to log in three times with a wrong password, the account locks them out.

Access Server Baseline Performance Test Case

The following topics describe baseline performance test cases for the Oracle Access Manager for Access Server:

- Login Test Case
- LoginNavi Test Case

Login Test Case

The Login operation consists of one authentication and one authorization operation. Based on the test deployment, 100,000 user sessions are active during a given test. Sessions are not logged out once created. Instead, each session remains quiescent after login.

This test is designed to have about 85% successful authentications and 15% unsuccessful authentications.

LoginNavi Test Case

The LoginNavi operation consists of one login followed by 10 authorizations yielding a total of 11 operations per user session. The number of sessions and the iteration setting are the same as those in the Login test cases.

The test is designed to produce 5% failed authorizations. It was configured to redirect the authorization failure to a particular page to avoid 404 errors.

Integrated Baseline Performance Test Case

All above test scenarios were run together with the mix ratios specified in each scenario. This is a more realistic test case that represents the real-life usage of Oracle Access Manager. If both the Access Server and Identity Server are to be deployed, Oracle recommends that you size the system starting with this use case.

3 Performance Tuning

Once you have installed and configured Oracle Access Manager, you want to be sure to get the best possible performance out of the product.

This chapter provides information for experienced Oracle Access Manager administrators on how to optimize Oracle Access Manager performance. This chapter includes the following topics:

- Guidelines for Directory Tuning
- About LDAP Tools
- Tuning the Identity System
- Tuning Groups in the Identity System
- Tuning Workflows
- Tuning the Access System
- Tuning the Caches
- Tuning Your Network
- Resource-Intensive Operations

For general deployment recommendations, see Chapter 1. For details about Policy Manager tuning factors for Apache Web servers, see the *Oracle Access Manager Installation Guide.*

Guidelines for Directory Tuning

Oracle Access Manager stores its data in a Lightweight Directory Access Protocol (LDAP) directory. When diagnosing problems that appear to be due to Oracle Access Manager, you may want to check the performance of the directory. Additionally, many performance issues can be resolved by avoiding trips to the directory, especially for write operations.

The following sections discuss these topics:

- Checking the Performance of the Directory
- Directory Connection Pool Size
- Storing Workflow Tickets in the Directory
- Indexing Attributes in the Directory
- Changing the Number of Access Server-to-Directory Server Connections
- Deleting and Archiving Workflows

- Setting Read and Write Permissions for Administrators
- Configuring the Searchbase
- Applying Search Constraints
- Increasing Connections to the Directory in the Identity System
- Changing Directory Content
- Adjusting Cache Settings
- Deleting ObSyncRecord Entries from the Directory
- Performance Considerations for Microsoft Active Directory

Checking the Performance of the Directory

If directory performance is slow, it affects the performance of the Access Server or Identity Server. See your directory vendor's documentation for details. For Oracle Internet Directory, see the chapter on performance optimization in the *Oracle Internet Directory Administrator's Guide* for details.

Directory Connection Pool Size

The Identity Server and the Access Server open a configurable number of connections to LDAP servers. If you perform a large number of time-consuming searches of user profiles, you can increase the database connection pool size to improve performance.

You configure the connection pool size in the Identity System Console or Access System Console. You specify the details for connections used for accessing configuration data and policy data in configuration files, not through the System Console. These files reside in the /oblix/config/ldap directory for the Identity Server, Access Server, and Policy Manager. At startup, a number of connections are opened based on an Initial Connections setting. As needed, more connections are opened until the Maximum Connections setting is reached. Connections remain open until the Identity or Access Server shuts down or the directory server stops responding.

Differences Between Configured and Actual Connection Pool Size

The number of connections that the Identity or Access Server opens to the directory is different from the number of connections that you specify in the Identity System Console or Access System Console. This is because certain connection details are picked up from configuration files. Connections are specified in the following files:

- **Connections for accessing configuration data:** Three database configuration files reside in *component_install_dir*/oblix/config/ldap. The setting for the number of connections applies to each of the configuration files. The default value is 1, which means at any time at least three connections are open.
- **Connections for accessing policy information:** Four locator files are used to manage policy definitions. These locators reside in *AccessServer_install_dir*/oblix/config/ldap. The setting for the number of connections to be opened is on a per-locator-file basis. The default value is 1, which means at any time at least four connections are open.
- **Connections for database profiles that are created during setup:** During product setup, Oracle Access Manager automatically creates one or two database profiles, depending on if the configuration base and search base are the same or disjoint. Each database profile has one instance and the number of connections is set to 1. If the database profile supports authentication operations, a separate connection

pool is used for authentication operations. For example, if a database profile has one database instance and the number of connections is set to 1, two connections may be opened—one for authentication and one for other LDAP operations.

To illustrate the difference between the configured and actual number of connections, suppose that you have performed product setup without configuring disjoint domains. In this case, nine connections are opened by default:

- One each, for a total of three, for the configuration data.
- One each, for a total of four, for policy data
- One each, for a total of two, for a database profile with one database instance and one configured connection.

As another example, suppose that you configure two database profiles, each with one database instance and an Initial Connections setting of three, and a Maximum Connections setting of five. All database instances apply to the same directory, and configuration and user data are stored in the same directory. Authentication operations are supported. In this example, the minimum connections opened to the directory is 19, as follows:

- One each, for a total of three, for configuration data.
- One each, for a total of four, for policy data
- Three each, for a total of twelve, applies to the two database profiles.

  Each database profile will have six connections, since there will be two connection pools with three connections each. There will be one pool for authentication requests and another for other operations.

In this scenario, if you set the maximum number of connections to 2, the number of connections to the directory will range from 19 to 27.

Configuring the Connection Pool

The following procedure explains how to configure the number of LDAP connections.

To increase the connection pool size for user data

1. Navigate to Identity System Console, System Configuration, Directory Profiles.
2. Click the name of a user data directory server instance in the Configure LDAP Directory Server Profiles list on the Configure Profiles page.
3. Click the name of a directory server instance in the Database Instances list of the Modify Directory Server Profiles page.
4. On the Modify Database Instance page, modify the two parameters below:
   - **Maximum Connections**: Increases the number of available connections in the pool. The default value is 1. No upper limit is enforced; you can experiment with this setting to find a suitable value.
   - **Initial Connections**: Specifies how many connections are opened at database initialization. The default value is 1. There is no upper limit.

Storing Workflow Tickets in the Directory

As described in the *Oracle Access Manager Identity and Common Administration Guide*, by default each workflow step generates a ticket and Oracle Access Manager writes the ticket to the directory.

One way to avoid unnecessary directory write operations is to minimize the number of steps in a workflow. Another way to avoid unnecessary directory write operations is to change Oracle Access Manager default behavior of creating tickets for every step in a workflow.

Tickets are of little value when:

- The information in a step has little value for the next step.
- A participant triggers the workflow but there are no participants for other steps.

Oracle recommends that you monitor the number of workflow tickets that are stored in the directory server, and periodically delete old tickets manually or using a script-based utility.

Workflow Example

Suppose you create a workflow consisting of the following steps:

1. **Initiate**: The user initiates a self-registration.
2. **External action**: The request is passed to an external process.
3. **Enable**: The workflow is enabled.

If no one participates in steps 2 and 3, the workflow may not require a ticket.

Writing Workflow Tickets to the Directory

The `WFInstanceNotRequired` flag determines whether every workflow step generates a ticket. This flag is set to false by default, which means all workflow instances are written to the directory. When set to true, workflow instances are only written to the directory server when:

- A user action is required.
- Any errors are encountered (for example, the commit action fails).
- Any subflows are triggered.

If workflows do not require any other input or approval steps, it is a good idea to set the `WFInstanceNotRequired` flag to true.

enable Setting the `WFInstanceNotRequired` flag to true reduces the directory size and speeds up the workflow process. This parameter prevents the generation and storage of workflow tracking data that can generate overhead for the directory server. The disadvantage of setting this flag to true is that, because the instances are not being stored, you are not able to see all of the workflows that were executed from the Oracle Access Manager Workflow Monitor.

To avoid creating tickets for every workflow step

1. Set the WFInstanceNotRequired flag to true in the following file:

   IdentityServer_install_dir/identity/oblix/data/common/workflowdbparams.xml

2. Restart the Identity Server.

For more information, see the chapter on configuring the User, Group, and Organization Manager in the*Oracle Access Manager Identity and Common Administration Guide*.

Indexing Attributes in the Directory

Indexing improves search performance in the directory, although at the cost of slower database modification and creation operations and disk space. Oracle Access Manager automatically indexes key attributes when you run the setup program. The index files are specific to your directory server type, and are stored in:

IdentityServer_install_dir/identity/oblix/data/common

where *IdentityServer_install_dir* is the directory where the Identity Server is installed.

Index types include the following:

- An equality index improves searches for directory entries that contain a specific attribute value.
- A presence index improves searches for directory entries that contain a specific attribute.
- A substring index improves searches for entries that contain specific text strings.

You can enhance performance if you index attributes that are read during various operations. These include:

- Attributes used in searches that are triggered *indirectly* by user interaction.
- Attributes used in searches that are explicitly invoked by users.
- Attributes used in mapping filters for authentication schemes.
- Attributes in filters for dynamic member definitions in Group Manager.

Your directory documentation describes how to index attributes.

Oracle Access Manager provides an index file fragment showing how Oracle Access Manager-specific attributes can be indexed for each supported directory server. For example, the Oracle Access Manager attributes that can be indexed to improve performance in a directory managed by Sun (formerly iPlanet) directory server are listed in the file:

IdentityServer_install_dir/identity/oblix/data/common/iPlanet5_oblix_index_add.ldif

An example of an index entry in this file is:

```
index obclass pres,eq,sub
```

This means that the attribute obclass can be indexed for presence (*pres*), equality (*eq),* or substring (*sub)* .

Note: The iPlanet5_oblix_index_add.ldif file needs to be loaded to the directory server using ldapmodify.

Limitations of Indexing

Carefully weigh the benefits of indexing directory attributes for search operations against the performance hit incurred when these attributes are modified. When an attribute value is modified, indexes for that attribute may need to be rebuilt by the directory server. However, this is accomplished varies by directory server.

Be sure to read the manufacturer's guidelines for indexing. Avoid over-indexing attributes.

Indexing and User Deactivation

After you deactivate a significant number of users, Oracle Access Manager performance can start to degrade. If you experience this problem but want to maintain all deactivated users in the directory, use an equality index for the following attributes:

- ObIndirectManager
- ObUniqueMemberstr

An equality index allows you to search efficiently for entries containing a specific attribute value.

Indexing and Workflows

If it takes a long time to delete a workflow, index any attributes that the system checks during a delete operation. Performance issues with the delete workflow function usually result from attributes that need to be indexed.

Index the following attributes using the types of equality, presence, and substring:

- ObWorkflowID
- ObContainerID
- ObWFStepID
- ObWFTargetID
- ObIsWorkflowProvisioned
- ObLocationDN
- OblixGID
- Manager
- Secretary

An equality index type improves searches for directory entries that contain a specific attribute value. A presence index type improves searches for entries that contain a specific attribute. A substring index type improves searches for entries that contain specific strings.

For example, if you search for an attribute with a substring index type as follows:

```
cn=*lane
```

The search would match common names containing these strings:

```
John Lane Jane Tulane
```

If you conducted this search:

```
telephonenumber= *123*
```

The search would return all entries with telephone numbers that contain 123.

Indexing and Groups

The following attributes are used for group expansion and could be indexed to improve performance:

- Any attribute configured with the obSDynamicMember semantic type
- The obGroupExpandedDynamic attribute of the oblixAdvancedGroup object class
- All user attributes used in the dynamic filters of the groups to be expanded.

Indexing and Search Constraints

You can limit the performance impact of searches by forcing users to use a minimum number of characters for a search. This is controlled through the searchStringMinimumLength parameter. The default value for this parameter is 0.

To set a minimum number of search characters

1. Locate seachStringMinimumLength in:

   IdentityServer_install_dir/identity/oblix/apps/common/bin/oblixappparams.xml

2. Set its value to the minimum number of characters for a search.

   For instance, if you set it to 6, users could not search for "Smith" because it has only 5 characters. The constraint only applies to the longest search string supplied by the user. For example, if the search is on both surname and job title, and the user enters "manager" for job title, the longest string ("manager") has 7 characters, the search is allowed. A value of 0 turns off checking.

The `searchStringMinimumLength` constraint is enforced for searches that are conducted using the Oracle Access Manager user interface and for clients using other interfaces to Oracle Access Manager (for example, Identity XML clients).

To enforce a per-attribute minimum number of characters

1. Set the searchStringMinimumLength to 0 in the catalog.
2. In the JavaScript code, find the function validateSearchAndSubmit() in the file misc.js.
3. Modify the JavaScript code to handle the per-attribute checking you require.

> **Note:** This technique does not enforce the constraint on users of Identity XML clients.

As with other parameters, you can set searchStringMinimumLength to one value in the global oblixappparams.xml catalog, and to a different value in one or more of the application-specific catalogs. For example, to override the global value and set searchStringMinimumLength to a different value for User Manager, specify the parameter and its value for User Manager only in:

IdentityServer_install_dir/identity/oblix/apps/userservcenter/bin/userservcenterparams.xml

Changing the Number of Access Server-to-Directory Server Connections

By default, each Access Server opens only one connection to a primary directory server and one connection to a failover directory server (if failover is configured). This may not be optimal for systems with heavy loads.

To configure additional Access Server-to-directory server connections for Oracle Access Manager configuration and policy data, use the command line tool configureAAAServer. The *Oracle Access Manager Access Administration Guide* provides information on this tool. In environments with very high loads (perhaps 1,000 hits/second on the Access Server), creating twenty connections has significantly improved performance.

Note: If your directory is running on a low-powered system, the directory itself may be the limiting factor. In this case, increasing the number of connections may have little benefit. If the computer running the directory is the problem, increase the number of directory server instances and configure load balancing among them.

Deleting and Archiving Workflows

To prevent the Oblix tree from getting too large, periodically delete or archive workflows. Deleting a workflow instance removes the directory entries associated with that instance. Archiving a workflow instance keeps a record of the instance in an LDIF file and deletes the instance from the directory.

The frequency for archiving or deleting depends on how frequently your workflows are used.

Archived workflows are stored in LDIF format. The default archive file is:

IdentityServer_install_dir/identity/oblix/data/common/wfinstance.ldif

Multiple archive operations add information to this file. You can change the archive file by changing entries in the following files:

- User Manager

  IdentityServer_install_dir/identity/oblix/apps/userservcenter/bin/usc_wf_params.xml

- Group Manager

  IdentityServer_install_dir/identity/oblix/apps/groupservcenter/gsc_wf_params.xml

- Org Manager

  IdentityServer_install_dir/identity/oblix/apps/objservcenter/osc_wf_params.xml

The entry to change is similar to the following:

```
<NameValPair ParamName="archiveFileName" Value="wfinstance.ldif"/>
```

To delete or archive a workflow

1. In User, Group, or Organization Manager, click Requests.
2. Click Monitor Requests.

   For subflows, if the first step has not been processed, the Date Processed field will be empty.
3. In the Search fields, select your search criteria
4. Click Go.

   The results appear below the search fields.
5. Click individual check boxes or the Select All button.
6. Click Next or Previous as necessary to see other results.
7. Click the Delete or the Archive button.

Setting Read and Write Permissions for Administrators

By default, administrators can view all attributes in the directory so that they can perform initial setup of Oracle Access Manager. Because end users only have read permissions for specific attributes, there is a difference in performance for an administrator and an end user.

The parameter BypassAccessControlForDirAdmin controls whether the Master Identity Administrator or a delegated administrator is permitted to view all attributes in the directory. By default, the value of the parameter BypassAccessControlForDirAdmin is set to true. You can set the value of this parameter to false in the following file:

IdentityServer_install_dir/identity/oblix/apps/common/bin/globalparams.xml

If the BypassAccessControlForDirAdmin flag is set to false, performance is the same for non-administrative and administrative users. This is because attribute read and write permissions are applied to the administrative users.

The following is an example of the parameter entry:

```
<SimpleList>
  <NameValPair ParamName="BypassAccessControlForDirAdmin" Value="true">
  </NameValPair>
</SimpleList>
```

Configuring the Searchbase

Searchbase configuration can affect Oracle Access Manager performance.

Guidelines for searchbase configuration include:

- Set the searchbase for the user object class at a point in the people branch where all users can be found, for example, the root.
- Minimize the number of searchbases that you configure per user, group, or organization object class.
- It is more efficient to configure attribute access controls to the class attribute of the user object than it is to limit user access by setting multiple search bases.
- Use the default searchbase filter objectclass=* whenever possible.
- Instead of defining searchbase filters, configure read and write permissions for attributes.
- Avoid configuring searchbases using substitution syntax.
- Avoid search bases or ACLs that contain substring searches, for example `"...(...=*something*)..."`
- Configure workflow rules and roles to control the user's ability to view and modify attribute values.

For more information, see the chapter on making schema data available to the Identity System in the *Oracle Access Manager Identity and Common Administration Guide*.

Setting a Searchbase Filter

When configuring a searchbase, if you add a filter and enter a filter other than the default (objectclass=*), the Resource Filter Search scope takes effect. The Resource Filter Search scope has no effect unless you define a filter.

The default searchbase filter (objectclass=*) instructs Oracle Access Manager to apply the user's search criteria to the entire section of the directory tree that has been selected as the searchbase. If you specify other criteria, for example, (telephone=408*), Oracle Access Manager retrieves all users who satisfy that criteria, then applies your search criteria to the subset specified by the filter instead of to the entire tree.

This can degrade performance, especially if the filter retrieves a large number of entries. For example, if you specify (objectclass=wwmOrgPerson) in the filter, Oracle Access Manager may recover all users (assuming all users satisfy this filter) under the specified tree before applying your search criteria. This can seriously degrade performance. You do not have to specify (objectclass=wwmOrgPerson) because the searchbase is already set for that object class. In general, setting read and write privileges for attributes is a better way of controlling user access than setting a searchbase filter. To optimize directory performance, avoid defining complex filters for the searchbase. In the case of searchbases for a tab, only objects of the class which that tab applies to are searched. There is no need to mention (objectclass=*) explicitly.

If you must enter a filter, you can limit the performance impact by setting the resourceSearchFilter scope parameter to 1.

Applying Search Constraints

The larger the number of entries that a search actually or potentially returns, the longer the search takes. For an interactive application such as Oracle Access Manager, large result sets may be unmanageable for the end user.

Your directory may allow you to limit the time that the directory server spends on a search, the size of the result, or both.

Increasing Connections to the Directory in the Identity System

The LDAP Database Instance Maximum Connections is the maximum number of connections allowed from the Identity Server to the directory servers. This value defaults to 1, but you may see a performance improvement by setting this value to more than 1. (With a SQL profile, the default is 5.)

There is no optimal value for the maximum connections. There are variables to consider such as directory configuration and hardware. You may want to consider setting the maximum directory connections no higher than the number of threads set for a Identity Server. For example, if the Identity Server is configured with only 20 threads, there is no benefit in having more than 20 connections because no Identity worker threads can take advantage of the additional connections.

You may want to increase the maximum connections by increments of 5, and monitor your system performance. If performance is worse, decrease the number of connections. The Initial Connections and the Maximum Connections settings work together. When an Identity Server starts, it opens the number of connections to the directory as specified in the Directory Profile Initial Connections. Those connections are then pooled and used by the Identity Server.

Note: Additional connections can introduce overhead that in turn can hurt performance. For example, if you restart the directory server, the Identity Server has to reconnect the configured number of connections.

Changing Directory Content

This section describes modifications that can be made to the directory content to affect Oracle Access Manager operation. For a discussion of the tools needed to make these changes, see "About LDAP Tools" on page 3-16.

Ordering the Columns in a Search Results List

A search of the Policy Manager for policy domain names or policy names returns a default set of columns in a default order. You can display different columns or change the order of columns. While this does not affect performance in terms of response times, it may improve your satisfaction with the results returned from a search.

To modify results for a policy or policy domain name search

1. Locate the DN, for example:

   ```
   obname=SDSearchColumnList, obapp=WRSC, o=Oblix, o=Company, c=US2
   ```

2. Under this DN, find the current column list.

   The column list consists of the values for obsearchresultcolumns.

   Possible values for a search of policy names are:

   - `WRORname`: Policy Name
   - `AuthentPolicyName`: Authentication Rule Name
   - `AuthorPolicyName`: Authorization Rule Name
   - `AdminPolicyName`: Auditing Rule Name
   - `URLPrefix`: URL for the domain controlled by the policy

   Possible values for a search of policy domain names are:

   - *SDName*: Policy Domain Name
   - *AuthentPolicyName:* Authentication Rule Name
   - *AuthorPolicyName*: Authorization Rule Name
   - *AdminPolicyName*: Auditing Rule Name
   - *AbsPathPattern*: Path to the controlled domain

For example, the following is an LDIF extract that shows the policy name, authentication rule name, authorization rule name, and URL for the domain controlled by the policy:

```
dn: obname=SDSearchColumnList, obapp=WRSC, o=Oblix, o=Company, c=US
objectclass: top
objectclass: OblixWRSSearchResultColumns
obname: SDSearchColumnList
obSearchResultColumns: WRORName
obSearchResultColumns: AuthentPolicyName
obSearchResultColumns: AuthorPolicyName
obSearchResultColumns: URLPrefix
```

To change the order or content of the displayed search results, modify this information and store it back to the directory.

Changing the Bind DN

Using Directory Manager as the bind DN bypasses search limits such as look through limit, size limit, and time limit. Large searches can tie up the directory server and increase the amount of processing that is required by the Identity Server.

You can create another user to use as a bind DN.

Note: This procedure illustrates the technique for Sun (formerly Netscape and iPlanet) directories. Consult your directory server's administration manual for instructions on how to create a directory user with the appropriate permissions.

Create your new user, for example orcladmin. Use the LDAP directory to give the user permissions to act as an administrator for Oracle Access Manager. For the searchbase, configuration base, and all branches underneath them, this user needs the following permissions:

- Read
- Write
- Add
- Delete
- Search
- Compare
- Selfwrite

To change bind DN permissions

1. From the Sun LDAP Server Admin Console, navigate to the directory server instance and open it.
2. Choose the Directory tab, then locate and right-click the branch for which you want to set permissions.

   For example:

   for o=Company, c=US, you would find the Company node and right-click to get a menu.
3. Choose Set Access Permissions in the menu.
4. Add a new access permission, or edit the one already in place.

   There should be two lines when you are done. The first is a default deny for everyone, the second is the allow statement.
5. Choose Allow, then search users and groups to find the new administrative user.
6. Set the rights for this user.

If you make this change after installing Oracle Access Manager, change the bind DN for the Identity Server to match the value you created in the directory.

Adjusting Cache Settings

Caching avoids the latency of unnecessary lookups or calculations. If they keep the results of recent requests close to the consumer, producers can respond quickly by returning cached results rather than recalculating them. The cost of a cache is usually extra RAM or disk space.

A directory that manages large entries will likely hit a maximum cache size limit first, while a directory of small entries might hit the maximum number of entries limit first. If you only care about one of these two criteria, set the other to an unrealistically large number, or match their limits by setting the maximum cache size first, then dividing that number by the size of each entry to get the number you should use for maximum entries in cache.

Deleting ObSyncRecord Entries from the Directory

A directory entry called ObSyncRecord is created each time a change is made to a cacheable item in Oracle Access Manager, for example a user entry, policy domain change, host identifier change, resource change. This directory entry enables the cache for different Oracle Access Manager Servers to stay in sync. If many updates are made to these entries over a short time interval, the number of ObSyncRecord entries that are written can cause a directory performance issue.

You may want to delete ObSyncRecord entries at a regular interval. If you do this, you may want to check each entry before deleting it to ensure that the entry has been in the directory for a time period that is greater than the cache-flush interval.

> **See Also:**
>
> - Details about cache flush intervals and operations in Chapter 5
> - Details about archiving and purging sync records, and about detecting and restoring corrupted sync records in the *Oracle Access Manager Access Administration Guide*
> - Details on flushing user data from the cache in the *Oracle Access Manager Access Administration Guide*

Performance Considerations for Microsoft Active Directory

The following performance tuning considerations apply to deployments that use Microsoft Active Directory:

- Pointing Directly to a Domain Controller to Avoid Potential Data Inconsistency Problems
- Using LDAP Over SSL Rather than ADSI to Connect to Microsoft Active Directory
- Fine Tuning Appropriate Active Directory Configuration Parameters to Optimize Performance

For performance tuning recommendations, see Chapter 3. For details about installing Microsoft Active Directory with Oracle Access Manager, see the *Oracle Access Manager Installation Guide*. When configuring Microsoft Active Directory for Oracle Access Manager, see the*Oracle Access Manager Identity and Common Administration Guide*.

Pointing Directly to a Domain Controller to Avoid Potential Data Inconsistency Problems

When deploying in a Microsoft Active Directory environment, always point directly to a domain controller to avoid potential data inconsistency problems.When using Microsoft Active Directory as the user and group directory, ensure that you point directly to the domain controller and not to the DNS alias. This avoids problems that are caused by transient inconsistencies. For example, this practice avoids the possibility of dynamic DNS or round robin aliasing diverting connections to servers that are slow, remote, or contain out-of-date data. To implement high availability in an Active Directory environment, configure each of the domain controllers as a directory connection, and then tune the performance for reads and writes from Oracle Access Manager.

This recommendation also applies to Active Directory forests that contain multiple sub-domains. For this you should create separate directory profiles for each sub-domain in addition to the root domain, with each sub-domain pointing at the appropriate domain controller server or servers.

Using LDAP Over SSL Rather than ADSI to Connect to Microsoft Active Directory

When deploying Oracle Access Manager on Windows against Microsoft Active Directory, it is important to consider whether using LDAP over SSL is more appropriate than ADSI, as LDAP over SSL typically performs and scales better than ADSI, particularly in environments with high transaction volume. Also, using LDAP over SSL allows Oracle Access Manager (Access Server, Identity Servers, and Policy Manager) to rely on specified Active Directory instances. In contrast, when using ADSI, the specific Active Directory instance that Oracle Access Manager connects to is determined on-the-fly. This instance select may be undesirable if the overall response and performance across all available Active Directory domain controllers varies significantly.

Fine Tuning Appropriate Active Directory Configuration Parameters to Optimize Performance

When deploying Oracle Access Manager in a Microsoft Active Directory environment, it is important that the configuration parameters in Active Directory be set appropriately, so that overall Oracle Access Manager performance is optimized. Table 2-1 lists the most relevant Active Directory configuration parameters.

Table 3–1 Active Directory Configuration Parameters

Active Directory Configuration Parameter	Description	Default Value	Impact for Oracle Access Manager
MaxActiveQueries	Specify the maximum number of concurrent LDAP search operations that are permitted to run at the same time on a domain controller. When this limit is reached, the LDAP server returns a "busy" error. Note: This control has an incorrect interaction with the MaxPoolThreads value. MaxPoolThreads is a per-processor control, while MaxActiveQueries defines an absolute number. Starting with Windows Server 2003, MaxActiveQueries is no longer enforced. Additionally, MaxActiveQueries does not appear in the Windows Server 2003 version of Ntdsutil.exe.	20	20 This value should be greater than the total number of service threads in Oracle Access Manager, for all service threads to be able to perform search operations at the same time. The total number of service threads in Oracle Access Manager is the summation of number of threads in Identity and Access servers and the number of threads in the Web server hosting the Policy Manager.
MaxConnections	Specify the maximum number of simultaneous LDAP connections that a domain controller will accept. If a connection comes in after the domain controller reaches this limit, the domain controller drops another connection.	5000	This value should be greater than or equal to the number of connections that Oracle Access Manager establishes with any Active Directory domain controller.
MaxConnIdleTime	Specify the maximum time, in seconds, that the client can be idle before the LDAP server closes the connection. If a connection is idle for more than this time, the LDAP server returns an LDAP disconnect notification.	900 seconds	If the maximum session time is set in Oracle Access Manager, then this value should not slightly higher than it.
MaxPageSize	Specify the value for controlling the maximum number of objects that are returned in a single search result, independent of how large each returned object is. To perform a search where the result might exceed this number of objects, the client must specify the paged search control. This is to group the returned results in groups that are no larger than the MaxPageSize value. To summarize, MaxPageSize controls the number of objects that are returned in a single search result.	1,000	This value should be greater than the number of entries returned in any search request made by any Oracle Access Manager component. Oracle Access Manager component performs search operations for the following two cases: ■ When a user requests search on other users ■ Oracle Access Manager component internally performs searches on configuration data while processing requests. If blank, the search is allowed (in general, any search that results in all user entries in the system), then this value should be greater than the number of users in the system. If in general this kind of user searches are restricted or no one does these kinds of requests, then this value should be greater than the highest number of nodes under the following two nodes in Oracle Access Manager configuration data: ■ obapp=PSC,o=Oblix,<co ■ nfig_base> ■ obcontainerId=workflo ■ wInstances,o=Oblix, ■ ,<config_base>
MaxPoolThreads	The maximum number of threadsper-processor that a domain controller dedicates to listening for network input or output. This value also determines the maximum number of threads per-processor that can work on LDAP requests at the same time.	4 threads-per-process or	If the Identity or Access Server run a single processor server, then this value should be greater than the number of connections established by Oracle Access Manager. This way, the domain controller can perform all operations in parallel.

About LDAP Tools

Directory applications use LDAP as a standard tool to create, modify, and report data stored in the directory. Tools are available to allow easy manipulation of this data. This section introduces these tools. More detail is available from the manufacturer of your server application.

Viewing Directory Content in LDIF Files

The structure of a directory, and the data contained within it, is represented by the content of an LDAP Data Interchange Format (LDIF) file. The file can be *output*, the formatted result of a request made to the directory by an LDAP reporting tool, such as LDAPSEARCH. It can also be *input*, data that is intended for insertion to the directory, either as entirely new data, or as an update to existing data, using an updating tool such as LDAPMODIFY.

The following is an example, part of an LDIF file taken from a Oracle Access Manager Demo Directory:

```
dn: cn=John Kramer, ou=Sales, o=Company, c=US
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: companyOrgPerson
cn: John Kramer
sn: Kramer
telephonenumber: 415-555-5269
facsimiletelephonenumber: 415-333-1005
title: Account Manager
departmentnumber: 1204
employeetype: Fulltime
employeenumber: 521-321-4560
givenname: John
.
.
Reporting Directory Content with LDAPSEARCH
```

LDAPSEARCH is one possible tool that can be used to report directory content. There are others, which use a different syntax, but the concepts are the same.

LDAPSEARCH can be used in either a command line or interactive mode. The command line approach is preferable, as it allows users to provide the text of the report request by means of a input file. It is easy to verify the content of this file before making the request. You can correct errors by changing a few characters in the file rather than retyping the full request, which would be necessary in the interactive mode.

LDAPSEARCH Command-Line Format

The command-line format for LDAPSEARCH is:

```
ldapsearch params filter attr_list
```

Note: Each of the three categories shown between <> is optional; if all are omitted, LDAPSEARCH drops into interactive mode, which is not discussed here.

The categories are as follows:

- **<params>**: P*arameters* tell LDAPSEARCH how to operate. One of them, *-f*, is used to specify a filter file. If instead the search filter is provided on the command line, all parameters must be stated before the filter is stated.
- **<filter>**: The *filter* instructs LDAPSEARCH to provide a subset of the data that would otherwise be provided. For example, a filter could require that only names beginning with N be reported. A filter provided on the command line must be enclosed in quotes.
- **<attr_list>:** The *attributelist*, if included on the command line, overrides the default attribute listing. The default list shows all attributes belonging to the directory entry, except operational attributes. If you wish to see only some of these attributes listed, provide their names in the command line, following the filter, and separated by spaces. If you want to see operational attributes, provide their names in the command line. If you follow the operational attributes with a * you get the default list of attributes as well.

LDAPSEARCH Command-Line Parameters

Parameters are always provided in the form:

```
-p pdata
```

where *p* is the parameter, preceded by a dash, and pdata is the information (data) required for the parameter, if any. If the data contains one or more spaces, it must be enclosed in double quotes:

```
-p "pdata with spaces"
```

Following is a list of commonly used parameters. See the reference document for your version of LDAPSEARCH, or use the parameter `/?` to see them listed:

- **-A**: Retrieve the attribute names only, not the attribute values.
- **-b**: Searchbase, the starting point for the search. The value specified here must be a distinguished name that is in the directory. Data provided for this parameter MUST be in double quotation marks. For example:

  ```
  -b "cn=Barbara Jensen, ou=Development, o=Oblix.com"
  ```

- **-D**: Distinguished name of the server administrator, or other user authorized to search for entries. This parameter is optional if anonymous access is supported by your server. For example:

  ```
  -D "uid=j.smith, o=Oblix.com"
  ```

- **-f**: Specifies the file containing the search filters to be used in the search. For example:

  ```
  -f filterfile
  ```

- **-h**: Host name or IP address of the computer on which the directory server is installed. This entry is optional; if no host name is provided, LDAPSEARCH uses the local host. For example:*-h myserver.com*
- **-H**: This generates a list of all possible LDAPSEARCH parameters.
- **-p**: Port number that the directory server listens at. For example:

  ```
  -p 1049
  ```

- **-s**: Scope of the search. The data provided for the scope must be one of the following:
  - **base:** Search only the entry specified in the -b option.
  - **one**: Search only the immediate children of the entry specified in the -b parameter; do not search the actual entry specified in the -b parameter.
  - **sub**: Search the entry specified in the -b parameter, and all of its descendants. That is, perform a subtree search starting at the point identified in the -b parameter. This is the default, if the -s parameter is not used.
  - **-S**: Designates the attributes to use as sort criteria, controlling the order in which the results are displayed. You can use multiple -S arguments if you want to sort on multiple criteria. The default behavior is not to sort the returned entries. In the following example, the search results are sorted first by surname and then, within surname, by first name:*-S sn -S firstname*
  - **-w**: Password associated with the distinguished name that is specified in the -D option. If you do not specify this parameter, anonymous access is used. For example:*-w password*

    Note: The Oracle Internet Directory LDAP tools have been modified to disable the less secure options -w *password* and -P *password* when the environment variable LDAP_PASSWORD_PROMPTONLY is set to TRUE or 1. If you use -q (or -Q), the command prompts you for the user password (or wallet password). Oracle recommends that you set this variable whenever possible.

  - **-x**: Specifies that the search results are sorted on the server rather than on the client. This is useful if you want to sort according to a matching rule, as with an international search. In general, it is faster to sort on the server than on the client.
  - **-z:** Specifies the maximum number of entries to return in response to a search request. For example:*-z 1000*

LDAPSEARCH Examples

To get the surname (sn), common name (cn), and given name for every employee named John in the Sales organization, from the directory server listening at port 392, entirely from the command line, you could provide the following information:

```
ldapsearch -p 392 -b "ou=sales, o=company, c=US" -s sub "givenname=John" sn cn
givenname
```

Results could be something like:

```
dn: cn=John Jackson, ou=Sales, o=Company, c=US
sn: Jackson
cn: John Jackson
givenname: John
dn: cn=John Kramer, ou=Sales, o=Company, c=US
sn: Kramer
cn: John Kramer
givenname: John
```

You can get the same results by using a filter file. For example, a file called namejohn containing the filter:

```
givenname=John
```

can be used, with the following command line:

```
ldapsearch -p 392 -b "ou=sales, o=company, c=US" -s sub -f namejohn sn cn
givenname
```

Changing Directory Content with LDAPMODIFY

The LDAPMODIFY tool changes or adds directory content. There are other tools, but the concepts are similar. LDAPMODIFY opens a connection to the specified server, using the distinguished name and password you supply, and modifies the entries based on LDIF update statements contained in a specified file. LDAPMODIFY can also be run in interactive mode, a method that is not discussed here.

If schema checking is active when you use LDAPMODIFY, the server performs schema checking for the entire entry before applying it to the directory. If the directory server detects an attribute or object class in the entry that is not known to the schema, then the entire modify operation fails. Also, if a value for a required attribute is not present, the modify operation fails. Failure occurs even if the value for the problem object class or attribute is not being modified.

Note: Keep schema checking turned on at all times to avoid accidently adding data to the directory that could be unusable or cause schema violations when schema checking is turned back on. Schema checking is controlled at the directory administration server console and is generally on by default.

LDAPMODIFY Command-Line Format

The command line format for LDAPMODIFY is:

- `ldapmodify <params>`

Note: The params category is optional; if it is omitted, LDAPMODIFY drops into interactive mode, which is not discussed here.

<params> These *parameters* tell LDAPMODIFY how to operate. One of them, *-f*, can be used to specify a file describing modifications to be made to the directory.

LDAPMODIFY Command-Line Parameters

Parameters are always provided in the form:

```
-p pdata
```

where p is the parameter, preceded by a dash and followed by a space, and pdata is the information required for the parameter, if any. If the data contains one or more spaces, it must be completely enclosed in double quotes:

```
-p "pdata with spaces"
```

Following is a partial list of commonly used parameters. Use the parameter `/?` to see all of them.

- **-a**: Allows you to add LDIF entries to the directory without requiring the *changetype:add* LDIF update statement, which is necessary in the interactive mode. This provides a simplified method of adding entries to the directory; in particular, this allows you to directly add a file created by LDAPSEARCH and modified to make changes.
- **-c**: Forces the utility to run in continuous operation mode. Errors are reported, but the utility continues with modifications. Default is to quit after reporting an error.
- **-D**: Distinguished name of the server administrator or other user authorized to change directory entries. This parameter is optional if anonymous access is supported by your server. For example:*-D "uid=j.smith, o=Oblix.com"*
- **-f**: Provides the name of the file containing the LDIF update statements used to define the directory modifications. For example:*-f changestomake.txt*
- **-h**: Hostname or IP address of the computer on which the directory server is installed. This entry is optional; if no hostname is provided, LDAPSEARCH uses the local host. For example:*-h mozilla*
- **-H**: Lists all possible LDAPMODIFY parameters.
- **-p**: Port number that the directory server uses. For example:*-p 1049*
- **-w**: Password associated with the distinguished name that is specified in the -D option. If you do not specify this parameter, anonymous access is used. For example: *-w password*

> **Note:** The Oracle Internet Directory LDAP tools have been modified to disable the less secure options -w *password* and -P *password* when the environment variable LDAP_PASSWORD_PROMPTONLY is set to TRUE or 1. If you use -q (or -Q), the command prompts you for the user password (or wallet password). Oracle recommends that you set this variable whenever possible.

LDAPMODIFY Examples

Suppose you want to change the stored given name of John Kramer, as reported under the discussion of LDAPSEARCH. The data reported back was:

```
dn: cn=John Kramer, ou=Sales, o=Company, c=US
sn: Kramer
cn: John Kramer
givenname: John
```

This output can be used to derive an input file, ToHarvey, whose content might be:

```
dn: cn=John Kramer, ou=Sales, o=Company, c=US
changetype:modify
replace:givenname
givenname: Harvey
```

The command line would then be:

```
ldapmodify - p 392 -f ToHarvey
```

If you were to now search the directory with the command line:

```
ldapsearch -p 392 -b "ou=sales, o=company, c=US" -s sub "givenname=Harvey" sn cn
givenname
```

The response would be:

```
dn: cn=John Kramer, ou=Sales, o=Company, c=US
sn: Kramer
cn: John Kramer
givenname: Harvey
```

Tuning the Identity System

To improve Identity System performance, you can tune the way the Identity System interoperates with the directory server, and you can ensure that the Identity System servers and plug-ins are configured optimally.

This section discusses the following topics:

- Tuning Identity System Searches
- Create Thread-Safe Plug-Ins
- Consider Pooling Identity Servers
- Configure Identity Servers from a File System Level
- Configure Identity Servers to Use 3 GB of Virtual Memory

Tuning Identity System Searches

As discussed in "Guidelines for Directory Tuning" on page 3-1, the directory server plays a major role in overall system performance for Oracle Access Manager. For the Identity System, the types of searches that users conduct in the directory can significantly affect performance.

This section discusses the following topics regarding optimization of Identity System searches in the directory:

- Restricting the Operators Used in a Search
- Requiring the User to Enter a Minimum Number of Characters in a Search Field
- Restricting the Number of Entries Returned on a Search

Restricting the Operators Used in a Search

When users conduct a search in an Identity System application, the search bar presents a drop-down list with options for matching the search input with a set of results. These options include the following:

- That contains
- Contains in order
- Equals
- Less than
- Greater than
- Begins with
- Ends with
- Sounds like

The "greater than" and "less than" operations can result in many entries being searched and retrieved. By eliminating these choices, you can improve the performance of search operations. You configure the search operations in a set of parameter files, as described in the following procedures.

In addition to providing a search bar, the Identity System applications also provide a query builder function that enables users to construct search filters. You can eliminate the "greater than" and "less than" choices from the query builder as well as the search bar.

> **Note:** See the appendix on configuration parameters in the *Oracle Access Manager Customization Guide* for more information on the parameters discussed in the following procedures.

To eliminate the "greater than" and "less than" search operations

1. To modify the search bar, open each of the following files in a text editor:
   - *Install_dir*\identity\oblix\apps\userservcenter\bin\userservcenterparams.xml
   - *Install_dir*\identity\oblix\apps\groupservcenter\bin\groupservcenterparams.xml
   - *Install_dir*\identity\oblix\apps\objservcenter\bin\objservcenterparams.xml
   - *Install_dir*\identity\oblix\apps\selector\bin\selectorparams.xml

   Where *Install_dir* is the directory where Oracle Access Manager is installed.
2. Find the entry for the ObEnhanceSearchList parameter in each of these files.
3. Edit this entry in each of these files so that it only contains the following parameters:

```
<ValNameList ListName="ObEnhanceSearchList" >
         <NameValPair ParamName="OOS" Value="MOOS"/>
         <NameValPair ParamName="OSM" Value="MOSM"/>
         <NameValPair ParamName="OEM" Value="MOEM"/>
         <NameValPair ParamName="OBW" Value="MOBW"/>
         <NameValPair ParamName="OEW" Value="MOEW"/>
   </ValNameList>
```

4. To modify the query builder, open the following file in a text editor:

   Install_dir\identity\oblix\apps\querybuilder\bin\querybuilderparams.xml
5. Edit the element ObQBOperatorsList to have only the following values:

```
<ValList ListName="ObQBOperatorsList" >
            <ValListMember Value="CND_CON"/>
            <ValListMember Value="CND_DNC"/>
            <ValListMember Value="CND_EQ"/>
            <ValListMember Value="CND_NEQ"/>
            <ValListMember Value="CND_PRE"/>
            <ValListMember Value="CND_NPR"/>
            <ValListMember Value="CND_BW"/>
            <ValListMember Value="CND_EW"/>
        </ValList>
```

Requiring the User to Enter a Minimum Number of Characters in a Search Field

If you require users to enter a minimum number of characters in a search, fewer entries are searched in the directory. For example, if users must enter at least three characters, a directory search is likely to only involve and return a subset of all possible entries. This, in turn, can improve performance.

To require the user to enter a minimum number of characters in a search field

1. To specify the minimum number of characters users must enter in the primary search bar, open the following file in a text editor:

   Install_dir\identity\oblix\apps\common\bin\oblixappparams.xml

   Where *Install_dir* is the directory where Oracle Access Manager is installed.

2. Set the value of the searchStringMinimumLength parameter to the minimum length of the string that users can input, as illustrated in the following example:

   ```
   <NameValPair ParamName="SearchStringMinimumLength" Value="3"/>
   ```

3. To specify the number of characters that users must enter in the search bar on the Group Manager's Manage Group Members page, open the following file in a text editor:

   Install_dir\identity\oblix\apps\groupservcenter\bin\
   groupservcenterparams.xml

   Where *Install_dir* is the directory where Oracle Access Manager is installed.

4. Set the value of the groupMemberSearchStringMinimumLength parameter to the minimum length of the string that users can input, as illustrated in the following example:

   ```
   <NameValPair ParamName="groupMemberSearchStringMinimumLength" Value="3"/>
   ```

Restricting the Number of Entries Returned on a Search

You can set a limit on the number of elements that can be returned as the result of a search in an Identity System application. This limits the effect that a search can have on performance.

You can configure the maximum number of search results that are returned from the directory server on the Size Limit parameter for the directory server instance profile. For example, if you set the value of this parameter to 1,000, a maximum of 1,000 entries can be returned in the search results. The default value of 0 indicates that an unlimited number of results can be returned.

You can specify different size limits for different directory server profiles. For example, you can configure a size limit of 0 (unlimited) for the directory server instances that Identity System administrators use, and you can configure a limit of 1,000 for the directory server profiles that are used by end users.

To restrict the number of entries returned on a search

1. From the Identity System Console, click System Configuration.

2. On the System Configuration page click Directory Profiles.

3. Click the link for the directory server profile to which you want to add a database instance.

   The Modify Directory Server Profile page appears.

4. Scroll down to Database Instances and click the database instance you want to configure.

   The Modify Database Instance page appears.

5. Configure the Size Limit parameter to indicate the maximum number of search results that can be returned from the directory server.

Create Thread-Safe Plug-Ins

Both the Access Server and Identity Server are multithreaded. Ensure that all Identity Event plug-ins are thread-safe. This recommendation also applies to Identity Event plug-ins.

Consider Pooling Identity Servers

It is a good practice to use at least two Identity Servers running in a pooled primary configuration. Pooled primary means using multiple Identity Servers that run as primary servers, with one or more WebPass instances connecting to the primary Identity Servers.

You can use separate Identity Servers as secondary servers when using a pooled primary approach. If you have only two servers, a pooled primary configuration is recommended over using one primary and one secondary server. When running a pooled primary configuration, it is best to use identical but separate hardware for the Identity Servers.

Advantages of pooled primary mode

- Increased performance through load balancing
- Increased availability through multiple servers
- Automatic failover

Disadvantages of pooled primary mode

- The cost of additional hardware.

  If there are no secondary servers, each primary server needs to be sized to handle the total expected load if the other primary servers are unavailable.
- Additional system configuration.

Configure Identity Servers from a File System Level

Identity Server configuration and stylesheet files must be identical on all servers. This applies to all configurations that use multiple Identity Servers. You should configure all Identity Servers from a file system level, that is, ensure that all directory and file system structures are identical.

Configure Identity Servers to Use 3 GB of Virtual Memory

On Windows, if the Identity Server causes high memory utilization, the system can crash. You can configure an Identity Server to use 3 GB of virtual address space, if 2 GB addressing is already enabled in the boot.ini file. Identity Server can now use 3 GB of virtual address space.

By default, the virtual address space of Identity Server is limited to 2 GB. You can configure a /3 GB switch in the Boot.ini file to allocate 3 GB of virtual address space to an Identity Server that uses `IMAGE_FILE_LARGE_ADDRESS_AWARE` in the process header. This switch allows applications to address an additional GB of virtual address space beyond the usual 2 GB limit.

The following example shows how to add the /3GB parameter in the Boot.ini file to enable Identity Server memory tuning:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(2)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(2)\WINNT="????" /3GB
```

> **Note:** The Boot.ini file typically resides in the system's root directory.

See the following URL for more information:

http://www.microsoft.com/whdc/system/platform/server/PAE/PAEmem.mspx

Tuning Groups in the Identity System

Group size can adversely affect performance. For instance, if you have groups of more than 30,000 users, operations involving these groups may be slow. Similarly, nested groups can result in slow performance during evaluation due to the large number of members in the nested groups.

The following sections discuss guidelines for group usage:

- General Recommendations for Tuning Groups in the Identity System
- Guidelines for Working with Large Static Groups
- Tuning the Group Manager Application
- Tuning the User ID Cache

General Recommendations for Tuning Groups in the Identity System

The following topics provide general recommendations for tuning groups to improve performance in the Identity System:

- Use Dynamic Groups Instead of Static Groups
- Turning off Dynamic Group Evaluation for the Identity Server
- Use Nested Groups with Caution

Use Dynamic Groups Instead of Static Groups

Oracle Access Manager evaluates dynamic group membership automatically based on a filter and user attributes. In general, this makes dynamic groups easier to manage than static ones. Also, evaluation of dynamic group membership is an inexpensive operation, enabling a dynamic group to be as large as you want without degrading performance.

If other applications only understand static groups, you can expand a group into a static list of members using the Group Manager application in the Identity System. To schedule periodic group expansion, you can also use the `expandGroup` IdentityXML or a Web service call to add the group expansion function to a cron job. See the *Oracle Access Manager Developer Guide* for details.

Turning off Dynamic Group Evaluation for the Identity Server

You can use the following procedure to improve performance for attribute access control through group evaluation. Whenever access to a resource is determined by a group, the membership to the group is evaluated in the following order: static, dynamic, and then nested. When you are not using dynamic groups, Oracle recommends that you turn off evaluation of that group type.

Functions that are impacted when you set `TurnOffDynamicGroupEvaluation` to `true` include View, Modify, and Notify attributes in profiles of the User, Group, and Organization Manager (when ACL for the attribute is defined based on group membership). When you use this parameter, the oblog.log file includes the following messages only if the corresponding group type is enabled in groupdbparams.xml:

- check for dynamic group membership
- check for nested group membership

> **Note:** Using the `TurnOffDynamicGroupEvaluation` parameter in the Identity Server groupdbparams.xml file, does not override IdentityXML calls that provide the same mechanism.

IdentityXML calls that provide the same mechanism are not overridden, as described in the *Oracle Access Manager Developer Guide*. These include:

- `checkDynamic`: Am I a member of a group; Is this person a member of a group
- `showDynamicGroups`: Get groups that a user is a member, owner, or administrator of; Get groups that I am a member, owner, or administrator of
- `showDynamicUserMembers`: View group members

You can use the following procedure to turn off dynamic group evaluation for attribute access control. Just set the `TurnOffDynamicGroupEvaluation` parameter in the groupdbparams.xml file to `true`. To restore evaluation of dynamic groups, just set the value to `false`.

To turn off dynamic group evaluation for the Identity Server

1. Open the following file in an editor:

   IdentiyServer_install_dir\identity\oblix\data\common\groupdbparams.xml

   Where *IdentityServer_install_dir* is the directory where the Identity Server was installed.

2. In groupdbparams.xml, set the value of the `TurnOffDynamicGroupEvaluation` parameter to `true`.
3. Restart the Identity Server.
4. Repeat for each Identity Server in your deployment.

For more information about parameters in the groupdbparams.xml file, see the *Oracle Access Manager Customization Guide*.

Use Nested Groups with Caution

An evaluation of nested groups requires expensive, repeated LDAP queries to determine the attributes of the groups' members. You may want to turn off nested group evaluation, or use this function selectively.

You can perform nested group evaluation selectively using IdentityXML and its related Web services call `viewGroupMembers`. This call can contain one of the following parameters:

- `showStaticUserMembers`
- `showDynamicUserMembers`
- `showNestedUserMembers`

See the *Oracle Access Manager Developer Guide* for details.

You can use the following procedure to turn off nested group evaluation for My Groups only, in the Identity System.

To turn off nested group evaluation from within the Identity System

1. In the Identity System Console, click Group Manager Configuration.
2. Click Configure Group Manager Options.
3. Deselect (uncheck) the options "Allow nested groups in My Groups pages" and "Show nested members of groups in the Manage Members page".

You can use the following procedure to improve performance for attribute access control through group evaluation. Whenever access to a resource is determined by a group, the membership to the group is evaluated in the following order: static, dynamic, and then nested. When you are not using nested groups, Oracle recommends that you turn off evaluation of that group type.

When you are not using nested groups, you can set the `TurnOffNestedGroupEvaluation` parameter in the groupdbparams.xml file to `true`. Functions that are impacted include View, Modify, and Notify attributes in profiles of the User, Group, and Organization Manager (when ACL for the attribute is defined based on group membership). When you use this parameter, the oblog.log file includes the following messages only if the corresponding group type is enabled in groupdbparams.xml:

- check for dynamic group membership
- check for nested group membership

> **Note:** Using the `TurnOffNestedGroupEvaluation` parameter in the Identity Server groupdbparams.xml file does not override IdentityXML calls that provide the same mechanism.

IdentityXML calls that provide the same mechanism is not overridden, as described in the *Oracle Access Manager Developer Guide*. These include:

- `checkNested:` Am I a member of a group; Is this person a member of a group
- `showNestedGroups`: Get groups that a user is a member, owner, or administrator of; Get groups that I am a member, owner, or administrator of
- `showNestedUserMembers`: View group members

You can use the following procedure to turn off nested group evaluation for attribute access control. Just set the `TurnOffNestedGroupEvaluation` parameter in the groupdbparams.xml file to `true`. To restore evaluation of nested groups, just set the value to `false`.

To turn off nested group evaluation for the Identity Server

1. Open the following file in an editor:

   IdentityServer_install_dir\identity\oblix\data\common\groupdbparams.xml

   Where *IdentityServer_install_dir* is the directory where the Identity Server was installed.

2. In groupdbparams.xml, set the value of the `TurnOffNestedGroupEvaluation` parameter to `true`.

3. Restart the Identity Server.

4. Repeat these steps for each Identity Server in the deployment.

For details about parameters in the groupdbparams.xml file, see the *Oracle Access Manager Customization Guide*.

Guidelines for Working with Large Static Groups

When working with groups in the Identity system, in general it is best to replace large static groups with dynamic groups. If you must use static groups, the following sections contain recommendations for using them:

- Exclude Group Membership Attributes from Panels and Search Results Tables
- Exclude Member Roles from Attribute Access Control Policies
- Performance Tuning for Evaluation of Large Static Groups

Exclude Group Membership Attributes from Panels and Search Results Tables

If you have large static groups, avoid displaying the member attribute on group profile panels, and avoid using the member attribute in search results tables.

Each time a user views a group profile page, Oracle Access Manager performs a number of evaluations to determine what members the user is permitted to view. When thousands of members are involved, it can take a long time to evaluate the user's permission to view each member.

If you remove the member attribute from the group profile panels, you can use the Manage Members page to handle membership operations, for example, search, adding members, deleting members, and so on. This page optimizes the management of large groups (defined as static groups with 1000 or more members), as opposed to defining the member semantic attribute as part of the group profile page. This significantly improves performance when managing large groups. For more information on configuring the user, group, and organization manager, see the *Oracle Access Manager Identity and Common Administration Guide*.

One caveat regarding removing the member attribute from the group profile panels is that the IdentityXML and Web service call `modifyGroup` cannot be used to update group membership. However, this call is not recommended because of the overhead it introduces. A preferable alternative is to use the following calls:

- `subscribeToGroup`
- `subscribeUserToGroup`
- `unsubscribeFromGroup`
- `unsubscribeUserFromGroup`

These calls do not require the member attribute to be configured on a group profile panel. These calls use the subscription policy for the group. These operations operate on the changed data only, and as a result they are faster operations for large groups than `modifyGroup`.

Exclude Member Roles from Attribute Access Control Policies

Adding a member role to an access control policy adds a large amount of overhead to the evaluation of the policy. This can affect response time in all areas where the policy is used.

Performance Tuning for Evaluation of Large Static Groups

For static groups that are particularly large, for example, groups with over 10,000 members, you may find that performance is affected. For situations where a static group has become too large, you have the option of using a different evaluation algorithm for the group.

If you modify the evaluation process for the group, you must make appropriate changes in your Identity System configuration to ensure that members of the group are still searched and evaluated as intended. These changes include the following:

- Members of this group and its subgroups are not displayed on the group profile page.
- If you search for members of the group, members in any subgroups of this group are not displayed.

  As a workaround, users can view the subgroup profile page and perform the search from the profile page of the subgroup.
- Subgroups of this group are no longer evaluated in an Identity System policy.

  For example, the subgroups and their members are not considered trustees in the following operations:

  – When evaluating read and write permissions for attributes

  – When defining a searchbase

  – When delegating administrative privileges

  – When adding workgroup participants

  As a workaround, you can include the subgroups directly in the policy.

To modify the evaluation of a large static group

1. Open the following file in an editor:

   Identity_Server_installdir\identity\oblix\apps\common\bin\globalparams.xml
2. In the globalparams.xml file, add the group DN to the `LargeStaticGroups` parameter.

   You can enter values for multiple groups using this parameter. The following is an example:

```
<ValList xmlns="http://www.oblix.com"
     ListName="LargeStaticGroups">
     <ValListMember Value="cn=testgroup1,o=mycompany,c=us"></ValListMember>
     <ValListMember Value="cn=testgroup2,o=mycompany,c=us"></ValListMember>
</ValList>
```

3. Save the file.
4. Restart the Identity Server.

5. If you have multiple Identity Servers, repeat this procedure on each server.

Tuning the Group Manager Application

The Group Manager is an application in the Identity System. Group size, especially large nested groups, can degrade the performance of operations on the Group Manager application pages. When possible, use dynamic groups instead of nested groups.

Three Group Manager pages can be tuned to optimize performance. The rest of this section discusses the following topics:

- Tuning the My Groups Page
- Tuning the View Members Page
- Tuning the Group Expansion Page

Tuning the My Groups Page

You can tune the performance of the My Groups page as follows.

To tune the performance of the My Groups page

1. In the Identity System Console, navigate to Group Manager Configuration, Group Manager Options.

   On this screen are several Boolean flags that you can set by clicking Modify. The *Oracle Access Manager Identity and Common Administration Guide* describes these settings. As an example of how these flags might influence the performance of your system, consider the setting labeled "Show nested groups". Showing nested groups can be a time-consuming operation, depending on the complexity of the group hierarchy. Setting this option to false limits group display to a simpler format, but can significantly improve the performance of the page.

2. To improve directory performance, index the following attributes, which are used in the My Groups profile:

   - Attributes configured with the ObSDynamicMember semantic type
   - Attributes configured as the ObSStaticMember semantic type
   - All user attributes used in group dynamic filters

3. Configure the optional group filter to further qualify results shown in the My Groups profile.

   The filter can be used to further qualify results under any searchbase. The filter would most likely be used in a FAT tree scenario where all entries are located under one container. The parameters that control the use of this filter are extra_group_filter and use_extra_group _filter_mygroups, found in the groupdbparams.xml catalog. The extra filter can be any LDAP filter and, in addition, may contain an Oblix rule substitution ($ $). When using a rule substitution, the substituted value comes from the user entry. This provides a means to link the values of attributes in a user entry with those in group entries. For example, a filter such as ou=ou would specify, that in order for a group entry to qualify, the *ou* of the group must be the same as that for the user.

4. Replace the default stylesheet, gsc_myprofile.xsl, with gsc_myprofile_simple.xsl.

   The default stylesheet for the My Groups page uses DHTML and layers to render the groups in a browseable tree format. If the My Groups page has to show many groups, the browser may take a long time to render the page. The gsc_myprofile_ simple.xsl stylesheet has a simpler, non-browseable interface and does not use DHTML and layers as much. Both stylesheets are located in the directories:

IdentityServer_install_dir/identity/oblix/lang/en-us/style0/ (stylesheet template)
IdentityServer_install_dir/identity/oblix/shared/ (wrapper stylesheet)

For details about stylesheets and styles, see the *Oracle Access Manager Customization Guide*.

To use gsc_myprofile_simple.xsl

1. Change the XML registry settings in the registry file:

   IdentityServer_install_dir/identity/oblix/apps/groupservcenter/bin/groupservcenterreg.xml

2. In this file, change the following line:

   ```
   <ObStyleSheet name="gsc_myprofile.xsl"/>
   ```

   to

   ```
   <ObStyleSheet name="gsc_myprofile_simple.xsl"/>
   ```

3. Then restart the Identity Server and Web server.

Tuning the View Members Page

You can tune the performance of the View Members page in the following ways:

- You can control the behavior of the View Members page using three Identity System Console options.

  These options can be turned on and off in the System Configuration, Group Manager Configuration, Group Manager Options page. See the *Oracle Access Manager Identity and Common Administration Guide* for information on these flags.

- You can constrain the search that can be done in the View Members page. Locate the groupMemberSearchStringMinimumLength parameter in the catalog:

  IdentityServer_install_dir/identity/oblix/apps/groupservcenter/bin/groupservcenterparams.xml

  This parameter controls the minimum number of characters that the user must type to be able to search for members of a group. Other than being restricted to group membership searches, its usage is identical to that of the searchStringMinimumLength parameter (see "Indexing and Search Constraints" on page 3-7 for details).

- You can index any user attributes used in group dynamic filters.

Tuning the Group Expansion Page

The following attributes are used in group expansion and should be indexed to improve performance:

- Any attributes configured with the ObSDynamicMember semantic type.
- The obgroupexpandeddynamic attribute of the oblixadvancedgroup object class.
- All user attributes used in the dynamic filters of the groups to be expanded.
- As discussed under see "Tuning the My Groups Page" on page 3-30, use the Set Searchbase feature to localize access to sub-domains appropriately.

  This may also improve the performance of the Group Expansion page.

The performance of the Group Expansion page may be improved by using the extra group filter feature, discussed under see "Tuning the My Groups Page" on page 3-30.

Tuning the User ID Cache

To tune the performance of group operations in the Identity System, you should also ensure that the cache for group IDs can accommodate the number of user entries in the directory. In general, the cache should be able to hold twice the number of entries that reside in the directory.

To tune the number of user entries in the user information cache

1. Open the following file in an editor:

   Identity_Server_Installdir\oblix\apps\common\bin\globalparams.xml

   Where *Identity_Server_Installdir* is the directory where the Identity Server was installed.

2. In globalparams.xml, set the value of the `UidInfoCache.maxNumElems` parameter to a value, in kilobytes, that is approximately twice the number of user entries in the directory server.

Tuning Workflows

Workflow performance can be tuned in several ways:

- Tuning workflowdbparams.xml
- Configuring Workflow Search Parameters.

Workflows are also mentioned in the sections on directory tuning and indexing. See "Storing Workflow Tickets in the Directory" on page 3-3 and "Indexing and Workflows" on page 3-6 for details.

Tuning workflowdbparams.xml

The workflowdbparams.xml file resides in the following location:

`IdentityServer_install_dir``/identity/oblix/data/common`

The following parameters can be tuned to assist with workflow performance. You must restart the Identity Server for changes to these parameters to take effect:

- **WfDefCacheMaxNoOfElements—**This parameter sets the size of the workflow definition cache. Set the value of this parameter to be higher than the number of defined workflows.
- **WfDefMaxNumStepDefFiltersPerSearch—**This parameter controls how many searches the Identity System performs on instance data. If users are participants in a large number of workflow steps, increasing the value of this parameter can reduce the number of times the directory is searched. To experiment with this parameter, increase its value in increments of 20. A higher number reduces searches to the directory, but increases the LDAP filter length. You can monitor the directory CPU utilization to determine an optimum value.

Configuring Workflow Search Parameters

The following guidelines are intended to help you tune workflow search performance:

- Check the number of search results that are returned per page. The default is 20. A higher number of results per page may cause slow performance.
- If workflow participants are specified as a filter, performance is slower. See if participants can be specified using a static group.

Tuning the Access System

This section describes ways to tune the performance of the Access Server, including:

- Configuring Password Validation by the Access Server
- Changing the Number of Request Queues and Threads
- Limiting the Number of Authorization Queries from WebGate
- Reducing Instability in the Access Server
- Securing AccessGate Clients
- Tuning the Handling of Groups in the Access System
- Tuning the LDAP Search Filter in the Policy Manager

Configuring Password Validation by the Access Server

By default, when Oracle Access Manager validates a user password as part of the validate_password plug-in, it passes the request to the user directory server. The directory server validates the password and returns the result to Oracle Access Manager. This operation is slow. In an environment with many authentications, it degrades Access Server performance.

You can control whether to use the Access Server or the directory server for password validation on a scheme-by-scheme basis.

Process overview: When using Access Server password validation

1. The first time a user's password is validated, the Access Server goes to the directory server for validation.

   If the password is valid, the Access Server caches an MD5 hash of the password. The Access Server never caches a clear text password.

2. The next time the user's password needs to be validated, the Access Server creates an MD5 hash of the supplied password and compares it to the hash of the cached password.

   - If the two hashes match, the user's password is considered valid.
   - If the two hashes don't match, the Access Server validates the password against the directory. If the directory validates the password, the Access Server hashes it and replaces the old hash in the cache.

The ObCredValidationByAS Parameter

To configure an authentication scheme so that the Access Server validates passwords, add the parameter ObCredValidationByAS with a value of true to the validate_password plug-in parameter list. To control the timeout of the cached password on a scheme-by-scheme basis, use the Time To Live parameter. The default value of the Time To Live parameter is 1800 seconds (30 minutes). To control the interval at which the Access Server goes to the directory for password validation, add the parameter obPwdHashTTL with a value equal to the number of seconds required.

The following is an example of the out-of-the-box validate_password plugin:

```
validate_password obCredentialPassword="password", obAnonUser="cn=anonymous,
o=Company, c=US"
```

The following is an example of validate_password configured to use Access Server password validation with the default Time To Live parameter:

```
validate_password obCredentialPassword="password", ,obCredValidationByAS="true",
obAnonUser="cn=anonymous,o=Company, c=US"
```

The following is an example of validate_password set to use the Access Server password validation with Time To Live set to 100 seconds:

```
validate_password obCredentialPassword="password", ,obCredValidationByAS="true",
obAnonUser="cn=anonymous,o=Company, c=US", obPwdHashTTL="100"
```

See Also: "About Cache Timeouts" on page 5-2

Changing the Number of Request Queues and Threads

The Access Server uses request queues to create a sequence of incoming requests that are processed by worker threads.

You can tune the performance of the Access Server by increasing the number of request queues from the default of 1. Multiple request queues can reduce contention between service threads and message threads. This can be particularly beneficial on multi-processor computers. See "To change the number of request queues" on page 3-36 for details. If you change the number of request queues, also change the number of threads per queue as recommended in "Estimating the Required Number of Threads and Queues" on page 3-35.

About Threads and Queues

The request queue holds requests from AccessGates until they are processed. By default, there is one request queue.

The Access Server uses three types of threads:

- Message threads

  Message threads accept new requests from AccessGates and append them to the request queue. The number of message threads is equal to the number of connections opened between the AccessGate and the Access Server.

- Service threads

  Service threads remove requests from the queue and process them. Each request queue has a fixed number of service threads. The number of service threads is configured in the Access System Console. See the *Oracle Access Manager Access Administration Guide* for details. By default, there are 60 service threads per request queue. The total number of service threads in an Access Server is equal to the number of request queues times the number of service threads per queue.

- Utility threads

  These perform housekeeping activities. There are usually between 20-40 utility threads, depending on the number of directory profiles configured for the Access Server, the number of file log writers defined, and so on.

Estimating the Current Number of Threads

To estimate the total number of threads, take the totals for each type of thread, as follows:

- Total message threads

  The netstat command enables you to determine the number of connections opened by AccessGates to the Access Server. Multiply this number by the number of message threads. For example, if there are 50 WebGates, each with 3 connections to an Access Server, there are 3 * 50 or 150 message threads.

- Total service threads

  The number of service threads is configured in the Access System Console. See the *Oracle Access Manager Access Administration Guide* for details.

- Total utility threads

  An average of 30 can be considered safe, using the guidelines in the previous paragraphs.

For example, if there are 50 WebGates, each with 3 connections to AccessGates, a value of 60 configured for the number of service threads in the Access System Console, and an estimated 30 utility threads, there are a total of 150 + 60 + 30 or 240 threads.

Estimating the Required Number of Threads and Queues

Having many queues with a relatively small number of threads per queue can cause problems if the requests come primarily on one queue. It may be helpful to configure more than the minimum number of threads per queue. You can control the number of worker threads from the command line or from the Access Server configuration page.

> **Note:** All command line utilities and tools must be run as the user who installed the product, as described in the *Oracle Access Manager Installation Guide*. Oracle recommends that you do not attempt to change ownership or permissions on files after installation.

If your Access Server handles more than 800 requests per second, you may need to change the number of request queues. A good rule of thumb:

- Set the number of queues equal to the number-of-peak-requests-per-second/800.
- Set the number threads per queue to 60/number-of-request-queues.

These recommendations are based on benchmark and performance tests. For example, if the peak request rate for an Access Server is expected to be 2000 requests per second, the number of queues should be 2000/800, or approximately 2. The number of threads per queue should be 60/2 = 30.

Avoid having too few or too many threads per queue. Four threads is adequate for benchmarking, but if the directory server is slow in responding you might need more. The optimal number for the total number of threads may be closer to 100 for a modest improvement in performance on a large, fast system. However, performance is not excessively sensitive to the number.

For example, an environment with 16 queues and 16 threads apiece (256 total threads) produced about 9,000 TPS during a lab test with directory access turned off. An environment with 16 queues and 4 threads apiece (64 total threads) produced about 9,400 TPS on the same configuration. For even the largest deployments, 8 queues should be enough, and 2-4 queues may be sufficient.

Note: The default queue size is 1. It is beneficial to reduce contention between service threads and message threads by modifying the "numQ" parameter in the Access Server globalparams.xml. This is particularly true on multi-processor computers. When the number of request queues is modified, service threads should also be changed (reduced) because each queue will have a set of service threads associated with it.

To change the number of request queues

1. When starting the Access Server from the command line, type the following:

   ```
   aaa_server -i install_dir -Q n
   ```

   where *n* is the number of request queues. The number of queues must be an integer to a maximum of 1024.

   When using Access Server service on Windows the -Q option must be specified as a startup parameter to the service. Alternatively, you can modify the following script to include this parameter:

   ```
   AccessServer_install_dir/access/oblix/apps/common/bin/start_access_server
   ```

2. Restart the Access Server for this parameter to take effect.

Limiting the Number of Authorization Queries from WebGate

When you define access policies for a policy domain, the WebGate by default queries the Access Server every time a user attempts to access resources in that domain. The more broad the policy domain, the more often the Access Server is queried. For example, if you configure root (/) as the policy domain in the Policy Manager, the WebGate contacts the Access Server every time someone tries to access a resource on the entire Web site.

To minimize the number of times the WebGate queries the Access Server, you can configure the `DenyOnNotProtected` flag. When set to true, `DenyOnNotProtected` denies access to all resources to which access is not explicitly allowed by a rule or policy. This can limit the number of times the WebGate queries the Access Server, and can improve performance for large or busy policy domains. See the *Oracle Access Manager Access Administration Guide* for details.

Reducing Instability in the Access Server

The following guidelines can reduce the risk of introducing instability to the Access Server when using API-Based plug-ins:

- Ensure that any API-based plug-ins that are deployed on the Access Server are thread-safe

  Both the Access Server and Identity Server are multithreaded. This recommendation also applies to Identity Event plug-ins.

- Ensure that all authorization and authentication API plug-ins are persistent, and improve performance by implementing connection pooling and caching
- Initialize all global and local variables used in the plug-in functions
- Ensure that all authorization and authentication plug-ins take input parameters from the Access Server in order to specify configuration information.

For more information, see "Plug-Ins" on page 3-51. For details about Oracle Access Manager APIs and plug-ins, see the *Oracle Access Manager Developer Guide*.

Securing AccessGate Clients

AccessGate client configuration includes a secret password between the Access Server and the AccessGate to prevent invalid clients from connecting to the Access Server. Oracle recommends that you implement SSL to encrypt the communication between the Access Server and AccessGate clients. In addition, treat the single sign-on token (typically the content of the ObSSOCookie) as a password, and do not provide it to external applications.

For more information, see the chapter on customizing access control with plug-ins in the *Oracle Access Manager Customization Guide*.

Tuning the Handling of Groups in the Access System

The way that you configure group-related features in the Access System can affect performance. By default, the Access Server checks for different types of group memberships when evaluating user membership in a group. Directory servers allow the following types of group membership:

- Static group membership

  In this type of group, each user is explicitly defined as a member.

- Dynamic group membership

  This type of membership is defined by an LDAP rule. Each user that satisfies this LDAP rule is a member of the group.

- Nested group membership

  A nested group consists of one or more static groups, dynamic groups, or both.

The Access Server evaluates group membership as follows:

- When you create an authorization scheme and assign it to one or more groups of users, for example, when you grant access to account data to all Finance group members.
- When you define an authorization or authentication action that contains the `obmygroups` attribute, the Access Server finds all the groups a user belongs to, and returns the list of groups in a cookie or header.

The following sections provide guidelines for group membership evaluation in the Access System.

- Using Dynamic Groups Instead of Static Groups
- Improving Performance During Group Search When Dynamic Groups Are Not Used
- Considerations for Nested Groups
- Considerations for ObMyGroups
- Improving Performance of ObMyGroups Evaluations
- Configuring the Access Server Group Cache Timeout and Maximum Elements

Using Dynamic Groups Instead of Static Groups

Dynamic group membership is evaluated automatically based on a filter and user attributes. In general, this makes dynamic groups easier to manage than static ones. Also, evaluation of dynamic group membership is an inexpensive operation, enabling a dynamic group to be as large as you want without degrading performance.

For more information on configuring user authentication, see the *Oracle Access Manager Access Administration Guide*. For details about improving performance when dynamic groups are not used, see "Improving Performance During Group Search When Dynamic Groups Are Not Used".

Improving Performance During Group Search When Dynamic Groups Are Not Used

By default, user membership is determined by group membership when access to a resource is protected using group-based authorization rules. The evaluation order is static, dynamic, and then nested groups. The same order is followed for obmygroups attribute value evaluation.

Group membership is evaluated during:

- Group-based authorization rules
- Authentication and authorization actions that use the `obmygroups` parameter
- Groups that are assigned to Allow Access and Deny Access conditions in authorization schemes
- Access Tester operations with the Policy Manager

Today, the Access Server (and Policy Manager when using the Access Tester) evaluates the group for membership as a type, only if that type is enabled. To improve performance during group evaluations when you do not use dynamic groups, or when you have dynamic groups but do not want to evaluate them, you can turn off dynamic group evaluation using the following procedure.

> **Note:** Oracle recommends that you use this procedure with caution, because there are access implications for end users.

By default, the Access Server checks for static, dynamic and nested group memberships to determine authorization. However, suppose you have an authorization policy based only on dynamic group membership or you configure the authentication action headerVar or cookie as obmygroups. If you set `TurnOffDynamicGroupEvaluation` to a value of `true`, when a user who is a member of a dynamic group requests a protected resource the Access Server skips dynamic group evaluation (as instructed) and the user receives a message stating that access is denied. In the case of ObMyGroups headerVar, the group name does not appear in the message.

> **See Also:** The chapter on parameters in the *Oracle Access Manager Customization Guide*

To turn off dynamic group evaluation for the Access System

1. Open the following file in an editor:

   AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *AccessServer_install_dir* is the directory where the Access Server was installed.

2. In globalparams.xml, set the `TurnOffDynamicGroupEvaluation` parameter with a value of `true`. For example:

   ```
   <SimpleList>
     <NameValPair ParamName="TurnOffDynamicGroupEvaluation" Value="true" />
   </SimpleList>
   ```

3. Restart the Access Server.
4. Repeat for each Access Server in your deployment.
5. **Access Tester**: Repeat these steps to set the value of the `TurnOffDynamicGroupEvaluation` parameter to `true` in the Policy Manager globalparams.xml file, restart the Policy Manager Web server, and repeat for each Policy Manager in your deployment:

   PolicyManager_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *PolicyManager_install_dir* is the directory where the Policy Manager was installed.

For details about ObMyGroups, see "Considerations for ObMyGroups" on page 3-40. For details about the globalparams.xml file, see the *Oracle Access Manager Customization Guide*.

Considerations for Nested Groups

Use nested groups with caution. By default, the Access Server checks for static, dynamic and nested group memberships to determine authorizations. Evaluation of nested group memberships is extremely expensive to evaluate. LDAP directory servers must perform repeated queries to determine the values for attributes of members of nested groups. If you are not using nested groups, disabling the nested group membership check improves performance.

You can use the following procedure to disable nested group evaluation for the Access System.

Note: Oracle recommends that you use this procedure with caution, because there are access implications for end users.

The default behavior is to check for nested groups. However, suppose you have an authorization policy based only on nested group membership or you configure the authentication action headerVar or cookie as ObMyGroups. If you set `TurnOffNestedGroupEvaluation=true`, when a user who is a member of a nested group requests a protected resource the Access Server skips nested group evaluation (as instructed) and the user receives a message stating that access is denied. In the case of ObMyGroups headerVar, the group name does not appear in the message.

See Also: The chapter on parameters in the *Oracle Access Manager Customization Guide*

To turn off nested group evaluation for the Access System

1. Open the following file in an editor:

   AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *AccessServer_install_dir* is the directory where the Access Server was installed.

2. In globalparams.xml, set the value of the `TurnOffNestedGroupEvaluation` parameter to `true`.
3. Restart the Access Server.
4. Repeat for each Access Server in your deployment.
5. **Access Tester**: Repeat these steps to set the value of the `TurnOffNestedGroupEvaluation` parameter to `true` in the Policy Manager globalparams.xml file, restart the Policy Manager Web server, and repeat for each Policy Manager in your deployment:

   PolicyManager_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *PolicyManager_install_dir* is the directory where the Policy Manager was installed

For more information, see the chapter on configuring user authentication in the *Oracle Access Manager Access Administration Guide*. For details about the globalparams.xml file, see the *Oracle Access Manager Customization Guide*.

Considerations for ObMyGroups

As explained in the *Oracle Access Manager Access Administration Guide*, you can configure the `obmygroups` parameter in an authentication or authorization rule. This is useful when integrating the Access System with other applications. This parameter looks up a user's group memberships in the directory. This provides role-based information for the user. The following are examples:

- You can define an application as a set of URLs whose access is provided to members of only a few groups in the directory.
- You can set the `obmygroups` parameter to supply group membership information to an application to prevent the application from having to query the directory directly or determine if a group is static, dynamic, or nested.
- You can set the `obmygroups` parameter to enable an application to customize navigation or appearance based on what groups the user belongs to.
- Specific menu items and functions can be based on group membership.

When you configure the `obmygroups` parameter, by default Oracle Access Manager searches for all group objects in the Access System searchbase and builds a user and group relationship cache in the Access Server. This cache is called the Group Query Cache. No Oracle Access Manager ACLs apply to the query. All groups that the user belongs to are supplied, whether or not the user has read permissions for the class attributes for these groups. Oracle Access Manager evaluates the entire Group Query Cache for each resource that is protected by a policy with an action that contains `obmygroups`. All groups in the cache must be checked to see if the user is in that group. This is an expensive lookup.

The Group Query Cache expires every ten minutes. Until Oracle Access Manager 10*g* (10.1.4.3), you could not tune the Access Server group cache timeout value and you could not set the maximum number of elements in the Access Server group cache. Use the following guidelines when specifying `obmygroups` in an authorization action and to configure the Access Server group caches:

- Always configure actions that use `obmygroups` using an LDAP filter, for example:

  ```
  obmygroups:ldap:///o=company,c=us??sub?(group_type=role)
  ```

- In general, it is best to specify `obmygroups` in an authentication rule rather than an authorization action.

If possible, have the action be a cookie so that the data is available to other applications under the same Web server without incurring an additional toll.

- When you use `obmygroups` in an authorization rule, limit its use to as few resources (URLs) as practical.
- If you are not using dynamic groups, turn off evaluation for dynamic groups.

  See "To turn off dynamic group evaluation for the Access System" on page 3-38 for details.
- If you are not using nested groups, turn off evaluation for nested groups.

  See "To turn off nested group evaluation for the Access System" on page 3-39 for details.
- To improve performance during evaluations involving ObMyGroups for static, dynamic, and nested groups, you can use a new algorithm introduced in Oracle Access Manager 10*g* (10.1.4.3).

  See "Improving Performance of ObMyGroups Evaluations" on page 3-41.
- To configure details for the Access Server group caches

  See "Configuring the Access Server Group Cache Timeout and Maximum Elements" on page 3-43.

For more information about ObMyGroups, see the chapter on configuring user authentication in the *Oracle Access Manager Access Administration Guide*.

Note: You may find that using an IdentityXML call, `userGroupsProfile`, is a less resource-intensive method for returning the groups that are associated with a user. See the chapter on configuring IdentityXML parameters in the *Oracle Access Manager Developer Guide* for details

See Also: "About Cache Timeouts" on page 5-2

Improving Performance of ObMyGroups Evaluations

When you configure the `obmygroups` parameter in an authentication or authorization rule, by default Oracle Access Manager searches for all group objects in the Access System searchbase and builds a user and group relationship cache in the Access Server. Depending on the number of groups being searched, ObMyGroups processing might take a significant amount of time.

Oracle Access Manager 10*g* (10.1.4.3) provides performance enhancements for evaluations that include ObMyGroups. For instance, large groups can contain thousands of static members. In earlier releases the Access Server would read the whole group entry, including all attributes. Further, the Access Server would cache the entry during evaluation of obmygroups and group-based authorization. Today, however, retrieving all attributes except the desired attribute (uniquemember, groupfilter, and so on) depends on the LDAP query. Also, caching the whole entry has been disabled; only the attributes in the LDAP query are cached.

In addition, with 10*g* (10.1.4.3) you can configure a new algorithm to be used during group evaluation involving ObMyGroups: `TurnOffNewAlgorithmForObmyGroups`. This algorithm works equally when you have static, dynamic, and nested groups.

There are three scenarios where the new parameter results in performance improvement:

- Improved evaluation of ObMyGroups
- Improved evaluation of nested group membership
- Circular group evaluation

The new algorithm is enabled by default. In all cases, the new algorithm uses a bottom-up approach that first checks the group cache to see if the user is already evaluated. Following are details of performance improvements in each specific scenario.

Improved Evaluation of ObMyGroups: Originally, the Access Server shared common logic for evaluating ObMyGroups and group-based authorization. 10*g* (10.1.4.3) optimizes the group membership evaluation process for ObMyGroups HeaderVar. If you set `TurnOffDynamicGroupEvaluation` to `true`, dynamic group evaluation is skipped; if this setting is `false`, dynamic group evaluation proceeds.

Improved Evaluation of Nested Group Membership: 10*g* (10.1.4.3) optimizes group membership evaluation process for ObMyGroups. You can set `TurnOffNestedGroupEvaluation` to `true` to skip nested group evaluation; if this setting is `false`, nested group evaluation proceed.

Nested Circular Group Evaluation: Using the earlier 10g algorithm, the Access Server checked one level of circular dependency during nested group evaluation to determine if a group contained itself as a unique member. During nested circular group evaluation, the Access Server could become unresponsive. The 10*g* (10.1.4.3) algorithm streamlines processing and eliminates issues that could cause the Access Server to become unresponsive.

The `TurnOffNewAlgorithmForObmyGroups` in the Access Server globalparams.xml file, can be set as follows:

Table 3–2 Values for `TurnOffNewAlgorithmForObmyGroups` ***Parameter***

Value	Description
true	Use the original algorithm, available with earlier releases
false	The default value is false so that the new algorithm is used
No parameter	The default value is presumed and the new algorithm is used

To reconfigure the `TurnOffNewAlgorithmForObmyGroups` parameter

1. Open the following file in an editor:

   AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *AccessServer_install_dir* is the directory where the Access Server was installed.

2. **Use the Original 10g Algorithm**: Set the `TurnOffNewAlgorithmForObmyGroups` parameter value to `true` to turn off the new algorithm.

3. **Restore the 10*g* (10.1.4.3) Default Algorithm**: If you had used the earlier algorithm and want to restore the 10*g* (10.1.4.3) algorithm, set the value of `TurnOffNewAlgorithmForObmyGroups` parameter to `false` (or confirm that there is no parameter). For example:

```
<SimpleList>
  <NameValPair ParamName="TurnOffNewAlgorithmForObmyGroups" Value="false" />
```

```
</SimpleList>
```

4. Restart the Access Server.
5. Repeat for each Access Server in your deployment.
6. To further improve performance with the 10*g* (10.1.4.3) algorithm for nested groups, add the following parameter and value to the globalparams.xml file:

```
<SimpleList>
  <NameValPair ParamName="NestedQueryLDAPFilterSize" Value="10" />
</SimpleList>
```

Note: 10 is the default value for `NestedQueryLDAPFilterSize`. Choose a value using these guidelines:

- Large nested group data: lower value than the default value.
- Small nested group data: higher value than the default value.

Configuring the Access Server Group Cache Timeout and Maximum Elements

For efficient performance, the Access Server stores group information in several caches during ObMyGroups and group membership evaluation. However stale data needs to flushed periodically. There are no GUI items in the System Console to flush the Access Server group caches. Until Oracle Access Manager 10*g* (10.1.4.3), you could not tune the Access Server group cache timeout value and you could not set the maximum number of elements in the Access Server group cache.

Default Access Server group cache values are shown in Table 3–3.

Table 3–3 Default Access Server Group Cache Values

Cache Name	Default Timeout	Default Maximum User Elements
User Rule Cache	2400 seconds	100,000 elements
User Group Cache	2400 seconds Note: There currently is no default, but you can consider this as 2400 seconds.	10,000 elements
Group Definition Cache	2400 seconds	100,000 elements
Group Query Cache	600 seconds	100,000 elements

Today, however, two parameters are provided that enable you to set the cache timeout value and define the maximum number of elements in the group cache. Oracle Access Manager uses some internal caches for group evaluations, which also uses the new parameters for the group cache Timeout and maximum element. For more information, see the following topics:

- Defining the Access Server Group Cache Timeout
- Defining the Access Server Group Cache Element Size

Defining the Access Server Group Cache Timeout The `GroupCacheTimeout` parameter enables you to specify the amount of time an element remains valid in the Access Server group cache. This parameter must be added to the Access Server globalparams.xml file (or the Policy Manager file if you are using the Access Tester).

The original and default behavior is defined in Table 3–3.

The possible values for the `GroupCacheTimeout` parameter are shown in Table 3–4.

Table 3–4 GroupCacheTimeout Values

Value	Description
Positive Integer	Specifies the timeout in seconds
0	Specifies that the element never times out
-1	The default value, which ensures backward compatibility
Negative integers, or no parameter, or invalid values	Disables this parameter and restores the default value for each cache

To set the GroupCacheTimeout

1. Open the Access Server globalparams.xml file in an editor:

   AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *AccessServer_install_dir* is the directory where the Access Server was installed.

2. In globalparams.xml, set the value of the `GroupCacheTimeout` parameter using Table 3–4 as a guide.

3. Restart the Access Server.

4. Repeat for each Access Server in your deployment.

5. **Access Tester**: Repeat these steps to set the value of the `GroupCacheTimeout` parameter in the Policy Manager globalparams.xml file, restart the Policy Manager Web server, and repeat for each Policy Manager in your deployment:

   PolicyManager_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *PolicyManager_install_dir* is the directory where the Policy Manager was installed.

6. If you notice any degradation in performance with the new values, restore the default values as shown in Table 3–3.

Defining the Access Server Group Cache Element Size

The `GroupCacheMaxElement` parameter specifies the maximum number of elements that can be stored in the Access Server group cache.

The original and default behavior is defined in Table 3–3.

You use a positive integer value to specify a maximum number of elements to be stored in the cache. The possible values for the `GroupCacheMaxElement` parameter are shown in Table 3–5.

Table 3–5 GroupCacheMaxElement Values

Value	Description
Positive Integer	Specifies the maximum number of elements
0	Specifies an infinite number of elements can be stored
-2	The default value (ensures backward compatibility)
No parameter, or invalid values	Disables this parameter and restores the default value for each cache

To set the GroupCacheMaxElement

1. Open the Access Server globalparams.xml file in an editor:

   AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *AccessServer_install_dir* is the directory where the Access Server was installed.

2. In globalparams.xml, set the value of the `GroupCacheMaxElement` parameter using Table 3–5 as a guide.

3. Restart the Access Server.

4. Repeat for each Access Server in your deployment.

5. **Access Tester**: Repeat these steps to set the value of the `GroupCacheMaxElement` parameter in the Policy Manager globalparams.xml file, restart the Policy Manager Web server, and repeat for each Policy Manager in your deployment:

   PolicyManager_install_dir\access\oblix\apps\common\bin\globalparams.xml

   Where *PolicyManager_install_dir* is the directory where the Policy Manager was installed.

6. If you notice any degradation in performance with the new values, restore the default values as shown in Table 3–3.

Tuning the LDAP Search Filter in the Policy Manager

You can configure a parameter named `ldapFilterSizeLimitInBytes` in the globalparams.xml file for the Policy Manager. This parameter controls the size of the LDAP search filter. The default value is 1024 (1Kb) if you do not set the value explicitly in globalparams.xml, or if it is set to anything lower than 1.

See the appendix on parameter files in the *Oracle Access Manager Customization Guide* for details.

Tuning the Caches

In Oracle Access Manager, you can tune a number of caches to enhance performance. For more information, see the following topics:

- Tuning the Policy Cache
- User Cache Tuning
- Tuning the URL Prefix Cache
- WebGate Cache Tuning
- Sizing the Maximum Elements in WebGate Cache
- Tuning the Internal DBAgent Cache
- Sizing the Maximum Elements in WebGate Cache

For information on enhanced cache flush operations, see Chapter 5.

Tuning the Policy Cache

A policy cache contains information about policy domains, excluding URLs, policy descriptions, and display names. A policy cache element contains the following:

- Rules

- Actions associated with rules
- Filters, groups, and roles associated with rules
- Policy conditions for rules
- All policy domains
- All policies
- All authentication schemes

You tune the policy cache by setting the Maximum Elements in Policy Cache and Policy Cache Timeout parameters.

> **Note:** These parameters are also documented in the *Oracle Access Manager Access Administration Guide.*

Calculating Maximum Elements in a Policy Cache

To estimate the requirements, calculate the total number of authorization rules in all policy domains and policies. To do this, search the Oblix directory tree using the following filter:

```
"(objectclass=oblixpolicyrule)"
```

When determining the number of elements in a policy cache, be sure to account for future growth of the number of rules.

Calculating Memory Requirements for the Policy Cache Elements

To calculate the memory requirements for the Access Server's policy cache elements, check the memory requirements of the directory used to store the policies.

This value is roughly the value you should provide on the Access Server.

Calculating Policy Cache Timeout

When a policy domain, rule, or policy is created or modified, the administrator has the ability on each page to Update Cache. When these pages are saved, this forces an immediate flush of the relevant policy cache, ensuring that the change takes effect immediately.

You can have the caches time out on a regular basis as a security precaution or to clean up if the administrator does not press the Update Cache button during a policy change. The *Oracle Access Manager Access Administration Guide* discusses automatic flushing of the Access System cache.

See Also: "About Cache Timeouts" on page 5-2

User Cache Tuning

The tuning parameters available for this are Maximum Elements in User Cache and User Cache Timeout.

Calculating the User Cache Timeout

The user cache contains all user attributes and values used in audit and HTTP actions. You need to determine the shortest time acceptable for not changing bulk user attribute values. If the value is too large, values that are considered important from a risk-management perspective may not make their way into Oracle Access Manager until the cache is flushed. This can be a problem if the administrator forgets to update the cache after making changes. The administrator, delegated identity administrator, or user may change a user value in the directory and think it has been implemented when, in fact, there will be a delay until the value in the cache is flushed.

On the other hand, avoid setting the timeout or the cache size so small that data is flushed while a user is accessing a site, since this forces another trip to the directory.

To estimate a reasonable interval for updating user values, you can track average session times over a 24-hour period. The User Cache Timeout value can probably be set to this value or slightly higher than this value.

See Also: "About Cache Timeouts" on page 5-2

Calculating Maximum Elements in the User Cache

If you know the value for User Cache Timeout, you next measure the number of users who access resources during this time interval.

When you know the maximum number of concurrent users during the time interval, the number represents the maximum number of cache elements. This is because each element contains a user DN and their attributes and values used in the audit logs and HTTP actions.

Calculating Memory Requirements for User Caches

If you know the user cache timeout and the maximum number of elements in the user cache, you can calculate hardware requirements.

To begin, calculate the requirements per cache element. An element in the user cache refers to all user attributes and their values used in all the rules. The formula for this is:

```
cache element size = size of DN + size of all attribute values to be cached + 50
bytes for overhead
```

You can now calculate total memory requirements as follows:

```
Max. elements in user cache memory requirements = Max. elements x cache element
size
```

Tuning the URL Prefix Cache

The URL prefix cache sets the interval for flushing a URL prefix based on the value of the URL Prefix Reload Period in the Access Server configuration page. The URL prefix reload period is an additional safeguard that provides automatic flushing. If a URL prefix is added to a policy domain or policy, the administrator can click the Update Cache button to force an automatic cache flush.

The URL prefix reload period specifies the amount of time (in seconds) that elapses before URLs are reloaded from the directory server. In the Access Server configuration page, `URL Prefix Reload Period (seconds)` field, enter a number representing the frequency with which new URLs are recognized by this Access Server (the time between automatic cache flushes).

The default value for the URL prefix reload period is 7200 seconds (or two hours). For example, to flush the cache every 10 minutes, enter 600 in the `URL Prefix Reload Period (seconds)` field. 600 seconds = 10 minutes. In this case, URLs are reloaded from the directory server every 10 minutes. This can be helpful in cases where a particular URL Prefix cache flush request did not reach an Access Server.

WebGate Cache Tuning

WebGate caches information on authentication and on whether or not a resource is protected. WebGate cache tuning refers to the total number of unique URLs expected over the timeout interval.

The chances of an authentication scheme changing quickly are very low. The chances of a URL prefix being changed or added are low. If an administrator adds, changes, or deletes a URL prefix, the pages contain an Update Cache button that can be used to force an immediate cache flush.

The default value for the WebGate cache timeout is zero (0). This means that the cache is not automatically updated and is flushed only when the administrator clicks the Update Cache button. If you are not comfortable with the default, choose an appropriate interval before the URL is automatically flushed from the cache.

See Also: "About Cache Timeouts" on page 5-2

Sizing the Maximum Elements in WebGate Cache

WebGate can cache URLs to avoid trips to the Access Server or directory server when determining if a URL requires protection. The default value for Maximum Elements is 100,000. To estimate an appropriate value for this parameter, examine the number of URLs that the Web server is protecting and the HTTP operations associated with them.

Web browsers place an internal limit of 4096 bytes on URLs. Most URLs are smaller than 4096 bytes. You may want to determine upper limits for the cache and choose the same size or a smaller size than 4096, depending on what you believe is an average URL size.

To calculate memory requirements per cache element:

```
memory per cache element = URL (4096 bytes) + overhead (128 bytes) = 4224 bytes
```

To calculate the memory requirements for maximum cache elements:

```
memory required = 100,000 elements x 4224 bytes/element = 422,400,000 bytes = 422
MB of memory
```

Memory requirements may be smaller if the maximum number of elements is downsized according to the needs of the Web server that the WebGate is on and the average size of the URLs.

Tuning the Internal DBAgent Cache

Oracle Access Manager maintains an internal, short-lived DBAgent cache. It contains the entire list of attributes of all the structural and auxiliary object classes read when the Identity Server retrieves information from the directory server during view and modify profile operations from the User Manager, Group Manager, or Organization Manager.

Caching the values of bulky attributes can impact performance. Unlike the OSD cache, it is not possible to directly access the DBAgent cache. However, you can improve performance by tuning the DBAgent cache to disable the reading of unwanted attributes that meet the following conditions:

- Attribute with a bulky value (description, for example)
- Attribute that is not needed every time (no need to view or modify the attribute)

You can eliminate the reading and caching of unwanted attribute values during view and modify profile operations performed in a browser window by creating a list of specific attributes in the Identity Server globalparams.xml file located in:

IdentityServer_install_dir/identity/oblix/apps/common/bin/globalparams.xml

The list (known as a negative list because it eliminates stated attributes) is associated with the `negativeListForEntityAttributes` parameter. The list identifies the specific attributes that are not read or cached during view and modify profile operations from a browser. You can view the profile page without the values of the attributes that appear in the negative list. The directory server log shows only the attributes that are not on the list.

Your list might look something like the following example:

```
 <ValList
       xmlns="http://www.oblix.com"
       ListName="negativeListForEntityAttributes">
       <ValListMember
           Value="oblocationtitle"></ValListMember>
       <ValListMember
           Value="obdescription"></ValListMember>
</ValList>
```

Oracle recommends that you do not include attributes that might be used for evaluation in the negative list. If included, such entries might result in evaluation failure as a result of misconfiguration. For example, do not include such attributes as:

- obgroupsubscriptiontype
- obgroupdynamicfilter
- objectclass
- obgrouptype
- and the like (this is not an exhaustive list)

The negativeListForEntityAttributes parameter is employed during READ LDAP operations to retrieve partial entries (not fetching attribute in a negative lists) and then cache them in the DBAgent cache. However, this applies to only browser-based requests.

> **Note:** Attributes in the negative list are never read from the directory server during view and modify profile operations from the User Manager, Group Manager, or Organization Manager when you are using a browser. However, IdentityXML reads and caches values of all attributes even if some appear in a negative list.

If a request comes from IdentityXML, all READ LDAP operations that are performed as part of the request will ignore attributes in the negative list and will read and store values for ALL attributes in the DBAgent Cache.

By default, there is no list, which means that no attributes are eliminated. To eliminate unwanted attributes, you must explicitly create the list. You can add this at the end of the globalparams.xml file.

To create a negative list to eliminate attributes from view and modify panel operations

1. Locate the globalparams.xml file in the following path:

IdentityServer_install_dir/identity/oblix/apps/common/bin/globalparams.xml

2. Open the file in an editor and add a list of attributes that you want to eliminate at the end of the file. For example:

```
  <ValList
       xmlns="http://www.oblix.com"
        ListName="negativeListForEntityAttributes">
        <ValListMember
            Value="oblocationtitle"></ValListMember>
        <ValListMember
            Value="obdescription"></ValListMember>
 </ValList>
```

3. Save the file.

4. Restart the Identity Server.

Tuning Your Network

The performance of the overall network, or network latency, is a major factor in the performance of the system. A reduction in network latency is reflected in the performance of Oracle Access Manager.

It is not necessary, and not even desirable, for you to deploy load balancers between Oracle Access Manager components, even if you use load balancers for other application deployments. Avoid placing load balancers between an Access or Identity Server and any of the following components:

- WebGates or AccessGates
- WebPass
- Directory servers

Oracle Access Manager uses the LDAP protocol to perform update operations, and it provides keep alive, failover, and failback functionality to handle LDAP and network outages, replication, and related activities. The built-in features of Oracle Access Manager are often the same or better than similar features provided by a load balancer.

In some cases, you can negatively affect the performance of Oracle Access Manager by placing a load balancer between its components. For example, a load balancer may terminate a connection that it interprets as idle without triggering a response that Oracle Access Manager can detect and adjust to. This can cause outages.

The following are guidelines for tuning your network:

- You can consider adding an SSL accelerator or load balancer outside of the Oracle Access Manager system to improve the performance of your network.

  Deploying a load balancer in front of the Web servers or application servers is a best practice for increasing availability and performance of Web-based applications, including Oracle Access Manager. However, load balancers are not recommended between the Oracle Access Manager components themselves.

- You may want to set the DenyOnNotProtected flag to True in the WebGate configuration page (or in the WebGateStatic.lst file for pre-10.1.4 versions of Oracle Access Manager).

  This reduces trips between the Access Server and the directory. The DenyOnNotProtected flag forces the WebGate check its own cache to determine if a URL is protected.

- Reduce the latency between devices in high-traffic parts of the network.

  You can place the Identity and Access servers closer to client applications than to the directory. During normal operations there is more traffic between WebGates and Access Servers than between Access Servers and the directory. Similarly, there is more traffic between WebPass instances and Identity Servers than between Identity Servers and the directory. These servers have the greatest interaction with the directory at startup. After that time, much of the configuration information that these servers need can be read from their caches. One exception is if the administrators are performing system configuration, for example, if they are defining and testing workflows. In this case, placing the Access and Identity servers in the hosting site can help the response time for the administrators.

Be Sure Your Computers Are Working Properly

If you experience performance issues, you may want to check system CPU usage. For example, if a domain controller is not working well, performance can suffer.

Resource-Intensive Operations

Resource-intensive operations include login forms, password management, and plug-ins.

The following are issues to examine to optimize performance.

Time to Process Calls to Various Components

For performance tuning purposes, you can log the time it takes to process calls to external components. For example, you may want to know what Identity Servers are processing the most requests from the most WebPass instances.

Use the information logs to identify components that are processing a heavier load or are taking a particularly long time to service requests. See the chapter on logging in the *Oracle Access Manager Identity and Common Administration Guide* for details.

Login Forms

Login forms increase the overall load on Web servers. This may mean that you need more Web servers, depending on the size of the server and the form content. Dynamic contents and graphics adversely affect performance. An extremely high number of logins can cause network latency.

Password Management

Password management causes many directory write operations. When you implement password management, be aware that there are performance implications if you expect it to be used frequently.

Plug-Ins

Oracle Access Manager plug-ins and .exe files can affect performance. When you develop customizations for Oracle Access Manager, be aware of the impact of these customizations.

To minimize performance impact:

- When creating an action for the Identity Event API, because an action incurs overhead, identify all the possible events to attach the action to and choose the least frequently used one that yields the desired result.
- Evaluate the sequence in which actions are executed.

  For example, suppose that you develop an .exe file to send an email message through the SMTP server when a user performs a particular action. Depending on how you write this program, the Identity Server may be unable to perform further actions until it receives a reply from the SMTP server. If it is the case that the SMTP server is down, there may be a large impact on performance.
- When comparing EXEs and LIBs, note that LIB actions execute quickly because they are binary code modules compiled from C or C++ source code. However, their perceived speed depends on the function they perform.
- Avoid EXEC-type plug-ins, which spawn external processes.

  For example, if you want an Identity Event API plug-in to trigger other events in the Identity Server using IdentityXML, ensure that the request goes to an Identity Server instance other than the one triggering the Event API plug-in to balance the system load. Also, consider Identity Event plug-ins that use C/C++ as shared objects (.so files) for performance and stability reasons
- Ensure that all Identity Event plug-ins are thread-safe.
- Identity Event API plug-ins are not persistent, so you must initialize all global and local variables used in the plug-in functions.
- Use a single library for multiple event plug-in functions to minimize runtime image size.
- Ensure that the plug-in supports using a configuration file to alter and adapt its operation in case of requirement changes.

For more information about developing plug-ins and using Oracle Access Manager APIs, see the *Oracle Access Manager Developer Guide*.

4 Failover and Load Balancing

Failover and load balancing are vital for Oracle Access Manager availability and performance. Load balancing distributes request processing across multiple servers. Failover redirects requests to alternate servers if the originally requested server is unavailable or too slow.

This chapter discusses the following topics:

About Load Balancing with Oracle Access Manager

Oracle Access Manager uses load balancing to maximize performance by distributing server requests across multiple physical servers. You configure load balancing among the components listed below.

Load balancing between Oracle Access Manager Web components and servers includes:

- WebPass requests to one or more Identity Servers
- WebGate requests to one or more Access Servers

  You cannot use hardware load balancers to load balance requests from WebPass and/or WebGates to the Identity and/or Access Servers because the hardware load balancers do not understand the proprietary Oracle Access Manager protocols. Hardware load balancers can load balance only at the connection level, not the request level. The connections to the Oracle Access Manager Servers are persistent connections.

Load balancing among Oracle Access Manager Servers and directory servers includes:

- Identity Server to one or more directory servers
- Access Server to one or more directory servers

About Load Balancing of LDAP Data

Oracle Access Manager supports multi-master LDAP environments. Failover and load balancing are supported for user and policy data. However, failover and load balancing is not supported for configuration data. Configuration data includes system configuration, attribute definition, attribute access controls, workflow definition, and workflow instance data.

Before you configure load balancing for LDAP writes of user and group data, note that LDAP replication is likely to produce undesirable behavior. An update to one instance may take some time to be reflected on the another instance. This limitation is inherent to all LDAP directory services since LDAP replication does not guarantee transaction integrity.

For user and group data, clustering or segmenting may be useful for distributing the load for read and write operations. This is particularly the case if separate user populations exist in separate branches of the DIT, for example, if one server stores partner data under the ou=partners branch, and another server stores supplier data under the ou=suppliers branch. In cases like this, you can maintain each set of data on a different primary LDAP server for read and write operations. Also, for availability purposes, each server cluster can fail over to the other LDAP servers in a cross-over fashion. This load-balances read and write operations across LDAP multi-masters.

Some LDAP servers, for example Oracle Internet Directory, provide true load balancing capability for reads and writes. These servers guarantee immediate availability when an update occurs on any of the multi-mastered LDAP servers that are configured in this fashion. When using these types of servers, you can configure Oracle Access Manager to provide load-balancing for read and write operations for user and group data.

Oracle Access Manager does not support round robin write operations on either the user and group directories, nor the configuration branch containing Oracle Access Manager meta-data. Data corruption could result when WebGates and AccessGates are found in a round robin configuration with various Access Servers even if each Access Server points to a single, but different, instance of the configuration data hosted on separate, but multi-mastered directories. This is particularly important when Policy Management is turned on and the automatic cache flush is enabled on the Identity Server (or both). Every cache update triggers at least one write operation in the Policy Manager configuration data.

Configuring Load Balancing for Web Components

Oracle Access Manager supports both simple round-robin and weighted round-robin load balancing of requests from its Web components (WebPass and WebGate) to their associated servers (Identity and Access). You configure load balancing of Web component requests to Oracle Access Manager Servers from the perspective of the web component using two key fields:

- **Maximum Connections**: The maximum connections you want opened for a given instance of the Web component. This is the total number of live connections you want established to one or more primary Oracle Access Manager Servers at any given time. If the Web component cannot establish a connection to one of its primary servers, it tries to establish a connection to a secondary server.

- **Initial Connections**: The initial connections from the Web component to its associated Oracle Access Manager Servers. This applies to both primary and secondary Oracle Access Manager Servers.

There are several procedures in this section, to use depending upon your environment:

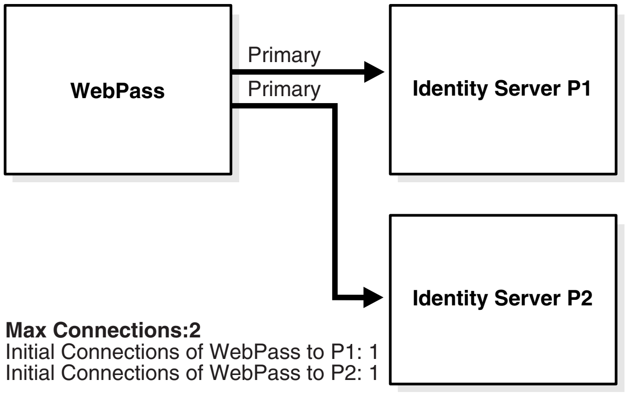
- Configuring Simple Round-Robin Load Balancing
- Configuring Weighted Round-Robin Load Balancing

Configuring Simple Round-Robin Load Balancing

Configuring simple round-robin load balancing of Web component requests means that a Web component opens a single connection to each of its associated primary Oracle Access Manager Servers in the order they are listed, and distributes the requests evenly among them.

For example, assume that you have a single WebPass and two primary Identity Servers as shown in Figure 4–1. In this case, WebPass opens a connection to each Identity Server in the order they are listed. WebPass sends request1 to Identity Server P1, request2 to Identity Server P2, request 3 to Identity Server P1 and so on.

Figure 4–1 Simple Round-Robin Load Balancing of Web Component Requests


WebPass
Primary
Primary
Identity Server P1
Identity Server P2
Max Connections:2
Initial Connections of WebPass to P1: 1
Initial Connections of WebPass to P2: 1

The following procedure guides as you configure a simple round-robin load balancing configuration. The maximum connections in this case is the total number of live connections you want established to one or more primary Identity or Access Servers at any given time. If the Web component cannot establish a connection to one of its primary servers, it tries to establish a connection to a secondary server. The initial connections apply to both primary and secondary Oracle Access Manager Servers.

To configure simple round-robin load balancing

1. Access the Web component configuration whose requests you want to load balance.

   For example:

   - From the Identity System Console, select System Configuration, WebPass.
   - From the Access System Console, select Access System Configuration, AccessGate Configuration.

   For more information about Web component configuration, see the *Oracle Access Manager Identity and Common Administration Guide* and *Oracle Access Manager Access Administration Guide*.

2. Enter the Maximum Connections you want opened for this WebPass or WebGate.
3. Leave the Initial Connections to each associated Oracle Access Manager Server at the default, which is 1.

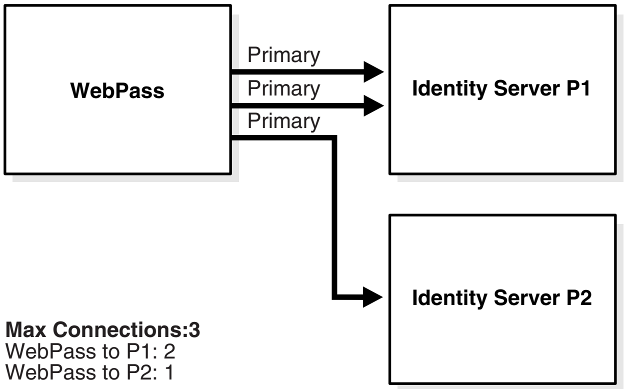
Configuring Weighted Round-Robin Load Balancing

You may want to configure weighted round-robin load balancing of Web component requests if your Oracle Access Manager Servers have varying performance capacities. The primary difference when configuring weighted load balancing is that you adjust the initial connections you want established to each server.

Figure 4–2 provides an example. Assume you have two primary Identity Servers as shown in Figure 4–2. However, Identity Server P1 can handle additional load. Then you may want to configure WebPass to open two connections to Identity Server P1, and one connection to Identity Server P2. The Maximum Connections for that WebPass would be 3: Identity Server P1, request2 to Identity Server P2, request 3 to Identity Server P1 and so on.

> **Note:** When you associate an AccessGate with an Access Server cluster, Oracle Access Manager automatically configures the number of connections between the AccessGate and all the Access Servers in the cluster based on the maximum number of connections that is specified for the cluster. Load balancing is dynamically configured and Oracle Access Manager ensures that the AccessGate routes requests to the most lightly loaded Access Servers in the cluster.

Figure 4–2 Weighted Load Balancing Layout Using Two Servers and no Failover



Here as well, your maximum connections specification is the total number of live connections you want established to one or more primary Identity Servers at any given time. If WebPass cannot establish a connection to one of its primary servers, it tries to establish a connection to another primary server.

Again, initial connections applies to both primary and secondary servers. When you associate an AccessGate with an Access Server cluster, Oracle Access Manager automatically configures the number of connections between the AccessGate and all the Access Servers in the cluster based on the maximum number of connections specified for the cluster. Load balancing is dynamically configured and Oracle Access Manager ensures that the AccessGate routes requests to the most lightly loaded Access Servers in the cluster.

To configure weighted round-robin load balancing of Web component requests

1. Access the Web component where you configure load balancing.

   For example:

   - From the Identity System Console, select System Configuration, WebPass.
   - From the Access System Console, select Access System Configuration, AccessGate Configuration.

2. Enter the Maximum Connections you want opened for a given WebPass.
3. Enter the Initial Connections to each associated Oracle Access Manager server to reflect that server's capacity.

Configuring Load Balancing Among Oracle Access Manager and Directory Servers

Simple round-robin load balancing of Oracle Access Manager server requests to two or more directory servers is supported.

> **Note:** Oracle Access Manager generally does not support load balancing for configuration data because many functions are dependent on data carried over from the previous request. In a load balanced environment, this data may not yet be available to the replicated server

The instructions for configuring load balancing for directory servers vary depending on the type of component (Identity Server, Access Server, Policy Manager), and whether you are configuring load balancing for user data or configuration data. See Table 4–1 for a summary of data store types and operations.

> **Note:** Oracle Access Manager does support a hardware load balancer for LDAP communication with Oracle Access Manager components. Rules and constraints described in this chapter apply to Oracle Access Manager internal load balancing functionally as well as to a hardware load balancer. For more information, see Note 793152.1 on My Oracle Support (formerly MetaLink) at: `http://metalink.oracle.com`.

Previous versions of Oracle Access Manager managed directory connection information solely through XML configuration files. Recently, Oracle Access Manager provided the ability to manage this information through the interface using the Directory Profile page in the Identity System Console and the Access System Console. However, some configuration and policy data is still managed through the XML files. Therefore, you can find yourself using combined methods for configuring load balancing.

Note: The Identity Server depends on a profile configured for the policy tree to do referential integrity for the policy directory. During Policy Manager setup, this profile is created for the policy directory for the Identity Server component. Whenever a user makes DN changes from the Identity System, the Identity Server uses this profile to update any DN references in the policy directory.

Table 4–1 Configuring Load Balancing for Directory Servers

Component	Data Store	Operation
Identity Server	Users	READ—Configured in the Directory Profile. See "Configuring Load Balancing for User Data" on page 4-6. WRITE—Not supported
	Configuration	READ—Not Supported WRITE—Not Supported
Access Server	User	READ—Configured in the Directory Profile. WRITE—Not supported Note: Write operations apply to the Access Server only if password policy is enabled.
	Configuration	READ—Configured via the ConfigureAAAServer command line tool. See "Configuring Load Balancing of Configuration & Policy Data" on page 4-7. WRITE—n.a.
	Policy	READ—Configured via the ConfigureAAAServer command line tool. WRITE—n.a. Note: If the Access Management Service for the Access Server is On, you cannot load balance requests from an Access Server to the data store containing policy information.
Policy Manager	User	READ—Configured in the Directory Profile

This section includes the following procedures:

- Configuring Load Balancing for User Data
- Configuring Load Balancing of Configuration & Policy Data
- Adjusting Connection Pooling for a Directory Server Instance

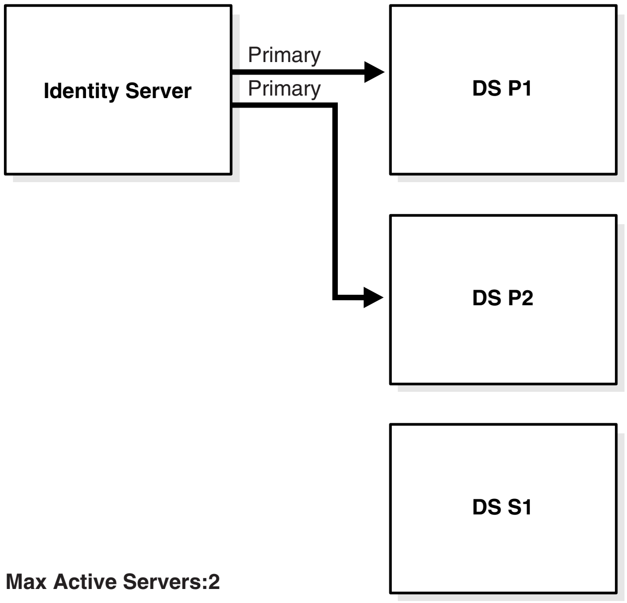
Configuring Load Balancing for User Data

By default, Oracle Access Manager creates a directory profile for each installed component. When configuring load balancing for Oracle Access Manager requests to directory servers containing user data, you use the Directory Profile page. For more information on directory profiles, see the *Oracle Access Manager Identity and Common Administration Guide*.

Maximum Active Servers: The total number of live primary directory servers you want up and running at all times. Requests are distributed evenly across these servers.

As shown in Figure 4–3 assume you have 2 primary directory servers and one secondary directory server. You want to load balance between the primary directory servers, so you should enter Max Active Servers = 2. The secondary directory server only comes into play during failover and does not impact this setting.

Figure 4–3 Load Balancing of Oracle Access Manager Requests to Directory Servers



To configure load balancing for user data

1. Access the Directory Profile page.

   For example:

   - From the Identity System Console, select System Configuration, Directory Profiles.
   - From the Access System Console, select System Configuration, View Server Settings, Directory Options.

2. Under Configure LDAP Directory Server Profiles, select the name of the profile for the component and data where you want load balancing.
3. Enter the Maximum Active Servers available for load balancing.

Configuring Load Balancing of Configuration & Policy Data

When you configure load balancing of Access Server requests to directory servers containing configuration and/or policy data, use the configureAAAserver tool. The following instructions assume that you have two or more primary directory servers set up.

Note: Load balancing for configuration data is generally not supported for the Identity Server. These instructions address the Access Server only. The ConfigureAAAServer is an older tool that does not have the most current naming conventions. Maximum Connections as shown in the tool is equivalent to the Maximum Active Servers as explained earlier in see "Configuring Load Balancing Among Oracle Access Manager and Directory Servers" on page 4-5

To configure load balancing of configuration and policy data for the Access Server

1. From the command prompt, access the configureAAAServer tool located in *AccessServer_install_dir*/access/oblix/tools/configureAAAServer
2. Run configureAAAServer utility using the reconfig *install_dir* option.

   where *install_dir* is the name of the directory where your Access Server is installed.

   For example:

   ```
   configureAAAServer reconfig "c:\Program Files\COREid1014\access"
   ```

3. Enter the number that corresponds to the Access Server security mode. These are the Access Servers that connect to the directory servers.
   - 1) Open
   - 2) Simple
   - 3) Cert

   You are then be asked if you want to specify failover information for Configuration or Policy.
4. Select Yes (Y).
5. Specify whether the data is stored in:
   - 1) Oblix tree
   - 2) Policy tree
6. Enter 1 to add a failover server at the following prompt.
   - 1) Add a failover server
   - 2) Modify a failover server
   - 3) Delete a failover server
   - 4) Modify common parameters
7. Enter the following information:
   - Directory server name
   - Directory server port

     For LDAP in an Active Directory forest environment, use port 3268 for Open mode and port 3269 for SSL mode. These two are the global catalog ports.
   - Directory server login DN
   - Directory server password
   - Directory Server security mode

  - 1) Open
  - 2) SSL

8. Enter 1 as the priority since you are configuring load balancing
   - 1) Primary
   - 2) Secondary
9. Enter 4 to modify common parameters.
10. Enter 1 to specify the Maximum Active Servers.

    Maximum Connections as shown in the tool is equivalent to the Maximum Active Servers as explained in "Configuring Load Balancing Among Oracle Access Manager and Directory Servers" on page 4-5.
11. Enter the total number of primary directory servers available for load balancing.
12. Enter 5 to Quit.
13. Enter 1 to commit changes.

Adjusting Connection Pooling for a Directory Server Instance

In addition to specifying load balancing of requests evenly across directory server instances using the Max Active Servers parameter, you can also adjust connection pooling within a specific directory server instance by entering the Initial and Maximum Connections for that specific server instance. The request is sent on the connection with the least load.

Similar to configuring load balancing, you must adjust directory pool connections from the following places:

- **Directory Profile Page**: You use this page to configure directory pool connections for user data.

  See "To adjust directory connection pooling from the directory profile" on page 4-9 for details.
- **ConfigureAAAServer Tool**: You use this tool to configure directory pool connections for policy data.

  See "To adjust directory connection pooling using the ConfigureAAAServer tool" on page 4-10 for details.

To adjust directory connection pooling from the directory profile

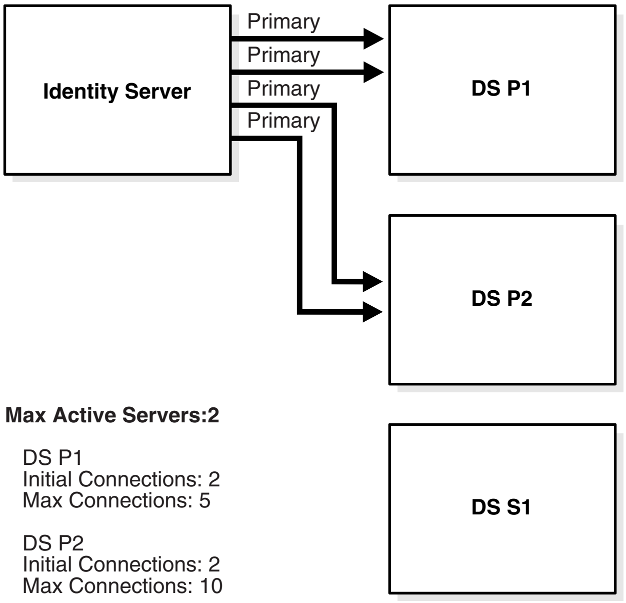
1. From the Directory Profile page, select a specific DB (directory) instance.
2. Enter the Initial Connections you want opened to this directory server.

   The default is 1.
3. Enter the Maximum Connections allowed to this directory server instance.

   If any of the connections to the directory server are lost, then Oracle Access Manager can attempt to open connections up to the maximum entered.

   When the first request is sent to a directory server, the initial number of connections is opened. When connections are lost, or when there are more service threads than connections, new connections are opened (up to the maximum). See Figure 4–4 for an example.

Figure 4–4 Adjusting Connection Pooling for Directory Server Instances



For example, assume there are 5 initial connections and 10 maximum connections to a directory server. When a service thread makes a request to the directory server, then 5 initial connections are created. Assume now, that there are 5 concurrent active service threads that require information from the directory server. They use all of the existing 5 connections.

If the concurrent service threads increase beyond 5 more connections, Oracle Access Manager can create up to 10 max connections to that directory server. When the limit of 10 connections is reached, and there are 11 or more concurrent service threads, the 11th service thread will get one of the least loaded connections from the pool of 10 connections. The connection is then shared by more than one service thread.

> **Note:** There is no general optimal value for the initial and maximum connections, given variables such as directory configuration, hardware, and so on. However, you should not set the number of connections higher than the number of threads for a given Oracle Access Manager server.

To adjust directory connection pooling using the ConfigureAAAServer tool

1. See "To configure load balancing of configuration and policy data for the Access Server" on page 4-8.
2. Specify the Modify Common Parameters option when prompted.

About Failover with Oracle Access Manager

Oracle Access Manager uses failover to provide uninterrupted service. Failover involves re-directing requests to another server when the original request destination fails.

Failover is accomplished by configuring primary and secondary servers and identifying specific parameters for the failover process. Failover can be configured between:

- Oracle Access Manager Web components to Oracle Access Manager Servers
  - WebPass requests to secondary Identity Servers
  - WebGate requests to secondary Access Servers
- Oracle Access Manager Servers to directory servers
  - Identity Server to secondary directory servers
  - Access Server to secondary directory servers
- Policy Manager to directory servers
  - Oracle Access Manager now provides the ability to perform failover from the Policy Manager to secondary directory servers.

Primary Versus Secondary Servers

Oracle Access Manager components first attempt to connect to a primary server.

If the primary server is unavailable, a connection attempt may be made to a secondary server. Oracle Access Manager continues to attempt to connect to the primary server, and when the connection is re-established, the connection to the secondary server is dropped. Any server can be configured as a primary or a secondary server. For example, you could designate less powerful or geographically distant servers as secondary.

About Failover Between Oracle Access Manager and Directory Servers

Failover for Oracle Access Manager Servers occurs when the number of live primary directory servers drops below the number in the Failover Threshold field. The instructions for configuring failover from Oracle Access Manager components to directory servers vary depending on the type of component (Identity Server, Access Server, Policy Manager), and whether you are configuring failover for user data or configuration data.

> **Note:** The Identity Server depends on a profile configured for the policy tree to do referential integrity for the policy directory. During Policy Manager setup, this profile is created for the policy directory for the Identity Server component. Whenever a user makes DN changes from the Identity System, the Identity Server uses this profile to update any DN references in the policy directory.

Table 4–2 Supported Failover Configurations for Directory Servers

Component	Data Store	Operation
Identity Server	User	READ/WRITE—Configured in the directory profile.
	Configuration	READ/WRITE—Configured in the directory profile and XML configuration files.
Access Server	User	Read/WRITE—Configured in the directory profile. Write is only applicable if a password policy is enabled.

Table 4–2 (Cont.) Supported Failover Configurations for Directory Servers

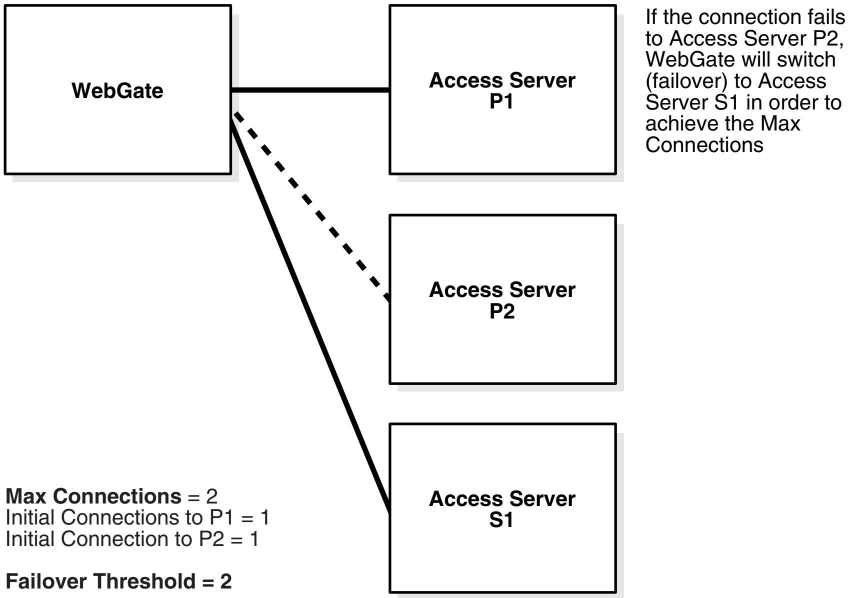
Component	Data Store	Operation
	Configuration	READ—Configured via the ConfigureAAAServer command line tool.
	Policy	READ—Configured via the ConfigureAAAServer command line tool.
Policy Manager	User	READ—Configured in the directory profile.
	Configuration	
	Policy	READ—XML configuration files.

Note: The Identity Server depends on a profile configured for the policy tree to do referential integrity for the policy directory. During Policy Manager setup, this profile is created for the policy directory for the Identity Server component. Whenever a user makes DN changes from the Identity System, the Identity Server uses this profile to update any DN references in the policy directory.

Configuring Failover of Web Components

Configuring failover enables a WebPass or WebGate to check the health of its connections, and failover to secondary Oracle Access Manager Servers in case one or more primary servers go down. You configure failover from the perspective of the Web component. See Figure 4–5 for an example.

Figure 4–5 Basic Failover Scenario between a WebGate and its Access Servers


WebGate
Access Server P1
Access Server P2
Access Server S1
If the connection fails to Access Server P2, WebGate will switch (failover) to Access Server S1 in order to achieve the Max Connections
Max Connections = 2
Initial Connections to P1 = 1
Initial Connection to P2 = 1
Failover Threshold = 2

Failover Threshold: The key to configuring failover is the Failover Threshold field in the Web component configuration. This specifies the minimum number of live primary connections required. If the number of live connections drops below the failover threshold, then the Web component attempts to establish connections to its secondary servers in the order they are listed. The default is the maximum number of connections.

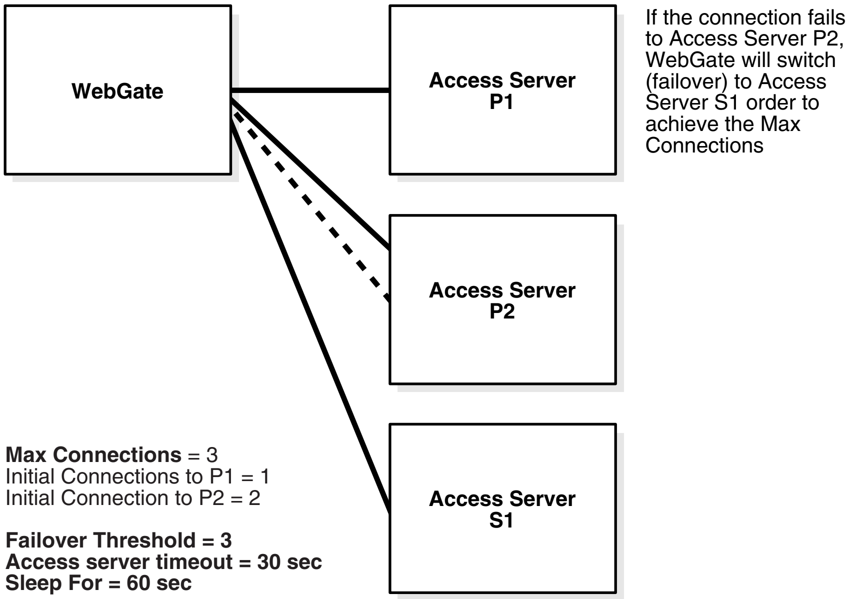
Sleep For Interval: The default interval is 60 seconds. After this interval, the WebPass or WebGate checks to see if the number of valid connections equals the maximum number of connections configured. If the number of valid connections does not equal the maximum number of connections (drops below the failover threshold), the Web component tries to establish connections to its secondary servers in the order they are configured. Then, at the interval specified, the Web component tries establishing a connection to the primary servers. When it establishes the connection, it drops the connections to the secondary servers after finishing the request it is servicing.

Timeout Threshold: Specifies how long (in seconds) the Web component waits for a non-responsive Oracle Access Manager server before it considers it unreachable and attempts to contact another. No value indicates there is no timeout, and the Web component waits endlessly for a response from the Oracle Access Manager server. Leaving no timeout can potentially result in a hung session or default to a "TCP" timeout. For example:

- For an existing WebPass, enter a value for the Identity Server Timeout Threshold.
- For an existing AccessGate, enter a value for the Access Server Timeout Threshold.

In Figure 4–6 a WebGate communicates with two primary Access Servers (Access Server P1 and Access Server P2).

Figure 4–6 Failover Scenario between a WebGate and its Associated Access Servers


WebGate
Access Server P1
Access Server P2
Access Server S1
If the connection fails to Access Server P2, WebGate will switch (failover) to Access Server S1 order to achieve the Max Connections
Max Connections = 3
Initial Connections to P1 = 1
Initial Connection to P2 = 2
Failover Threshold = 3
Access server timeout = 30 sec
Sleep For = 60 sec

- The Maximum Connections is 3. Assume that Access Server P2 can handle additional load, and therefore two initial connections are established to it.
- The Failover Threshold is 3. If the number of valid connections drops below the failover threshold, WebGate tries to establish a connection to its secondary server, Access Server S1. Assume that Access Server P2 dropped its second connection.

- The Access Server Timeout Threshold parameter is set for 30 seconds. If the WebGate does not receive a response from the Access Server P2 in 30 seconds, it considers the server unreachable and attempts to failover to Access Server S2.
- The Sleep For parameter is set for 60 seconds. Every 60 seconds this WebGate checks whether the number of valid connections to primary servers equals the specified number of maximum connections. If not, WebGate continues its attempts to re-establish connections to failed primary Access Server P2.

To configure failover for Web component requests

1. Access the WebPass or WebGate Web component where you configure failover.

   For example:

   - From the Identity System Console, select System Configuration, then click the WebPass link in the left navigation pane, then select a WebPass, then click Modify.
   - From the Access System Console, select Access System Configuration, AccessGate Configuration, All, Go, then select a WebGate.

   See the *Oracle Access Manager Identity and Common Administration Guide* and the *Oracle Access Manager Access Administration Guide* for more information.

2. In the Failover Threshold field, enter the required number of live connections from the Web component to its primary Oracle Access Manager server.
3. Enter the Sleep For interval in seconds.
4. Enter a Timeout Threshold to specify how long (in seconds) the Web component waits for a non-responsive Oracle Access Manager server before it considers it unreachable and attempts to contact another:

   - **WebPass**: Enter a value for the Identity Server Timeout Threshold.
   - **AccessGate**: Enter a value for the Access Server Timeout Threshold.

5. Save your changes.

Configuring Directory Failover for User Data

You use the Directory Profile page when configuring failover of Oracle Access Manager requests to directory servers containing user data.

Failover Threshold: The required number of live primary directory servers. If the number of primary servers drops below the failover threshold, Oracle Access Manager fails over to a secondary directory server.

When the Oracle Access Manager server sends a request on one of its directory connections, and the LDAP SDK returns a connection or server-down error, the directory server is assumed not to be available. If the number of primary directory servers drops below the failover threshold, then Oracle Access Manager attempts to establish connections to its secondary servers in the order they are listed.

If there is a primary server available when failover occurs, the Oracle Access Manager server fails over to the primary server first.

Sleep For: The number of seconds before the watcher thread wakes up and attempts to re-establish connections and create new connections if the connection was down. The watcher thread maintains a pool of good connections at all times.

By default, Oracle Access Manager creates a directory profile for each installed component. You must access this page to configure directory failover. For more information on directory profiles, see the *Oracle Access Manager Identity and Common Administration Guide.*

To configure directory failover for user data

1. Navigate to the System Console and access the Directory Profile page:

   For example:

   - From the Identity System Console, select System Configuration, Directory Profiles.
   - From the Access System Console, select System Configuration, Server Settings.

2. Select the link to the directory profile that contains connection information for the component and data where you want failover.
3. Enter the Failover Threshold.
4. In the Sleep For field, enter the number of seconds before the watcher thread wakes up and attempts to re-establish connections and create new connections if the connection was down.
5. Add the Database Instances and indicate their status as secondary servers.

Configuring Directory Failover for Configuration and Policy Data

The instructions for configuring failover from Oracle Access Manager components to directory servers vary depending on the type of component (Identity Server, Access Server, Policy Manager), and whether you are configuring failover for user data or configuration data. See Table 4–2 on page 4-11 for details about supported failover configurations for directory servers.

Task overview: Configuring directory failover for configuration and policy data

1. See "Configuring Identity Server Failover for Configuration Data" on page 4-15 for details.
2. See "Configuring Access Server Directory Failover for Configuration and Policy Data" on page 4-17 for details.

Configuring Identity Server Failover for Configuration Data

For the Identity Server, most configuration data is still managed through the XML configuration files. However, multi-language and referential-integrity data is managed through the Directory Profile page.

When the primary configuration data directory server is down, there is no way for the Identity Server to read any configuration entries. Therefore, the Identity Server reads failover.xml to obtain bootstrap secondary directory server information. See "Sample Failover.xml" on page 4-16 for an example.

Task overview: Configuring Identity Server failover for Configuration data

1. Configure failover for configuration data.

   See "To configure Identity Server directory failover for configuration data" on page 4-16 for details.

2. Create the encrypted password.

   See "To create the encrypted password for the bind DN" on page 4-16 for details.

3. Configure failover.xml.

   See "To create failover.xml" on page 4-16 for details.

To configure Identity Server directory failover for configuration data

1. From the Directory Profile page, enter failover specifications for the directory profile containing the configuration branch of the tree, as described in "Configuring Directory Failover for User Data" on page 4-14.
2. Create a file called failover.xml and add it to *IdentityServer_install_dir*/identity/oblix/config/ldap directory.

To create the encrypted password for the bind DN

1. Locate the obencrypt tool in:

   AccessServer_install_dir/access/oblix/tools/ldap_tools/

2. Run obencrypt password.
3. Copy and paste the encrypted password into your failover file, as described in "To create failover.xml" on page 4-16.

To create failover.xml

1. Copy and paste the existing sample_failover.xml template into the directory:

   IdentityServer_install_dir/identity/oblix/config/ldap

2. Open the copy with a text editor and add failover information for secondary servers using "Sample Failover.xml" on page 4-16 as a guide.
3. Copy and paste the encrypted password into your failover file.
4. Rename the copy to failover.xml.
5. Repeat as necessary for each applicable Identity Server.

Sample Failover.xml

```
?xml version="1.0" encoding="ISO-8859-1"?>
<CompoundList xmlns="http://www.oblix.com" ListName="failover.xml">
<!-- # Max number of connections allowed to all the active ldap servers -- note
this is the same as Max Active Servers>
<SimpleList>
<NameValPair ParamName="maxConnections" Value="1"></NameValPair>
</SimpleList>
<!-- # Number of seconds after which we switch to a secondary or reconnect to a
restarted primary ldap server -->
<SimpleList>
<NameValPair ParamName="sleepFor" Value="60"></NameValPair>
</SimpleList>
<!-- # Max amount of time after which a connection to the ldap server will expire
-->
<SimpleList>
<NameValPair ParamName="maxSessionTime" Value="0"></NameValPair>
</SimpleList>
<!-- # Minimun number of active primary ldap servers after which failover to a
secondary server will occur -->
<SimpleList>
```

```
NameValPair ParamName="failoverThreshold" Value="1"></NameValPair>
</SimpleList>
<!-- # Specify the list of all secondary ldap servers here -->
<ValList xmlns="http://www.oblix.com" ListName="secondary_server_list">
<ValListMember Value="sec_ldap_server"></ValListMember>
</ValList>
<!-- # Specify the details of each secondary ldap server here -->
<ValNameList xmlns="http://www.oblix.com" ListName="sec_ldap_server">
<NameValPair ParamName="ldapSecurityMode" Value="Open"></NameValPair>
<NameValPair ParamName="ldapServerName"
Value="instructor.oblix.com"></NameValPair>
<NameValPair ParamName="ldapServerPort" Value="9002"></NameValPair>
<NameValPair ParamName="ldapRootDN" Value="cn=Directory Manager"></NameValPair>
<NameValPair ParamName="ldapRootPasswd" Value="000A0259585F5C564C"></NameValPair>
<NameValPair ParamName="ldapSizeLimit" Value="0"></NameValPair>
<NameValPair ParamName="ldapTimeLimit" Value="0"></NameValPair>
</ValNameList>
</CompoundList>
```

Configuring Access Server Directory Failover for Configuration and Policy Data

The following procedures describe configuring failover from the Access Server to one or more directory servers containing configuration and/or policy data.

- To configure Access Server failover for configuration and policy data
- To add a failover directory server using the ConfigureAAAServer tool

> **See also:** The section on configuring Access Server directory failover for Oracle and policy data, and the section on configuring policy manager failover in the *Oracle Application Server Enterprise Deployment Guide*.

To configure Access Server failover for configuration and policy data

1. Using the Directory Profile page, enter the failover for the directory profile containing the `oblix` branch of the tree.

   See "Configuring Directory Failover for User Data" on page 4-14 for general instructions.

2. Repeat the above procedure for the directory server containing policy data, if applicable.

   For the Access Server, most configuration data is still managed through the configuration files. However, multi-language and referential-integrity data is managed through the Directory Profile page.

3. Add failover information using the configureAAAServer tool, as described next.

To add a failover directory server using the ConfigureAAAServer tool

1. From the command line, navigate to the folder where configureAAAserver tool is located.

   For example, the default location is:

   AccessServer_install_dir\access\oblix\tools\configureAAAServer

   where *AccessServer_install_dir* is the directory where the Access Server is located.

2. Run configureAAAServer tool using the following arguments:

   configureAAAServer reconfig *AccessServer_install_dir*

For example:

configureAAAServer reconfig "c:\Program Files\COREid1014\access"

3. Enter the number that corresponds to the Access Server security mode for Access Servers that connect to the directory servers.
   - 1) Open
   - 2) Simple
   - 3) Cert

   You are then be asked if you want to specify failover information for configuration or Policy.
4. Select Yes (Y).
5. Specify whether the data is stored in the:
   - 1) Oblix tree
   - 2) Policy tree
6. Enter 1 to add a failover server at the following prompt.
   - 1) Add a failover server
   - 2) Modify a failover server
   - 3) Delete a failover server
   - 4) Modify common parameters
   - 5) Quit
7. Enter the following information:
   - Directory server name
   - Directory server port

     For LDAP in an Active Directory forest environment, use port 3268 for Open mode and port 3269 for SSL mode. These two are the global catalog ports.
   - Directory server login DN
   - Directory server password
   - Directory Server security mode
     - 1) Open
     - 2) SSL
   - Priority Enter 2 as the priority
     - 1) Primary
     - 2) Secondary
8. Enter 5 to Quit.

   You are prompted to commit the changes.
9. Select Y to commit your changes.

ConfigureAAAServer automatically creates the following XML files in *AccessServer_install_dir*/access/oblix/config/ldap

- AppDBfailover.xml

- ConfigDBfailover.xml
- WebResrcDBfailoverxml

To configure Policy Manager failover

1. Copy the WebResrcDBfailover.xml file from the Access Server configuration directory to the Policy Manager installation directory.
2. Copy the AppDBfailover.xml file from the Access Server configuration directory to the Policy Manager install directory.
3. Copy the ConfigDBfailover.xml file from the Access Server configuration directory to the Policy Manager installation directory.

Configuring Failover Based on Directory Server Availability

A "heartbeat" mechanism polls all primary directory server connections to verify the availability of the directory service. If the directory service is not available, the heartbeat mechanism immediately initiates failover to a secondary directory server if one is configured. A failback mechanism switches from the secondary directory server back to the primary server as soon as the preferred connection is recovered.

A heartbeat_ldap_connection_timeout_in_millis parameter in globalparams.xml determines the time limit for establishing a connection with the directory server. If the time limit is reached, the Identity and Access Servers start establishing connections with another directory server. This parameter enables the Identity and Access Servers to proactively identify when a directory server is down, and it enables failover without requiring a directory service request and TCP timeout. Oracle recommends that you enable this function.

You configure the polling interval by setting the Sleep For (Seconds) parameter for each directory profile. When the host cannot be reached, further attempts to connect to that host are blocked for the specified Sleep For interval.

> **Note:** If your network is slow and the heartbeat_ldap_ connection_ timeout_in_millis is set to a low value (for example, 10 milliseconds), the heartbeat mechanism can incorrectly indicate that directory is unreachable when it is up.

To set the polling interval in the Identity System

1. From the Identity System Console, select System Configuration, Directory Profiles.

   The Configure Profiles page appears.

2. In the Configure LDAP Directory Server Profiles section of the page, click the link for the profile that you want to modify.

   The Modify Directory Server Profile page appears. This directory server profile is used by the Oracle Access Manager servers that are selected in the Used By lists on this page.

3. Enter the interval in the Sleep For (Seconds) field.

To set the polling interval in the Access System

1. From the Access System Console, select System Configuration, then click Server Settings.

2. In the Configure LDAP Directory Server Profiles section of the page, click the link for the profile that you want to modify.

   The Modify Directory Server Profile page appears. This directory server profile is used by the Oracle Access Manager servers that are selected in the Used By lists on this page.

3. Enter the interval in the Sleep For (Seconds) field.

To set the time limit for establishing a connection with the directory

1. Open the following file:

   component_install_dir/identity/apps/common/bin/globalparams.xml

   Where *component_install_dir* is the location where the Access or Identity Server was installed.

2. Edit the value for the heartbeat_ldap_connection_timeout_in_millis parameter.

   Specify the amount of time that the Identity or Access Server is to wait for a connection to be established with the directory server. The default value is 4000 (4 seconds). A value of -1 specifies that the platform's connection timeout limit should be reached before attempting to establish a connection.

To turn the heartbeat mechanism on or off

1. Open the following file *component_install_dir*/identity/apps/common/bin/globalparams.xml

   Where *component_install_dir* is the location where the Access or Identity Server was installed.

2. Edit the value for the heartbeat_enabled parameter.

   This parameter activates or deactivates the heartbeat mechanism. By default, it is set to true (on). A value of false deactivates the mechanism.

Configuring Failover Based on Directory Server Response Time

You can configure the Identity Server, Access Server, and Policy Manager to wait for a configurable amount of time (in milliseconds) for a response from a primary directory server. If no response is received within the configured time limit, the component fails over to a secondary directory server if one is configured.

The LDAPOperationTimeout setting in globalparams.xml controls the time that the Oracle Access Manager component waits for the directory server to respond.

When processing a user request, an Oracle Access Manager component can issue multiple LDAP queries. The LDAPOperationTimeout parameter applies to each query independently of other queries involved in processing the same request. For example, the LDAPOperationTimeout parameter sets a time limit for the directory server to process a single entry of a search result. If an entry in the search result set is not received within this time limit, the component fails over to a secondary server. To configure the time to wait for all search result entries, use the Oracle Access Manager administration console to configure the Time Limit parameter in the directory profile. See the *Oracle Access Manager Identity and Common Administration Guide* for details.

The default value for the LDAPOperationTimeout parameter of -1 causes the Oracle Access Manager component to wait indefinitely for the directory server to respond.

The rest of this section discusses the following topics:

- Guidelines for Configuring Failover Based on Directory Server Response Time
- Configuring the LDAPOperationTimeout and LDAPMaxNoOfRetries Parameters
- Testing the LDAPOperationTimeout Value

> **WARNING: If you keep the default value of -1 for LDAPOperationTimeout and the directory server hangs, Oracle Access Manager can hang, too. If you do not have issues with directory server hanging, you can use the default of -1.**
>
> **Also, if you set the LDAPOperationTimeout parameter to too low a value, the primary directory server can fail over even if it is operational. If the value is also too low for the secondary server, an infinite loop can occur. See the following sections for details.**

Guidelines for Configuring Failover Based on Directory Server Response Time

You determine the value for LDAPOperationTimeout based on your environment, for example, the amount of data in the directory server, network latency, the number of indexes defined for the directory server, whether the Oracle Access Manager component and the directory server communicate using SSL, and so on.

The following are guidelines for estimating the amount of time that an Oracle Access Manager component should wait before failing over to a secondary directory server:

- Conduct tests to determine the value, for example, send time-consuming requests to the Oracle Access Manager component in a comparable environment to the one you are configuring.
- Check for the message, "LDAPMaxNoOfRetries exceeded. Please verify configured value of LDAPOperationTimeout in globalparams.xml" in the following log file:

  Component_install_dir\oblix\logs\oblog.log

  This message appears in logs that are configured at the Warning level. The message indicates that a request timed out before an operational directory server could return the result. If you see this message, you should increase the value of the LDAPOperationTimeout parameter. A value that is too low can result in an infinite loop, where operational directory servers do not have adequate time to return a result.

As a safeguard for setting too low a value for the LDAPOperationTimeout parameter, The LDAPMaxNoOfRetries parameter limits the number of times that the Identity Server, Access Server, or Policy Manager can retry an LDAP operation or query in the configured directory servers. As with the parameter LDAPOperationTimeout, this parameter applies to each query independently of other queries involved in processing a request. Set the value to be greater than the number of directory servers that communicate with the Oracle Access Manager component. This ensures that at least one attempt is made to query each configured directory server.

Configuring the LDAPOperationTimeout and LDAPMaxNoOfRetries Parameters

The following procedure describes how to configure these parameters.

To configure the amount of time to wait for a response before failing over

1. Open the following file:

component_install_dir/apps/common/bin/globalparams.xml

2. Set the LDAPOperationTimeout parameter to one of the following values:

   - A positive number that indicates a time in milliseconds.
   - A value of -1 enables the directory server to determine the time to spend on the request.

3. Set the value of the LDAPMaxNoOfRetries parameter.

   The default of 0 indicates that the number of retries is equal to the number of primary and secondary directory servers that are configured to communicate with the component. A value of -1 indicates an infinite number of retries. A whole number indicates a number of permitted retries.

4. Restart the component.

Testing the LDAPOperationTimeout Value

Some Oracle Access Manager functions require multiple LDAP operations. For some of the LDAP operations involved in a request, the LDAPOperationTimeout value may be adequate, and the operation will be successful. For other operations in the request, the LDAPOperationTimeout value may be too low, and the next LDAP operation in the request may fail, assuming that the directory server is unavailable. This can result in an inconsistent state.

If you set the LDAPOperationTimeout parameter to too low a value, the primary directory server can fail over even if it is operational. If the value is also too low for the secondary server, an infinite loop can occur.

Setting the LDAPOperationTimeout parameter to a higher value does not degrade Oracle Access Manager performance. As soon as the directory server returns the results, the Oracle Access Manager continues processing.

To test for an appropriate value for the LDAPOperationTimeout parameter, you can test particularly time-consuming operations in Oracle Access Manager.

Note: In several steps of the following procedure, you investigate the processing of attributes that are configured using the DN Prefix semantic type. The DN Prefix semantic type is required for the person and group structural object classes and for all structural object classes in Organization Manager. The DN Prefix specifies the relative distinguished name (RDN) of an object. The RDN is the leftmost part of the distinguished name (DN). The DN Prefix is used when creating an object through a workflow.

To test for the optimal LDAPOperationTimeout value

1. Configure the LDAPOperationTimeout value.

   See "Configuring the LDAPOperationTimeout and LDAPMaxNoOfRetries Parameters" on page 4-21 for details.

2. Find an attribute with the DN Prefix semantic type in the User Manager, as follows.

   From the Identity System Console, click User Manager Configuration, then click Tabs, then click the link for the User Manager tab, then click Modify Attributes, and identify the attribute that is configured with the DN Prefix semantic type.

3. In the User Manager, click My Profile, then click Modify, and modify the attribute that was configured using the DN Prefix semantic type.
4. Go to the log file *component_install_dir*\oblix\logs\oblog.log, and be sure the following message does not appear in the file, "LDAPMaxNoOfRetries exceeded. Please verify configured value of LDAPOperationTimeout in globalparams.xml."

   This message appears in logs that you configure at the Warning level. If the message appears, increase the value of LDAPOperationTimeout and conduct the test again.
5. Find an attribute with the DN Prefix semantic type in the Group Manager, as follows.

   From the Identity System Console, click Group Manager Configuration, then click Tabs, then click the link for the Group Manager tab, then click Modify Attributes, and identify the attribute that was configured with the DN Prefix semantic type.
6. In the Group Manager, click My Groups, select a group, then click Modify, and modify the attribute that has the DN Prefix semantic type.
7. Look for the message described in step 4, and if it appears, increase the value of LDAPOperationTimeout and conduct the test again.
8. Find an attribute with the DN Prefix semantic type in the Org. Manager, as follows.

   From the Identity System Console, click Org. Manager Configuration, then click Tabs, then click the link for an Org Manager tab, then click Modify Attributes, and modify the attribute that was configured with the DN Prefix semantic type.
9. Look for the message described in step 4, and if it appears, increase the value of LDAPOperationTimeout and conduct the test again.
10. Determine if the LDAPOperationTimeout value is adequate for deactivating or deleting a user who is a member of a static group as follows.

    In the User Manager, view a user profile, delete the user, and check the oblog.log file as described in step 4.
11. Identify other time-consuming operations in your environment and perform them.

    After conducting the operation, check the oblog.log file, as described in step 4.

5 Cloning and Caching

A clone is a copy of a component created on a remote system using an already-installed component as a template. The in-memory storage that keeps a copy of recently used information is known as a cache. This chapter contains the following topics:

About Cloned and Synchronized Components

A clone is a copy of a component created on a remote system using an already-installed component as a template. This is accomplished by cloning the configuration of an already installed component the configuration of an already installed component instead of using the command line or the installation GUI to install a Oracle Access Manager component. Cloning creates a copy of a component on a remote system using an already-installed component as a template.

Synchronizing allows you to harmonize two installations of the same Oracle Access Manager component when one is more up-to-date than the other. Synchronization can be used to upgrade or repair installations on similar platforms.

See the *Oracle Access Manager Installation Guide* for more information on cloning and synchronizing.

About Caching Recent Information

Oracle Access Manager retrieves frequently used information from a cache rather than accessing information in the directory server. *Caching* allows faster information retrieval. To improve performance, Oracle Access Manager uses several different caches for different types of information.

The Oracle Access Manager system (OSD) caches include configuration settings and group information. The Identity and Access Systems contain different caches, as described in "About Identity System Caches and Cache Flushing" on page 5-4.

The rest of this section discusses the following topics:

- About Caching and Performance
- About Cache Timeouts

About Caching and Performance

Caching can have an impact on Oracle Access Manager performance due to a number of factors including the number of elements in the cache and the cache timeout.

The optimum number of elements in a cache is a balance between:

- The number of elements required to be in the cache. This depends on the information being cached and system usage.
- The RAM available for caching.

The higher the number of elements in a cache, the greater the probability of finding the requested information and improving performance.

In a worst case scenario, if the cache uses a lot of memory and the computer on which the component runs does not have enough RAM, the operating system could spend too much time swapping pages in and out of memory. This would decrease performance.

The minimum cache size for optimal performance would be:

N = XR

where:

N = the number of entries in the cache. X = the number of new sessions per second. R = the average length of a session in seconds. For example, if:

X = 100 sessions per second
R = 600 seconds per session
then N = 60,000 number of entries in the cache

The default cache size for Oracle Access Manager components is sufficient for most installations.

The cache timeout also plays a role in performance during cache flush operations.

About Cache Timeouts

The cache timeout specifies how long an element is held in the cache. When the time expires, the element is removed from the cache and must be retrieved from the directory.

If the information changes, but the cache does not, the Oracle Access Manager component uses outdated information until the information expires from the cache. Updating caches when you change the information is one way to avoid this problem. If updating caches is not possible, such as when a user's information is changed through other software, then configuring a reasonable timeout for the cache is another solution.

A shorter timeout means information about the user is more recent, but the Identity Server and the Access Server must fetch data more often from the directory. This may reduce performance. Setting a long timeout means out-of-date information could remain in the cache for a longer time period.

> **Note:** In general, setting a lower cache timeout value means the data is more recent. However, a cache timeout value of 0 means the cache never expires.

Table 5–1 identifies the cache timeouts that can be specified for the Identity System.

Table 5–1 Timeout Parameters for Caches for the Identity System

Identity System Timeout Parameter	Description
GroupCacheTimeout	Set the period for which the group object is valid, as described in "Managing the Group Objects Cache" on page 5-6.
oisClientTimeoutThreshold	Set the time period for Identity Server cache flush, as described in "Configuring Cache Flush for Identity Servers" on page 5-8.

Table 5–2 identifies the cache timeouts that can be specified for the Access System.

Table 5–2 Timeout Parameters for Caches for the Access System

Access System Timeout Parameter	Description
Time To Live	Use this to set the timeout of the cached password on a scheme-by-scheme basis, as described in "Configuring Password Validation by the Access Server" on page 3-33.
Group Query Cache timeout	Previously, this timeout was not configurable, and you could not flush this cache. The cache was cleared only when the cache timeout limit is reached or when you restarted the Access Servers. Today, however, this can be configured as described in "Configuring the Access Server Group Cache Timeout and Maximum Elements" on page 3-43.
Policy Cache Timeout	The Policy Cache Timeout value can be: ■ Estimated, as described in "Calculating Policy Cache Timeout" on page 3-46 ■ Set on the Access Server Configuration page, as described in "Adding an Access Server Instance", in the *Oracle Access Manager Access Administration Guide*.
User Cache Timeout	The User Cache Timeout value can be: ■ Estimate the value, as described in "Calculating the User Cache Timeout" on page 3-46 ■ Set on the Access Server Configuration page, as described in "Adding an Access Server Instance", in the *Oracle Access Manager Access Administration Guide*
WebGate Cache Timeout	The WebGate Cache Timeout value can be: ■ Estimate the value, as described in "WebGate Cache Tuning" on page 3-47 ■ Set on the AccessGate Configuration page, as described in "Adding an AccessGate", in the *Oracle Access Manager Access Administration Guide*

Table 5–2 (Cont.) Timeout Parameters for Caches for the Access System

Access System Timeout Parameter	Description
Cache timeout for WebGate and Access client configurations	There are two ways to reduce off-time network traffic between both the WebGate and Access Server, and the Access Server and the LDAP directory server: ■ Changing WebGate polling frequency for configuration information, as described in "Changing the WebGate Polling Frequency" in the *Oracle Access Manager Access Administration Guide* ■ Changing the default configuration cache timeout (`clientConfigCacheTimeout`) for WebGate and Access client configurations that are cached in the Access Server, as described in "Changing Default Configuration Cache Timeout", in the *Oracle Access Manager Access Administration Guide*

About Identity System Caches and Cache Flushing

Identity Servers can communicate with one another and do so primarily for cache flush requests. When a cache is updated on one server, that server tells the other servers to update their caches.

The Identity System has several caches. Each can receive a cache flush request if data is modified, as described in Table 5–3. The Identity System and the Access System use different user and group caches. See the Access System caches in Table 5–6 on page 5-10.

Table 5–3 Identity System Caches and Cache Flush Operations

Identity System Cache	Description	Cache Flush
Configuration Data Also known as system data or Oracle Access Manager-specific data (OSD)	Includes object classes, attributes, tabs, and panels.	Automatic when configuration data is modified Alternatively, you can clear the cache as described in "Managing Identity System Caches" on page 5-5.
Basic Configuration Data Entered in Setup	Contains the configuration base (configuration DN) and searchbase.	Automatic when basic configuration data is modified
Group Objects	Contains group details specified in the Group Manager.	Automatic when a group is modified Alternatively, you can explicitly request a cache flush as described in "Managing the Group Objects Cache" on page 5-6.
Name Cache also known as UidInfoCache	Contains the DNs of user and group objects. Is flushed whenever DNs itself or attribute like name, flags get modified objects	Automatic when the DNs or attribute flags (name, for example) are modified
Access control cache	Contains resource operations, searchbase, and the like	Automatic when ACL or searchbase is modified

Table 5–3 (Cont.) Identity System Caches and Cache Flush Operations

Identity System Cache	Description	Cache Flush
Auditing Cache	Contains information about master audit policies, message format, and so on	Automatic when a change occurs in master audit policies, message format, and so on
Workflow Cache	Contains workflow definitions and participants	Automatic when a workflow is modified
User-Defined Portal Inserts	Contains portal insert backurls, images used for buttons, and mouseover text	Invoked explicitly through a URL, as described in the *Oracle Access Manager Customization Guide.*

For details about managing Identity System caches, see "Managing Identity System Caches" on page 5-5.

The Access System caches information on password policies, policy domains, user credentials, and authentication schemes. Some operations performed in Oracle Access Manager impact the evaluation of access policies in the Access System. For example, if you deactivate a user in the Identity System, that change must be reflected in the Access System so the user does not have access to the resources that the Access System protects. For more information, see "About Access System Caches" on page 5-8.

Managing Identity System Caches

Oracle Access Manager caches configuration information for specific data such as object classes, attributes, panels, and tabs used by Oracle Access Manager applications. This information is automatically cached during startup.

The following topics provide details:

- Managing the OSD Cache
- Managing the Group Objects Cache
- Configuring Cache Flush for Identity Servers

Managing the OSD Cache

You must be a Master Administrator to view, reload, or clear the OSD cache. The following topics provide more details on managing the system cache:

- Viewing OSD Cache Content
- Clearing the OSD Cache
- Loading the OSD Cache

Viewing OSD Cache Content

You must be a Master Administrator to view the OSD cache.

To view the OSD cache contents

1. Launch the Identity System Console.
2. Click the System Configuration tab and select View Server Settings.
3. In the View Server Settings page, click Cache.
4. In the Cache page, click View Cache Contents.

The cached OSD information is displayed.

Clearing the OSD Cache

This procedure describes how to clear the OSD cache. You must clear the OSD cache before you reload it to update existing information. You must be a Master Administrator to clear the OSD cache.

To clear the OSD cache

1. Launch the Identity System Console.
2. Click the System Configuration tab and select View Server Settings.
3. In the View Server Settings page, click Cache.
4. In the Cache page, click Clear memory cache.

   The cache is cleared.

Loading the OSD Cache

The following procedure describes how to load the OSD cache. You must be a Master Administrator to reload the OSD cache.

Note: You must first clear the cache and then reload the information.

To load the OSD cache

1. Launch the Identity System Console.
2. Click the System Configuration tab and select View Server Settings.
3. In the View Server Settings page, click Cache.
4. In the Cache page, click Load memory cache.

   The latest information is loaded into the cache.

Managing the Group Objects Cache

This section includes the following topics:

- Configuring Group Cache Parameters
- Clearing the Group Cache

Configuring Group Cache Parameters

The Identity System provides a cache for group objects to boost performance for all group functions, especially those involving computation of parent (isMemberOf) groups, and children or nested groups.

A group is cached only when Oracle Access Manager makes its first request for that group. Subsequent requests for the group are directed to the cache. When you modify a group, it is removed (flushed) from the cache along with any parent, child, or nested groups.

In a multi-server configuration, when a group is modified on one Identity Server, all other Identity Servers are notified so that they remove the group and all its dependent groups from their caches. When Oracle Access Manager receives another request for the modified group, it re-stores the group in the cache. This ensures that the cache has the most recent information for the group.

When you search for a group, Oracle Access Manager searches the directory, not the cache.

You manage the group cache using the groupdbparams.xml configuration file.

To configure group cache parameters

1. Go to the file groupdbparams.xml located in:

   IdentityServer_install_dir \identity\oblix\data\common

   where *IdentityServer_install_dir* is the directory where the Identity Server is installed.

2. Configure values for the parameters in the groupdbparams.xml file that control the cache-related functions described in Table 5–4.

Table 5–4 Parameters in groupdbparams.xml

Parameter	Description
GroupCacheTimeout	The time period (in seconds) for which the object is valid. By default, the Group Cache never times out. Default = 0.
GroupCacheMax NumElements	The maximum number of group objects that can be stored in the cache. Default = 10000. If the cache is at its maximum size, objects that are not accessed often are replaced by those that are more frequently accessed. Consider the memory limitations of your system before setting the value for this parameter.
GroupCacheDisabled	Indicates whether the cache can be used or not. Default = false. A value of false indicates that the cache can be used. A value of true indicates that the cache cannot be used. If a cache is disabled, all reads of group objects go to the directory.
GroupCacheRead FromMaster	Forces reads of groups so that the cache can do reads from the master replica in a replicated environment. This ensures that the cache does not contain old information in case a read from a consumer replica occurs before the consumer replica has received the updated information from the master replica. Default = false If value = true, it indicates that the cache should read from the master replica.

Clearing the Group Cache

On occasion, it may be necessary to remove a group from the cache to maintain cache integrity. For example, you may have to remove a group that has been modified using an application other than Oracle Access Manager to ensure that the updated group is read when the cache receives a read request for the group.

A Master Identity Administrator can clear the entire group cache through the Identity System Console, as described in the following procedure. Alternatively, a Master Identity Administrator can clear the group cache with an Identity XML function named *flushGroupCache*. For details about this method, see the IdentityXML chapter in the *Oracle Access Manager Developer Guide*.

To clear group caches from the Identity System Console

1. From the Identity System Console, click the Group Manager Configuration tab.
2. On the Group Manager Configuration page, click Group Cache.
3. On the Group Cache page, click the Clear group cache link.

   The group cache is cleared and a message appears confirming whether or not the operation was successful.

Configuring Cache Flush for Identity Servers

Identity Server cache flush is accomplished with the `oisClientTimeoutThreshold` parameter in the globalparams.xml file. By default, cache flush for Identity Servers is synchronous and occurs every 60 seconds. However, you can set the parameter to flush the Identity Server cache asynchronously, as shown in Table 5–5.

Table 5–5 `oisClientTimeoutThreshold` ***Values in globalparams.xml***

Value	Description
60	By default, cache flush for Identity Servers is synchronous and occurs every 60 seconds
-1	Triggers asynchronous Identity Server cache flushing
No value	Triggers asynchronous Identity Server cache flushing
No parameter at all	Triggers asynchronous Identity Server cache flushing

In synchronous mode, the Identity Server sends a cache flush request and waits for a response. In asynchronous mode, the Identity Server does not wait for a response.

To configure cache flush for Identity Servers

1. Locate the following file:

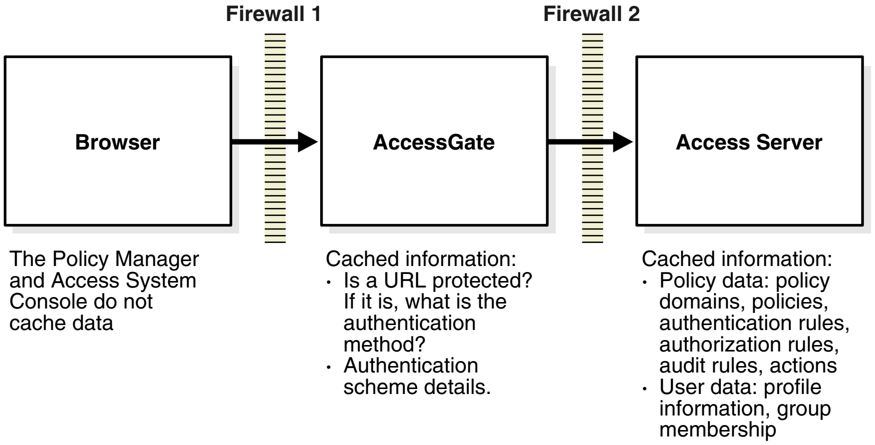
   IdentityServer_install_dir/identity/oblix/apps/common/bin/globalparams.xml
2. Set the `oisClientTimeoutThreshold` parameter using the guidelines in Table 5–5.
3. Save the file.
4. Repeat these steps for each Identity Server.

About Access System Caches

If your deployment does not include the Access System, you can skip this section.

The Access System caches information on password policies, policy domains, user credentials, and authentication schemes. Figure 5–1 illustrates how caching works in the Access System. It is not necessarily a deployment recommendation.

Figure 5–1 Caching in the Access System


Firewall 1
Firewall 2
Browser
AccessGate
Access Server
The Policy Manager and Access System Console do not cache data
Cached information:
• Is a URL protected? If it is, what is the authentication method?
• Authentication scheme details.
Cached information:
• Policy data: policy domains, policies, authentication rules, authorization rules, audit rules, actions
• User data: profile information, group membership

The following topics provide more information about Access System caches:

- Elements, the Cache Timeout, and Off-Time Network Traffic
- Access Server Cache Configuration
- Cache Configuration Using Replicated Directories
- AccessGate Cache Configuration
- Performance Improvements Using Asynchronous vs. Synchronous Cache Flush Mode
- Performance Improvements Using Mixed-Mode Communication for Cache Flush Operations

Elements, the Cache Timeout, and Off-Time Network Traffic

You can specify the maximum number of elements in a cache. The Access System ensures that the total number of elements in a cache never exceeds the maximum specified for that cache.

For example, suppose you have set the Access Server's user cache to contain a maximum of 100,000 elements. If a user is added to the system when the cache is full, the Access System removes the user element that has not been used in the longest time and adds information about the new user to the cache.

> **Note:** WebGate and Access client configurations are cached in the Access Server. To reduce off-time network traffic between WebGate and Access Server and between Access Server and LDAP directory server, you can change the default configuration cache timeout. For details about reducing network traffic between components, see the *Oracle Access Manager Access Administration Guide*.

Access Server Cache Configuration

As seen in Table 5–6, the Access System caches data on policy domains, policies, users, host identifiers, and password policies. When information is updated for any of these, and the Update Cache box is checked, the Access Server caches are updated immediately.

Table 5–6 Access System Caches and Cache Flush Operations

Access System Caches	Description	Cache Flush
Access Server Cache		
Policy cache	Contains information for policy domains, policies, authentication rules, authorization rules, audit rules, actions associated with rules, policy conditions for rules, and authentication schemes. For more information, see "Tuning the Policy Cache" on page 3-45.	Automatic when any policy configuration changes and Update Cache is selected
Access Server user data cache	Contains information from the user profile that is required in both actions and rule-based access evaluation. Also included in this cache is the user's group-membership status (whether a user is member of a group or not, for example). For more information, see "User Cache Tuning" on page 3-46	When OIS notifies AAA
URL Prefix Cache	If a URL prefix is added to a policy domain or policy, it is updated if Update Cache is selected. For more information, see "Tuning the URL Prefix Cache" on page 3-47	Automatic if Update Cache is selected.
Host identifier cache	Contains host identifiers	Automatic if Update Cache is selected.
Password policies cache	Contains password policies	Automatic if Update Cache is selected.
WebGate Cache	Contains authentication scheme details, such as how to challenge a user for credentials and the Level of the scheme. Redirection URL, if any Protection information about every resource--including the authentication scheme, URL, and operation--that is passed to the AccessGate, such as: ■ The authentication scheme key ■ The protected resource's URL Details about an authorization scheme, such as: 1. The LDAP user attribute 2. The parameters specified in the authorization rules of the policy domain	Cache flush depends upon the value set for the WebGate cache timeout. For more information, see "AccessGate Cache Configuration" on page 5-13

The following topics provide additional information:

- The Policy Cache, Cache Timeout, and Elements
- Access Server User Cache and Cache Timeout
- Timeouts That Ensure Correct Behavior in Replicated Environments

The Policy Cache, Cache Timeout, and Elements

An Access Server's policy cache contains information about policy domains, policies, authentication, authorization, audit rules, and actions.

An Access Server caches policy information for efficient runtime lookup. If policy information is not in the cache, the Access Server must obtain it from the directory. Obtaining data from the directory is slow when compared to the AccessGate obtaining information from the Access Server, so preferably all policy information should be cached.

Cache Timeout: When policy information is changed through Policy Manager or the Access System Console, Access Server caches can immediately be updated by selecting Update Cache. Policy cache timeout becomes important when the update cache feature is not used. Cache timeout should be set to how quickly the policy information needs to be updated when Update Cache is not or cannot be used. The default policy cache timeout is two hours. This is an important parameter because it ensures that after some time span the changes are saved in case any unwanted internal or external event occurs.

Maximum Elements: If the Access Server host has sufficient memory to hold all the configured policy information, the maximum elements should be set according to the maximum of the following:

- Total number of policy domains
- Total number of authentication rules, default or otherwise
- Total number of authorization rules, default or otherwise
- Total number of audit rules, default or otherwise

This configuration works for a typical deployment.

Access Server User Cache and Cache Timeout

An Access Server's user cache includes information about the:

- User Profile, which is required both in actions and rule-based access evaluation
- User's group-membership status, such as whether a user is member of a group or not

The Access Server caches a user's profile information and group membership information. The profile information includes both the information required in actions and the information required for authorization. For example, if the authorization rule for a policy allows access to anyone whose `userLevel` attribute is set to greater than 3, Access Server caches include the userLevel attribute.

Similarly, an Access Server caches information about whether or not a user is a member of a group.

User and group information can be updated through the Identity Server or other applications. As user and group information changes, the Oracle Access Manager caches become out of date. For this reason, user cache timeout is crucial.

The timeout of the user cache should be proportional to the average time interval in which part of user profile--those relevant to Access System--changes. Parts of a user profile that are relevant to the Access System include:

- Attributes returned to AccessGate in actions
- Attributes used to control access
- Attributes used in dynamic group-membership definitions that may in turn be used to control access

The maximum elements should be equal to the number of different users that may access the system in the *Access Server user cache timeout* time period.

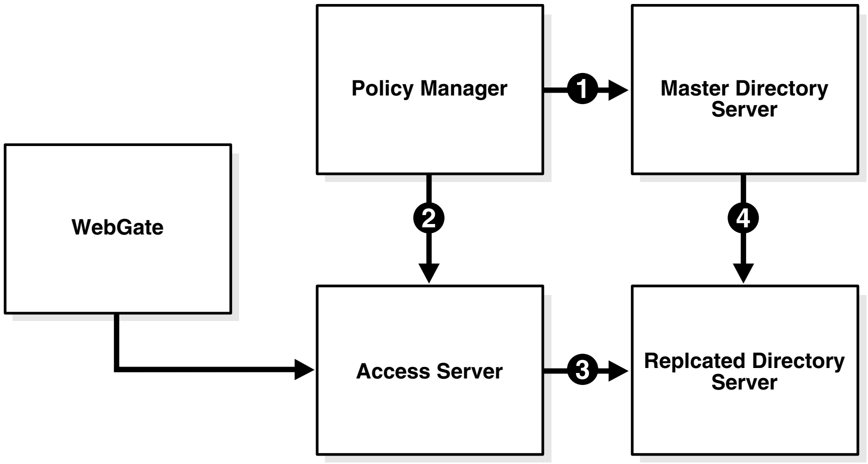
You can configure the Identity Server to notify the Access Server automatically when user or group information changes. The Access Server's caches are then automatically flushed and updated with new information. To configure the Identity Server to notify the Access Server when user information changes, you can set the `doAccessServerFlush` parameter in the Identity Server’s basedbparams.xml file: *IdentityServer_install_dir*/identity/oblix/data/common. This is currently supported only for user information, not for groups. For more information, see the parameters chapter of the *Oracle Access Manager Customization Guide*.

Cache Configuration Using Replicated Directories

When you change data using the Policy Manager, it modifies data in the directory and notifies the Access Server to flush its cache. The Access Server flushes its cache and reads the updated information from the directory server.

In a replicated directory environment, modified data is passed from the master directory server to replicated directory servers. There is a time lag before the modified data is reflected in the replicated directory servers. If the Policy Manager communicates with the master directory server and the Access Server communicates with a replicated directory server, the Access Server may flush its cache and read data before the modified data is reflected in the replicated server. As a result, the Access Server may cache old data. Thus, in a replicated environment, policy evaluation by the Access Server can be incorrect if it is based on old data. Figure 5–2 illustrates a replicated environment.

Figure 5–2 A Replicated Environment



Process overview: Replication

1. An Access Administrator uses the Policy Manager to modify data in the master directory server.
2. The Policy Manager sends a signal to the Access Server to flush its cache.
3. The Access Server flushes its cache and reads data from the replicated directory server.
4. Modified data is replicated from the master directory server to the replicated directory server.

Timeouts That Ensure Correct Behavior in Replicated Environments

To ensure that policy evaluation is accurate and that the Access Server reads the latest data, add the `splTimeout` parameter. This parameter enables you to specify a timeout (in seconds) on the element being modified. It allows for the time lag between replication from the master directory server to the replicated directory server. The `splTimeout` parameter takes precedence over the cache timeouts specified in the Access Server Configuration page of the Access System Console.

If you do not specify a value for the `splTimeout` parameter, the Access Server flushes its cache and reads the data as soon as it receives a flush signal from the Policy Manager.

The `splTimeout` parameter is located in the globalparams.xml file in:

AccessServer_install_dir/access/oblix/apps/common/bin/globalparams.xml

where *AccessServer_install_dir* is the directory where the Access Server is installed.

To ensure policy evaluation is accurate in a replicated environment

1. On the computer hosting the primary Access Server, locate the globalparams.xml file. For example:

   AccessServer_install_dir/access/oblix/apps/common/bin/globalparams.xml

2. Open the file and add the `splTimeout` parameter. For example:

```
<SimpleList>
    <NameValPair
        ParamName="splTimeout"
        Value="400"></NameValPair>
</SimpleList>
```

3. Save the file.
4. Restart the Access Server.
5. Repeat this procedure for each Access Server in your deployment.

AccessGate Cache Configuration

When you configure an AccessGate, you can specify the maximum number of elements in the cache as well as the cache timeout.

Each AccessGate caches the following information:

- Details about an authentication scheme, such as:
  - How to challenge a user for credentials
  - Level of the scheme.

For information about creating authentication schemes and their levels, see the *Oracle Access Manager Access Administration Guide*.

  - Redirection URL, if any

- Protection information about every resource--including the authentication scheme, URL, and operation--that is passed to the AccessGate, such as:
  - The authentication scheme key
  - The protected resource's URL
  - Whether the resource is protected in the system or not
  - If it is protected, the authentication method required for the resource.
- Details about an authorization scheme, such as:
  - The LDAP user attribute
  - The parameters specified in the authorization rules of the policy domain

When you configure or modify an authentication or authorization scheme, select Update Cache to update the AccessGate cache immediately with the updated scheme.

The number of URLs for which information is cached can be configured for each AccessGate. With out-of-the-box configuration, URL elements timeout every 30 minutes. When you make any policy changes that may change a URL's protection status or authentication method, select Update Cache to update the AccessGate cache immediately. This means the AccessGate's cache timeout period need not be short.

Assuming the system has sufficient memory, the maximum number of URLs cached should be at least equal to the number of distinct URLs processed by AccessGate in the amount of time specified in the Cache Timeout field.

For example http://www.myserver.com and http://myserver are treated as distinct URLs. If information about a URL is not in the cache, AccessGate makes a request to the Access Server. Fetching information from the Access Server takes longer than fetching information from a local cache, but not significantly longer.

Performance Improvements Using Asynchronous vs. Synchronous Cache Flush Mode

Access Server and system performance are enhanced by caching information, which eliminates the need to read information in the directory server for each request. When relevant information is updated and "Update Cache" is enabled, the Identity Server sends a cache flush request to the primary Access Server through the Access Manager SDK. The primary Access Server in turn sends the same cache flush request to all Access Servers directing them to flush the corresponding entry from their cache.

- Flush user cache
- Flush password policy
- Flush password redirect URLs
- Flush lost password management policy

Previous releases of Oracle Access Manager used a synchronous mode for cache flush requests from Identity Servers to Access Servers. With synchronous mode the Identity Server sends a cache flush request to the primary Access Server and the Identity Server does not proceed until it receives a response.

Note: A similar situation occurs when the Policy Manager sends a cache flush request to the Access Server. For example, if a policy is enabled or disabled within the Policy Manager, this generates a cache flush request to the Access Server.

However, any delay in the system causes a delay for the user. Further, if the response is not received within the set `OISTimeoutThreshold`, WebPass resends the request to the Identity Server which starts the cache flush cycle anew. If WebPass keeps resending the same requests to the Identity Server this could lead to a no-response situation for the end user.

Oracle Access Manager 10*g* (10.1.4.3) provides an asynchronous cache flush option to help streamline performance and avoid delays associated with synchronous cache flush operations on the Access System. The flow of information is the same whether you use the synchronous or asynchronous method. With the asynchronous method, however, the thread does not wait for a response from the Access Server before notifying the Identity Server. Instead, the request arrives at the Access Server and a response is sent immediately to the Identity Server.

You can use either the synchronous or asynchronous cache flush method by setting the value of `syncOperationMode` in the AccessGate configuration in the Access System Console. For more information about using the asynchronous cache flush method, see "Configuring Asynchronous Access System Cache Flush" on page 5-31.

You can further enhance performance during Access System cache flush operations using a mixed-security mode for communication between Oracle Access Manager components, as described next.

Performance Improvements Using Mixed-Mode Communication for Cache Flush Operations

This section provides the following topics:

- About Mixed Security Modes with Oracle Access Manager
- About Oracle Access Manager Caching and Performance

About Mixed Security Modes with Oracle Access Manager

When installing and configuring Oracle Access Manager, specific transport security guidelines must be observed. The transport security guidelines within the chapter on preparing for installation in the *Oracle Access Manager Installation Guide* correctly state that:

- Transport security between all Identity System components (Identity Servers and WebPass instances) must match: either all open, all Simple mode, or all Cert mode.
- Transport security between all Access System components (Policy Managers, Access Servers, and associated WebGates) must match: either all open, all Simple mode, or all Cert mode.
- When cache flushing is enabled on the Identity Server, the Identity Server communicates with the Access Server. In this case, the transport security mode between all Oracle Access Manager components must be the same.

Oracle Access Manager 10*g* (10.1.4.2.0) provided a method that enabled you to use Open mode communication for cache flush requests between the Identity and Access Server while retaining Simple or Cert mode for all other requests. This type of configuration is known as mixed security mode (or mixed transport security mode) communication.

Oracle Access Manager 10*g* (10.1.4.3) provides a streamlined method to implement mixed-mode communication for cache flush requests.

> **Note:** Cache flush requests do not contain sensitive data. During cache flush operations, only the LDAP configuration is read. As a result, Open mode communication is appropriate for cache flush requests. With mixed security mode communication, all except cache flush requests use Simple or Cert mode for secure communications.

With user or group modifications, the user or group DN is sent through the cache flush request. For policy modifications, the cache flush request contains the policy identifier.

For details about configuring mixed-mode communications, see "Enhancing Performance by Configuring Mixed-Mode Communication for Access Server Cache Flush Operations" on page 5-25.

About Oracle Access Manager Caching and Performance

When you install and configure an Identity Server, the Access Manager SDK is also deployed automatically. The Access Manager SDK is responsible for sending cache flush requests to the Access Server after user and group profiles are modified through Identity XML. In this case, an XML request is sent to the Identity Server through WebPass. If Access Server cache flush is enabled when the Identity Server processes the request, the Identity Server sends a cache flush request to the Access Server through the Access Manager SDK and the cache is flushed.

To enable Access Server cache flush, you must set the `doAccessServerFlush` parameter to `true` in the *IdentityServer_install_dir*/identity/oblix/data/common/basedbparams.xml file. This enables the Access Manager SDK to send requests to flush the Access Server cache. When you do this, Access Server caches are automatically flushed and replaced with the latest information. This is a best practice to ensure that all components have up-to-date information. For details, see "Automatically Flushing Access Server Caches" on page 5-17.

However, even though the automatic cache flush is a best practice, it can cause performance issues if you have multiple Access Servers that use a secure communication mode. Performance issues can occur when:

- There are frequent cache flush requests when the Identity System performs IdentityXML operations to modify a user or group profile.
- There is an SSL handshake for each request to each Access Server that is configured in Simple or Cert transport security mode.

  The SSL handshakes that are required in a secure multi-server environment can impede performance.

Using the latest Oracle Access Manager procedures, you can prevent bottlenecks and preserve system performance while enabling automatic cache flush and secure communications. For more information, see "Configuring Asynchronous Access System Cache Flush" on page 5-31.

Asynchronous cache flush operations enable the Identity System to contact the Access Server in asynchronous mode through the Access Manager SDK when there are changes to user and group information. For more information, see "Configuring Asynchronous Access System Cache Flush" on page 5-31.

Managing Access System Caches

If your deployment does not include the Access System, you can skip this section.

This section provides the following procedures and information:

- Turning Off the Access Server User Cache
- Automatically Flushing Access Server Caches
- Manually Flushing Access Server Caches
- Managing the Credential Mapping Cache

> **See Also:** "Configuring Asynchronous Access System Cache Flush" on page 5-31

Turning Off the Access Server User Cache

The following procedure describes how to turn off the Access Server user cache. You must be a Master Access Administrator to perform this task.

Turning off this cache reduces memory consumption. However it could also result in decreased efficiency in some cases. For example, when requests to the Access Server are frequent.

To turn off the user cache

1. Launch the Access System Console.
2. Navigate to Access System Configuration, then click Access Server Configuration.

   The Access Server Configuration page appears. The name, host, and port of each configured Access Server are listed on this page.
3. Click the link of the Access Server you want to modify.
4. On the Details for Access Server page, click Modify.

   The Modify Access Server page appears.
5. Set the value for Maximum Elements in User Cache to -1.
6. Save your changes.

Automatically Flushing Access Server Caches

The Identity System and the Access System use different user and group caches. An administrator may perform any of the following operations:

- Deactivating a user
- Changing user attributes
- Deleting a group
- Changing a password policy
- Changing redirect URLs

After any of the above operations take place, the directory server is updated with the new information. However, the Access Server may be unaware of these changes. For example, if you deactivate a user in Oracle Access Manager, that change should be reflected in the Access System so the user does not have access to its protected resources.

To ensure that the Access Server is informed of changes in the Identity System, you can manually flush the Access Server's user cache. Alternatively, you can configure the Identity Server to notify the Access Server of changes to user and group information. The Access Server caches are then automatically flushed and replaced with the latest information. For automatic Access Server cache flush, you must set the `doAccessServerFlush` parameter in the Identity Server’s basedbparams.xml file as described in Table 5–7.

Table 5–7 `doAccessServerFlush` ***parameter in basedbparams.xml***

Value	Description
true	Enables automatic Access Server cache flush.
false	Disables automatic Access Server cache flush. Note: This is the default.
No value	The default value, false, is presumed. Disables automatic Access Server cache flush.
No parameter	The default value, false, is presumed. Disables automatic Access Server cache flush.

Note: The user cache is not automatically flushed when changes are made to group membership through a dynamic filter or through static membership. Also, when you deactivate a user you must also disable the credential mapping cache. For details, see "Managing the Credential Mapping Cache" on page 5-21

The Access Management Service must be On when Access Manager SDK-complied code is used to connect to the Access Server. Features like automatic cache flush use the SDK that is bundled with the Identity Server. For automatic cache flushing, ensure that the Access Management Service is On in the configuration profiles of associated AccessGates and Access Servers. The Access Management Service should be Off if only WebGates are associated with an Access Server.

Note: The `UserMgmtNodeEnabled` parameter in the Access Server globalparams.xml file controls the enabling and disabling of a feature that manages WebGate memory growth. This feature is intended for Access Systems that maintain a very high number of cache flush operations per second. For more information on this parameter, see the chapter on parameters in the *Oracle Access Manager Customization Guide*.

Synchronizing Changes in the Directory Server: Any modification in a user profile or policy information is updated in the directory server. A global sequence number is updated in the directory server to track the latest changes. Before a cache flush, the Access Server checks for the number. However, if your environment includes more than one directory server, it is possible that the number for corresponding entries could be out of sync. For more information, see information on managing sync records in the *Oracle Access Manager Access Administration Guide*.

To flush the Access Server cache automatically when relevant changes occur

1. Navigate to *IdentityServer_install_dir*/identity/oblix/data/common/basedbparams.xml file.

   where *IdentityServer_install_dir* is the directory where the Identity Server is installed.

2. In the basedbparams.xml file, locate the `doAccessServerFlush` parameter and set it to `true`.

   ```
   <NameValPair ParamName="doAccessServerFlush" Value="true"/>
   ```

3. Confirm that the Access Management Service is On for AccessGates associated with this Access Server:

   a. From the Access System Console, click Access System Configuration, AccessGate Configuration, All, and click Go.

   b. From the resulting list, click a link for the appropriate AccessGate, and then click Modify.

   c. Set the Access Management Service to On, if needed, and then click Save.

4. Confirm that the Access Management Service for associated Access Servers is On:

   a. From the Access System Console, click Access System Configuration, Access Server Configuration.

   b. From the resulting list, click a link for the appropriate Access Server, and then click Modify.

   c. Set the Access Management Service to On, if needed, and then click Save.

5. Restart the Identity Server.

6. Restart the Access Server.

7. Add a dummy AccessGate using the configureAccessGate command line tool, as follows:

   configureAccessGate -i *IdentityServer_install_dir*/identity/AccessServerSDK -t AccessGate

8. Check the ObAccessClient.xml file to confirm that the port number contains the listen port that you configured for the AccessGate during installation:

   IdentityServer_install_dir/AccessServerSDK/oblix/lib/ObAccessClient.xml

Manually Flushing Access Server Caches

If you implemented automatic cache flushing for the Access Server, you can skip this topic.

If you did not implement automatic cache flushing for the Access Server, these caches contain outdated information until the cache timeout occurs. However, you can manually flush the Access Server's user and password policy caches in the Access System Console. You can flush stored information on a specific password policy or on all password policies from the Access Server cache. You can also flush cached information on all redirect URLs.

The following topics provide details:

- Flushing the Access Server User Cache Manually

- Flushing the Password Policy Cache Manually

> **See Also:** Configuring Asynchronous Access System Cache Flush on page 5-31

Flushing the Access Server User Cache Manually

You perform this task when you have multiple Access Servers and, due to an error or a crash, these servers get out of sync. You must be a Master Access Administrator to perform this task.

To flush the Access Server's user cache manually

1. Launch the Access System Console.
2. Go to Access System Configuration, User Access Configuration, and click Flush User Cache.
3. To flush a specific user's profile and group information, click Select User.

   The Selector page appears.
4. To search for a user, enter the name and click Go.

   The search results are listed under Selector.
5. Click Add to select a user, or to select all listed users, click Add All.

   The user name is listed under Selected.
6. Click Done to leave the screen.

   The selected user's name appears on the Flush User Cache screen.
7. Click Flush Cache.

   A dialog box requesting confirmation appears.
8. Click OK to remove the user's cached information; click Cancel if you do not want to remove the user's cached information.

Flushing the Password Policy Cache Manually

You perform this task when you have multiple Access Servers and, due to an error or a crash, these servers get out of sync. You must be a Master Access Administrator to perform this task.

To flush the password policy cache manually

1. From the Access System Console, click Access System Configuration, Common Information Configuration and then click Flush Password Policy Cache.
2. If you changed any information for a password policy, select that policy from the list under Flush All Cached Information for a Specified Password Policy, then click Flush Cache.

   For information about the order of password policy evaluation, see the *Oracle Access Manager Identity and Common Administration Guide.*
3. To flush all of the password policies, or if you deleted a password policy from the Identity System, then click Flush Cache to delete cached information for all password policies.
4. If you changed any of the redirect URLs on the Password Policy Management screen, click Flush Redirect URL.

For information about configuring the password redirect URLs, see the *Oracle Access Manager Identity and Common Administration Guide*.

5. Click OK to confirm your decision.

Managing the Credential Mapping Cache

By default, the credential mapping cache is enabled. In this state, the plug-in uses the cached credentials for the user.

When a user is deactivated, the Identity Server does not automatically flush the Access Server credential mapping cache. To deny a deactivated user access to protected resources you must manually disable the credential mapping cache. When the cache is turned off, the credential mapping plug-in communicates directly with the LDAP directory whenever it must map a user to a user profile (DN).

If you deactivate a user who is logged in, the user still has access to resources based on policy information and prior authentication. However, if you disable the credential mapping cache, when the user's session token expires or she logs out, she is not allowed access to a protected resource the next time she is authenticated.

To disable use of the credential mapping cache, you set the `obEnableCredentialCache` parameter to false in the credential_mapping plug-in. Table 5–8 shows the possible values for the parameter.

Table 5–8 Parameters for obEnableCredentialCache in the credential_mapping Plug-in

Value	Meaning
no value	Credential mapping cache turned on
true	Credential mapping cache turned on
false	Credential mapping cache turned off

The following example shows the credential_mapping authentication plug-in with the credential mapping cache turned off:

```
credential_mapping obMappingBase="%domain%",obMappingFilter="
(&(&(objectclass=user)(samaccountname=%userid%))
(|(!(obuseraccountcontrol=*)) (obuseraccountcontrol=ACTIVATED)))",
obdomain="domain",obEnableCredentialCache="false"
```

To set the obEnableCredentialCache parameter

1. In the Access System Console, Access System Configuration, Authentication Management.

2. Select the authentication scheme you want to modify, then click Modify.

3. Click the Plugins tab.

4. Add the obEnableCredentialCache="false" parameter to the credential_mapping plug-in.

Configuring Synchronous Cache Flush Requests between Multiple Access Servers

Oracle Access Manager 10*g* (10.1.4.3) provides a new function that enables you to set a wait period for sockets during synchronous cache flush requests between multiple Access Servers. This section includes the following topics:

- About Message Channels, Sockets, and Wait Periods
- Configuring Synchronous Cache Flush Requests Between Multiple Access Servers with a Wait Period

About Message Channels, Sockets, and Wait Periods

If `syncOperationMode` cache flush mode is not configured, by default all cache flush requests are asynchronous. For more information about using the asynchronous cache flush method, see "Configuring Asynchronous Access System Cache Flush" on page 5-31.

> **Note:** The Access Server is used to illustrate the information in this topic. However, this topic applies equally to Policy Manager and Access Manager SDK communication and Access Server and WebGate communication.

Message channels are used to manage TCP/IP communication between a WebGate and Access Server (or Policy Manager and Access Manager SDK). The Access Server also uses message channels to communicate with other Access Servers for cache flush requests.

Message channels use sockets for TCP/IP communication. Sockets are the end points of TCP/IP communication that are encompassed in a message channel. Two types of wait periods can be specified for sockets during cache flush operations:

- Indefinite wait period

  Earlier releases of Oracle Access Manager used an indefinite wait period for cache flush requests between the WebGate and Access Server. (a socket waits an infinite amount of time for I/O completion). In this case, the primary Access Server waits indefinitely to receive a response from peer Access Servers. If an Access Server hangs, other Access Servers are not sent the cache flush request. With synchronous cache flush request processing, the Identity Server would also hang until the response was received.

- Limited wait period

  You can configure a specified time period for I/O completion. If the expected operation is not completed within the specified time, an error is reported and the request is sent to other Access Servers. With synchronous requests, WebPass and Policy Manager does not hang if one Access Server hangs.

> **Note:** If an Access Server hangs, the cache flush request is sent to other Access Servers and an error is logged.

Oracle recommends defining a specific wait period for synchronous cache flush requests based on your environment. By default, the Access Server and Policy Manager wait indefinitely. You can limit this period by setting a positive integer value for the `CacheFlushTimeOut` parameter in the globalparams.xml file of the respective component (Access Server or Policy Manager). Values are specified in seconds, and then converted into milliseconds.

> **Note:** The `CacheFlushTimeOut` parameter should be tuned based on your deployment environment.

Table 5–9 describes this parameter and possible values.

Table 5–9 `CacheFlushTimeOut` ***parameter in globalparams.xml***

Value	Description
Positive integer (10, for example)	The number of seconds to wait for a response from a peer Access Server. Note: If specified, this timeout value is multiplied by 1000 (converted to milliseconds) and assigned to the Access Server client object.
Negative integer (-10, for example)	Presume the default. Note: A value of less than zero is the same as no value or no parameter.
No value	Presume the default.
No parameter	Presume an indefinite wait period. This is the default.

With a wait period specified, the following WARNING level log message occurs in the primary Access Server log file:

```
Cache Flush failed. Failed AAA server list= (List of Access Servers that are down)
```

For more information, see:

- Configuring Synchronous Cache Flush Requests Between Multiple Access Servers with a Wait Period for steps to configure the wait period for cache flush operations
- Error Handling for Message Channel Initialization During Cache Flush for details about error handling when a closed socket is encountered

Configuring Synchronous Cache Flush Requests Between Multiple Access Servers with a Wait Period

The `CacheFlushTimeOut` parameter should be tuned based on your deployment environment. You must be a Master Access Administrator to perform this task.

- For synchronous cache flush requests originating from the Access Manager SDK, you must add the `CacheFlushTimeOut` parameter to the globalparams.xml file for each Access Server.
- For synchronous cache flush requests originating from the Policy Manager, you must add the `CacheFlushTimeOut` parameter to the Policy Manager’s globalparams.xml file.

> **See Also:** "About Message Channels, Sockets, and Wait Periods" on page 5-22

To set a time period to wait for cache flush requests

1. On the computer hosting the primary Access Server, locate the globalparams.xml file. For example:

   AccessServer_install_dir/access/oblix/apps/common/bin/globalparams.xml

2. Open the file and add the `CacheFlushTimeOut` parameter under the UserMgmtNodeEnabled section. For example:

   ```
   <SimpleList>
       <NameValPair
           ParamName="UserMgmtNodeEnabled"
           Value="False"></NameValPair>
   </SimpleList>
       <SimpleList>
           <NameValPair
               ParamName="CcheFlushTimeout"
               Value="10"></NameValPair>
       </SimpleList>
   ```

3. Save the file.

4. Restart the Access Server.

5. Repeat this procedure for each Access Server in your deployment.

6. Repeat this procedure for cache flush requests originating from the Policy Manager by editing the Policy Manager's globalparams.xml file.

Error Handling for Message Channel Initialization During Cache Flush

Message channels are used to manage TCP/IP communication between WebGate and Access Server, and between Access Servers for cache flush requests. The message channel runs as a separate thread. The message channel initializes itself through a handshake with the peer Access Server. When the handshake is completed, a message is sent to the Access Server. This applies to all requests when a message channel is used with a set time period.

If an Access Server hangs during a cache flush operation, message channel initialization fails and the socket is closed. Components cannot read over a closed socket.

In earlier releases of Oracle Access Manager, operations persisted over a closed socket even if when message channel initialization failed. This resulted in socket errors. However, Oracle Access Manager 10*g* (10.1.4.3) enhances the network layer shared by WebGate and Access Server. As a result, errors that might occur as a result of message channel initialization failure due to a closed socket are avoided. Today, the message channel stops sending and receiving messages and a WARNING level log message is recorded.:

```
Warning Level Error message: "Error performing socket operations"
```

Web Gates and Access Servers are now are resilient if an Access Server hangs during a cache flush request, even when synchronous cache flush requests with a specified time period.

Enhancing Performance by Configuring Mixed-Mode Communication for Access Server Cache Flush Operations

Two methods are provided that enable you to enhance system performance by configuring mixed-mode communication between the Identity Server and Access Server during cache flush operations. This section provides the following topics:

- Method 1: Manual Access Server Configuration for Open Mode Cache Flush Requests
- Method 2: Automatic Mixed Mode Communication
- Logging and Cache Flush Operations Using Mixed Mode Communication

Method 1: Manual Access Server Configuration for Open Mode Cache Flush Requests

This topic contains the following discussions:

- About Method 1
- Configuring Access Servers Manually for Mixed Mode Transport Security
- Modifying WebGates After Manually Enabling Mixed Security Modes With Method 1

About Method 1

One method that enables you to use Simple or Cert transport security for all requests *except* cache flush requests was made available starting with Oracle Access Manager 10*g* (10.1.4.2.0). This method was described in the *Oracle Access Manager Patchset Notes Release 10.1.4 Patchset 1 (10.1.4.2.0) For All Supported Operating Systems*.

To enhance performance and simplify the routing of requests among components, you can designate a single Access Server to handle all cache flush requests. When a particular Access Server is designated to receive a cache flush request, it sends the same cache flush request to all remaining Access Servers on receipt of the request. You can ensure that Open mode communication is used when the dedicated Access Server sends cache flush requests and you can retain Simple or Cert mode for other types of requests.

Despite changing the security mode to Open for cache flush requests, as long as you initially configure the Access Servers in Simple or Cert mode, the Access Manager SDK or WebGate still communicates with the Access Servers in Simple or Cert mode. This enables you to preserve security for most operations.

The ability to configure an Access Server to use Open mode for some operations and a secure mode for other operations is known as a *mixed security mode*.

The following task overview describes how to configure your environment to handle cache flush requests using Open mode while retaining secure communication for other types of requests.

Task overview: Manually configuring Access Servers for Open mode cache flush requests while maintain secure communication for other requests

1. Install Access Servers in Simple or Cert mode (or upgrade your deployment and configure all Access Servers to use Simple or Cert mode).

   See *Oracle Access Manager Installation Guide* or the *Oracle Access Manager Upgrade Guide* for instructions.

2. Designate one Access Server or a cluster of Access Servers to handle all cache flush requests using related items in the Access Server Configuration pages.

   You can designate one Access Server for all flush requests from the Identity side by listing only one primary server for the AccessGate serving the Access Manager SDK. From the AccessGate Configuration page, choose a name and click Modify. From the Details for AccessGate page, click List All Access Servers and confirm that there is only one for the AccessGate. For more information about Access Server and AccessGate configuration, see the *Oracle Access Manager Access Administration Guide*.

   Note: For cache flush requests from the Policy Manager, you cannot designate a particular Access Server to receive the request. Policy Manager sends cache flush requests to all available Access Servers it locates through the LDAP directory server.

3. Configure WebGates and AccessGates to communicate with their respective Access Servers using Simple or Cert mode.

4. Convert all of the Access Servers to mixed transportation security mode either method 1 or method 2.

   See Also: "Method 2: Automatic Mixed Mode Communication" on page 5-28

5. Review the following topics, and perform tasks when needed:

   - "Caveats and Conditions for Method 1" on page 5-26
   - "Modifying WebGates After Manually Enabling Mixed Security Modes With Method 1" on page 5-27

Caveats and Conditions for Method 1 The following important conditions apply when you choose method 1 and enable mixed security mode communication manually.

After you configured mixed security mode communication, you might receive the following message when you view a list of Access Servers associated with a particular WebGate:

```
"Not Responding Transport security mismatch".
```

You can safely ignore this message. However, there is a second condition.

After configuring mixed security mode manually using method 1, you must follow a specific method to modify an AccessGate or WebGate. Otherwise, WebGate could not contact the Access Server when running the `configurewebgate` or `configureaccessgate` tool. Specifically, when you attempted to modify an AccessGate or WebGate, all previous **Preferred HTTP Host** settings were removed. To avoid this issue after configuring Access Servers using method 1, perform the task in "Modifying WebGates After Manually Enabling Mixed Security Modes With Method 1" on page 5-27.

Configuring Access Servers Manually for Mixed Mode Transport Security

The following procedure describes how to configure Access Servers to use Open transport security for cache flush requests, and SSL or Cert transport security for all other requests. This is useful when you have implemented automatic cache flush in an environment that primarily uses secure communication, as described in the chapter on caching in the *Oracle Access Manager Deployment Guide*.

Note: Your deployment must be 10*g* (10.1.4.2.0) or later.

You start this procedure by ensuring that every Access Server is configured to use either Simple or Cert mode for communication and that all WebGates or AccessGates communicate with all Access Servers in Simple or Cert mode. You then configure all Access Servers to use Open transport security mode and restart them. After restarting Access Servers, they retain the certificates required for secure communications and continue to send and receive most information using Simple or Cert mode. However, these Access Servers now use Open mode for cache flush requests.

To configure Access Servers manually to use mixed security mode: method 1

1. Ensure that your deployment is configured for secure communications (either Simple or Cert mode) according to transport security guidelines in the "Preparing to Install" chapter of the *Oracle Access Manager Installation Guide*.

   Transport Security: either Simple or Cert

   For details, see the *Oracle Access Manager Access Administration Guide*.
2. Run the `configurewebgate` or `configureaccessgate` tool to configure all WebGates (and AccessGates) to communicate with all Access Servers in the same secure mode.

   For details, see the *Oracle Access Manager Access Administration Guide*.
3. From the Access System Console, modify every Access Server configuration to change the mode from Simple or Cert to Open. For example:

   Transport Security: Open
4. Restart the Access Servers to enable them to use Open mode for cache flush requests.
5. Repeat Steps 3 and 4 for all Access Servers.
6. Review "Caveats and Conditions for Method 1" on page 5-26.
7. To modify an AccessGate or WebGate after performing steps 1-5, see "Modifying WebGates After Manually Enabling Mixed Security Modes With Method 1" on page 5-27.

Modifying WebGates After Manually Enabling Mixed Security Modes With Method 1

Use the following procedure to avoid issues described in "Caveats and Conditions for Method 1".

To modify an AccessGate or WebGate after manually enabling mixed transport security mode using method 1

1. Before modifying the AccessGate or WebGate, from the Access System Console reconfigure all Access Servers to use a secure transport security mode.

   Transport Security: either Simple or Cert
2. Modify the AccessGate or WebGate as needed.
3. From the Access System Console, reconfigure all Access Servers to use Open mode for transport security.

Transport Security: Open

4. Restart the Access Servers.

Method 2: Automatic Mixed Mode Communication

While Method 1 remains valid, 10*g* (10.1.4.3) provides a new method to enable mixed mode communication. This new method streamlines your effort and avoids some of the issues associated with Method 1.

The following topics describe Method 2:

- About Method 2: Automatic Mixed Mode Communication
- Using Method 2 for Automatic Mixed Mode Security with Access System Cache Flush Requests

About Method 2: Automatic Mixed Mode Communication

Oracle Access Manager 10*g* (10.1.4.3) provides automatic mixed mode communication with the `setAccessFlushInOpenMode` parameter in the globalparams.xml of the Access Server and Policy Manager (to flush the policy cache in Open mode).

If the parameter is absent from the file, or if it is set to "`true`", a cache flush message channel is created in Open mode to improve performance for cache flush requests. Otherwise, the configured transport security of the Access Server or Policy Manager are used, as follows:

Table 5–10 `setAccessFlushInOpenMode` ***for Automatic Mixed-Mode Communication***

Status	Description
No parameter	Parameter is consider to be "`true`" and Open mode is used for all Access Server cache flush requests.
True	Open mode is used for all Access Server cache flush requests. This is the default.
False	The transport security mode specified in the Access Server Configuration page (or during Policy Manager setup) is used for cache flush and all other requests.

Task overview: Enabling automatic mixed mode communication for Access System cache flush operations

1. Confirm that your 10*g* (10.1.4.3) deployment is configured for secure communications (either Simple or Cert mode) according to transport security guidelines in the "Preparing to Install" chapter of the *Oracle Access Manager Installation Guide*.

   Transport Security: either Simple or Cert

   For details about setting transport security, see the *Oracle Access Manager Access Administration Guide*.

2. **Automatic Mixed Mode**: Leave the globalparams.xml file as is. Automatic mixed mode communication for Access Server and Policy Manager cache flush operations is enabled by default.

3. **Disable Automatic Mixed Mode Security**: In the globalparams.xml file for the component, set `setAccessFlushInOpenMode` to "`false`", as described in "Using Method 2 for Automatic Mixed Mode Security with Access System Cache Flush Requests".

Using Method 2 for Automatic Mixed Mode Security with Access System Cache Flush Requests

By default, automatic mixed mode security for Access System cache flush operations is enabled in 10*g* (10.1.4.3).

> **Note:** To disable automatic mixed mode and use the configured transport security, set `setAccessFlushInOpenMode` to "`false`".

To enable or disable automatic mixed mode security for Access System cache flush requests

1. Ensure that your deployment is configured for secure communications with Simple or Cert mode according to the transport security guidelines in the "Preparing to Install" chapter of the *Oracle Access Manager Installation Guide*.
2. Automatic Mixed Mode Security: Take no action.
3. Disable Automatic Mixed Mode Security (optional): For each Access Server and Policy Manager perform the following steps.
   a. On the computer hosting the Access Server, locate the globalparams.xml file in the following directory path:

      ```
      component_install_dir\access\oblix\apps\common\bin\globalparams.xml
      ```

   b. At the end of the file, add the `setAccessFlushInOpenMode` parameter with a value of `false`. For example:

      ```
      <SimpleList>
           <NameValPair
           ParamName="UserMgmtNodeEnabled"
           Value="False"></NameValPair>
      </SimpleList>
      <SimpleList>
           <NameValPair
           ParamName="setAccessFlushInOpenMode"
           Value="False"></NameValPair>
      </SimpleList>
      </CompoundList>
      ```

   c. Restart the Access Sever so that the modified value takes effect (parameters in this file are read only once, when the Access Server is started).
   d. Repeat for all Access Servers.
   e. Repeat for all Policy Managers.

Logging and Cache Flush Operations Using Mixed Mode Communication

This topic includes the following discussions:

- About Cache Flush Logging with Mixed Mode Communication
- Enabling Cache Flush Logging for Mixed Mode Communication

About Cache Flush Logging with Mixed Mode Communication

The Oracle Access Manager logging feature enables you to analyze system performance and health, and to troubleshoot issues. You set up logging by editing a configuration file that is stored with each instance of a component. To set up logging for Access Servers, locate and edit the following file for each Access Server instance:

AccessServer_install_dir\access\oblix\config\oblog_config.xml

> **Important:** Do not change the default path or name of the log configuration file.

The configuration file contains XML statements that you can edit using a text editor. Each default, the log configuration file contains comments and samples that you can use when creating your own configuration.

You can send log output to a system log or a file using the LOG_WRITER parameter:

- A log file resides under the root installation directory of the component.
- The system file of the host for the component.

  If more than one component resides on the same host, all components send data to the system log file on that host.

You can send logs of a particular level, or logs of different levels, to more than one type of log writer. For instance, you can send Fatal data to the system log, and send Debug data to a file. Or, you can send Fatal data to both the system log and a file.

A complete definition for your log output includes a LOG_WRITER value and a LOGLEVEL. This complete definition is known as a log-handler. An example is shown here:

```
<!--Write all Debug logs to the system logger. -->
      <ValNameList xmlns="http://www.oblix.com" ListName="LogDebug3Sys">
         <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_DEBUG3" />
         <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
         <NameValPair ParamName="LOG_STATUS" Value="On" />
      </ValNameList>
```

LOGLEVEL_DEBUG3 records a large amount of data that is useful for debugging a performance-sensitive function. Using LOGLEVEL_DEBUG3 and flushing a cache by modifying user or policy information records a message in the output file. If the cache flush request between Access Servers was sent in Open mode with other requests using the configured mode (also known as mixed mode), each Access Server log file includes the following message:

```
Cache flush request to all AAA servers will be in open mode
```

If this message is not present in the Access Server log, then mixed mode was not used for the cache flush request. Instead, the configured transport security mode for that Access Server was used.

When you save changes to the log configuration file, they are picked up every 60 seconds automatically. For more information, see the chapter on logging in the *Oracle Access Manager Identity and Common Administration Guide*.

Enabling Cache Flush Logging for Mixed Mode Communication

You use the following procedure to enable cache flush logging to capture mixed mode communication methods.

To enable logging for cache flush operations

1. Locate the Access Server log configuration file, as follows:

   AccessServer_install_dir\access\oblix\config\oblog_config.xml

2. Open the file with a text editor and set the LOGLEVEL_ to DEBUG3:

```
<!--Write all Debug logs to the system logger. -->
      <ValNameList xmlns="http://www.oblix.com" ListName="LogDebug3Sys">
         <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_DEBUG3" />
```

3. Define the LOG_WRITER output location. For example:

```
      <ValNameList xmlns="http://www.oblix.com" ListName="LogDebug2Sys">
         <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_DEBUG3" />
         <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
         <NameValPair ParamName="LOG_STATUS" Value="On" />
      </ValNameList>
```

4. Save the file.
5. Test logging by modifying user or policy information, as usual.
6. Check the log output for the message that indicates mixed mode communication.

```
     Cache flush request to all AAA servers will be in open mode
```

Configuring Asynchronous Access System Cache Flush

If your deployment does not include an Access System, you can skip this section.

You must be a Master Access Administrator to perform tasks in this section. This section includes the following topics:.

- About Asynchronous Access System Cache Flush Operations
- Configuring Asynchronous Access System Cache Flush Operations

About Asynchronous Access System Cache Flush Operations

Oracle Access Manager 10*g* (10.1.4.3) provides new parameters to enable asynchronous cache flush operations between the Identity Server and Access Server. The following Access Server caches are flushed:

- Flush user cache
- Flush password policy
- Flush password redirect URLs
- Flush lost password management policy

The `syncOperationMode`" parameter value in the AccessGate Configuration determines whether you have synchronous or asynchronous cache flush operations, as shown in Table 5–11. By default, this parameter value is `false`.

Table 5–11 `syncOperationMode` ***parameter in the AccessGate Configuration***

Value	Description
true	Enables synchronous Access Server cache flush.
false	Enables asynchronous cache flush operations; disable synchronous Access Server cache flush. This is the default.
No value	Presumes the default, false. Enables asynchronous cache flush operations; disable synchronous Access Server cache flush.
No parameter	Presumes the default, false. Enables asynchronous cache flush operations; disable synchronous Access Server cache flush.

Configuring Asynchronous Access System Cache Flush Operations

You must be a Master Access Administrator to perform this task. This task is performed for Access Server and Access Manager SDK communication

See Also: "Performance Improvements Using Asynchronous vs. Synchronous Cache Flush Mode" on page 5-14

To enable asynchronous cache flush operations between the Identity and Access Servers

1. From the Access System Console, click Access System Configuration, AccessGate Configuration
2. Click All, click Go, then click a link for the appropriate AccessGate and click Modify.
3. In the section for User-Defined Parameters, enter the following parameter and value:

   Parameters: `syncOperationMode`

   Value: `true`
4. Click Save.

6

Reconfiguring the System

You can change basic components that you specified during Oracle Access Manager installation, such as the person object class or the directory server host. This chapter describes system-level reconfiguration.

This chapter includes the following topics:

- What Can Be Reconfigured
- Performing Reconfiguration That Requires Re-Running Setup
- Updating the LDAP Bind Password

What Can Be Reconfigured

There are a number of basic system components that can be reconfigured:

- You can configure Oracle Access Manager against a different directory server (for configuration or policy data).
- You can specify a new person or group object class.
- You can change the class attribute for the person or group object class.
- You can reconfigure the following characteristics of the directory:
  - The LDAP bind password
  - The host name
  - Port number
  - Domain name
  - Root DN
  - Root password
  - Configuration DN
  - Searchbase

Note: All actions except changing the LDAP bind password require re-running setup.

Performing Reconfiguration That Requires Re-Running Setup

During installation, data that you specify is written to a number of areas, including the following:

- setup.xml
- configInfo.xml
- ois_server_config.xml
- The directory server

The following procedure describes how to reconfigure Oracle Access Manager so that it works properly after you make any of the changes described in "What Can Be Reconfigured" on page 6-1.

To update the system configuration

1. Shut down the Web server that runs the WebPass.
2. Stop the Identity Server Service.
3. Back up your directory configuration data by exporting it to an LDIF file.
4. Rename the following file to ensure that you have a backup copy:

   `IdentityServer_install_dir/identity/oblix/config/ois_server_config.xml.bak`

5. From the directory that you navigated to in the preceding step, back up and then delete the following files:

   - setup.xml
   - configInfo.xml
   - ois_server_config.xml

6. Copy the file ois_server__config.bak to ois_server__config.xml.

   This action allows you to change the configuration settings when you re-run the setup program later in this procedure. It causes the Identity Server to retrieve settings from *ois_server*__config.xml during setup instead of retrieving the settings from the directory. The information in ois_server__config.xml is migrated to the directory when the Identity Server is restarted.

7. In the branch of the directory where your policies are stored, locate the WebResrcDB container.
8. In the WebResrcDB container, delete the following entries:

   - The entry for WebPass.

     The cn for this entry is the ID that you supplied when installing WebPass. Example: wp1_50.

   - The entry for the Identity Server.

     The cn for this entry is the ID that you supplied when installing the Identity Server. Example: ois1_50.

   - The entry with a timestamp for its ID.

     Example: 20010815T16221897. This entry connects the WebPass and Identity Server components.

9. In the branch of the directory where your policies are stored, locate the DBAgents container and delete all entries under this container.
10. Restart the Identity Server Service.
11. Restart the Web server that runs the WebPass.

12. From your browser, access the Identity System Console:

    `http://`*`server`*`:`*`port`*`/identity/oblix/`

13. Rerun the setup program, as described in the following procedure for the Identity System and change any settings that you want to change.

    The setup program displays the information that was previously configured for Oracle Access Manager. You can change the configuration information as needed when you rerun setup.

    See the *Oracle Access Manager Identity and Common Administration Guide* for details on rerunning setup for the Access System.

14. Restart the Identity Server.

    The information in ois_server_config.xml (the server name, port, administrator DN, password, searchbase, and configuration base) is migrated back to the directory and the information in the config.xml file is deleted.

To rerun Identity System setup

1. Shut down all but one Identity Server if there is more than one running.
2. Go to the only remaining running Identity Server host and open the setup.xml file:

   IdentityServer_install_dir/identity/oblix/config/setup.xml

3. Remove the status parameter (or change the status parameter value from "done" to "incomplete"), as shown below:

   For example:

   ```
   <NameValPair ParamName="status" Value="incomplete"></NameValPair>
   ```

4. Save the file.
5. Restart the Identity Server.
6. From your Web browser, launch the Identity System Console.

   You see a Setup page similar to the one that appears during the initial Identity System setup.

7. Initiate setup again and specify the new information.
8. After completing the setup, restart the other Identity Servers.

   The other Identity Servers should pick up the new information.

Updating the LDAP Bind Password

You may need to periodically update the LDAP bind password for the directory servers that communicate with Oracle Access Manager components. For example, you may want to update the LDAP bind password to comply with government regulations.

When you update the LDAP bind password for the directory server, you must also update corresponding entries in the Oracle Access Manager configuration directory. The configuration directory server stores Oracle Access Manager configuration data, including the directory server profiles that you defined in the Oracle Access Manager administrative console. Each directory server profile contains a Database Instance section that includes the password for the directory server.

Oracle Access Manager stores directory server profiles for the following components:

- The Identity Server
- The Policy Manager
- The Access Server

The LDAP bind password is stored in encrypted format in the configuration files. The configuration data files for the directory servers and the failover directory servers are stored in the following location:

component install dir/config/ldap

Oracle Access Manager provides the modify LDAP bind password tool (named ModifyLDAPBindPasswd) to enable you to reset the LDAP bind password in the Oracle Access Manager configuration files. You can reset the LDAP bind password without restarting any servers or re-running setup.

The following task overview is the recommended approach for automatic periodic updates of the bind password. You can launch the ModifyLDAPBindPasswd tool interactively instead of using a script. However, if you choose to update the password interactively, you must repeat the information for each Oracle Access Manager instance in your environment.

Task overview: Updating the LDAP Bind Password

1. Create an encrypted file that contains the updated password, as described in"Generating the Encrypted Password File" on page 6-9.

   Oracle recommends you use an encrypted file to provide the updated password. However, you can supply the password interactively when running the tool.

2. Update the LDAP bind password for configuration data with the first installed Identity Server, as described in "Updating the LDAP Bind Password for Configuration Data" on page 6-10.

3. Update the LDAP bind password in the configuration files for each additional instance of the Identity Server, Policy Manager, and Access Server, using slightly different options than you used in Step 2.

   See "Updating the LDAP Bind Password" on page 6-8 for details.

4. For each directory server host name variation, re-run the tool.

   For example, if you are running a host named "computer1" that resides in domain ".company.com," you can configure the host name in Oracle Access Manager as both "computer1.company.com" and "computer1." In this case, you must run the tool once for each configured host name.

   See the information on using host name variations in the *Oracle Access Manager Identity and Common Administration Guide* for details.

5. Modify the bind password in the directory server itself.

   Both the old and new passwords are stored in the Oracle Access Manager configuration files, so that the old password continues to be valid until you have completed all updates.

Additional information is provided in the following topics:

- About the ModifyLDAPBindPasswd Tool and Logs
- Parameters for the ModifyLDAPBindPasswd Tool
- About Using a Script
- Updating the LDAP Bind Password

- Changing the LDAP Bind Password When Running in ADSI Mode

About the ModifyLDAPBindPasswd Tool and Logs

You run the ModifyLDAPBindPasswd tool for each instance of each relevant component. You can run this tool from the command line or as a script.

The ModifyLDAPBindPasswd tool is available for all platforms. It is located in the following installation path of relevant Oracle Access Manager components, as follows:

component_install_dir/oblix/tools/modbinpasswd/ModifyLDAPBindPasswd

Where *component_install_dir* is the installation directory for the component (Identity Server, Policy Manager, and Access Server). You run the ModifyLDAPBindPasswd tool from the specific *component_install_dir* for which you want to update the directory bind password.

> **Note:** For security purposes, the ModifyLDAPBindPasswd tool checks the credentials for the directory server before making the changes.

If you run the tool from the command line, the tool prompts for any needed parameters and values that you failed to provide. However, you can create a script if you want to perform periodic updates of the password.

> **Note:** There is no rollback mechanism for this tool. You re-run the tool to ensure that the configuration files and directory server have the correct values.

If any errors occur during processing, they are written to a log file:

component_install_dir/oblix/tools/modbinpasswd/ModifyLDAPBindPasswd.log

For more information about using this tool, see the next topic, "Parameters for the ModifyLDAPBindPasswd Tool".

Parameters for the ModifyLDAPBindPasswd Tool

This topic provides tables of parameters for the ModifyLDAPBindPasswd tool. Each table focuses on a specific set of parameters, including those that are required for every procedure and use case, those that are needed in addition to required parameters for generating a password file, and those that are needed in addition to required parameters for other use cases.

Table 6–1 lists the parameters that are required to run the ModifyLDAPBindPasswd tool to update the password.

Table 6–1 Required Parameters for the ModifyLDAPBindPasswd Tool

Parameter	Description
`-i` *`install_dir`*	This is the installation directory for the Oracle Access Manager component for which the tool is being run. This is the first parameter that must be specified when using this tool.

Table 6–1 (Cont.) Required Parameters for the ModifyLDAPBindPasswd Tool

Parameter	Description
`-c` *`component`*	This is the type of component for which the tool is being run. The following are possible values for this parameter: ■ `is`—Use this option when running the tool for an Identity Server instance. ■ `pm`—Use this option when running the tool for a Policy Manager instance. ■ `as`—Use this option when running the tool for an Access Server instance. **Note**: The component must be consistent with the path. A mismatch results in an error. **Correct**: `... -i` *IdentityServer_install_dir* `-c is ...` **Incorrerct**: `... -i` *IdentityServer_install_dir* `-c pm ...`
`-t` *`target`*	This is the target to be updated. The following are possible values: ■ `all`—When running the tool on the first component for the first time, specify `all`. First time use: You must run the tool for the first time on the first installed Identity Server. The `all` value updates the LDAP bind password for the directory server in all directory server profiles and configuration files for the Identity Server. ■ `file`—Updates the LDAP bind password in all relevant configuration files. Use this option for subsequent Identity Servers, the Policy Manager, and the Access Server. ■ `ds`—Use this option to update the LDAP bind password for the Identity Server if user data is stored in a separate directory server from the configuration data. Run the tool from the Identity Server installation directory if you use the `ds` option.

Table 6–2 illustrates the command options that you use with the ModifyLDAPBindPasswd tool to generate a password in an encrypted file. These parameters are specified in addition to the standard required parameters in the previous table.

Table 6–2 Parameters for Creating an Encrypted Password

Parameter	Description
`-genpasswdfile`	Indicates that the tool should generate a password file. This file can be passed when you run the ModifyLDAPBindPasswd tool to update the password.
`filename`	You can supply either the full path and name of the file that will contain the password, or just the file name. If you do not supply an .xml extension, it is supplied automatically. This file is encrypted.

Table 6–3 lists the additional parameters for the ModifyLDAPBindPasswd tool. If you omit one of these parameters from the command line, the tool prompts for the

parameter if it is needed. There are no default values for these parameters. These parameters are in addition to the standard required parameters in Table 6–1.

Table 6–3 Additional Parameters for Changing the LDAP Bind Password

Parameter	Description and Caveats
`-h` *host*	The name of the computer that stores the directory profile where you want to update the directory server LDAP bind password.
`-p` *port*	The listen port for the computer that stores the directory profile where you want to update the directory server LDAP bind password.
`-D` *bind dn*	The bind DN for the directory server that stores the configuration data.
`-w` *bind password*	The bind password for the directory server that stores the configuration data in clear text format. Do not specify this option if you are using the `-j` option to pass in an encrypted password file.
`-j` *file_for_bind_ password*	Use this option if you are running the command using a script. This file contains all of the passwords that are required to update the bind password. If you use this option, the -w, `-x`, and `-y` options cannot be specified.
`-u` *host*	The computer that contains the directory server whose bind password you are updating. Do not specify this option if you are using the `-s` option to update the password on the same directory server that contains configuration data.
`-v` *port*	The listen port for the computer that contains the directory server where you want to change the bind password. Do not specify this option if you are using the `-s` option to update the password on the same directory server that contains configuration data.
`-E` *bind dn*	The bind DN for the directory server whose bind password you are changing. Do not specify this option if you are using the `-s` option to update the password on the same directory server that contains configuration data. This applies to the Identity Server, Access Server, and Policy Manager.
`-x` *old password*	The existing bind password for the directory server whose bind password you are updating. Do not specify this option if you are using the `-j` option. Also do not specify this option if you are using the `-s` option to update the password on the same directory server that contains configuration data.
`-y` *new password*	The new bind password for the directory server whose bind password you are updating. Do not specify this option if you are using the `-j` option to pass in an encrypted password file.
`-Z`	Valid if the directory server being updated is different from the configuration directory server. If you specify this parameter, the bind occurs in SSL mode. If you omit it, open mode is used. Do not specify this option if you are using the `-s` option to update the password on the same directory server that contains configuration data.

About Using a Script

If you choose to create a script rather than use the modifyldapbindpasswd tool, be sure of the path to the script and ensure appropriate permissions for the owner and those who will use the script.

When using a script, you must generate the password file as described in "To generate the encrypted password file" on page 6-9. This file contains all of the passwords that are required to update the bind password. Also, you must pass the encrypted password file using the `-j` *`file_for_bind_password`* parameter. In this case, the -w, -x, and -y options cannot be specified. For more information, see Table 6–3.

Updating the LDAP Bind Password

The following topics describe and illustrate tasks that you perform:

- About Updating the LDAP Bind Password
- Generating the Encrypted Password File
- Updating the LDAP Bind Password for Configuration Data
- Updating the LDAP Bind Password for User Data
- Updating the LDAP Bind Password for Policy Data

About Updating the LDAP Bind Password

The procedures for updating the LDAP Bind password include individual steps to ensure that the password is updated appropriately, as follows:

First Identity Server and Directory Profiles: You must update the LDAP bind password that is stored in the Identity Server configuration files (the config.xml files) and all directory profiles for the directory server. When running the tool on the first component for the first time, you must specify the -t parameter with the *`all`* option. The *`all`* option updates the LDAP bind password for the directory server in all directory server profiles and configuration files for the Identity Server.

```
$ modifyldapbindpasswd -i /opt/oam/identity -c is -t all
Please enter host machine for Configuration DS : YourConfigDSName
```

Remaining Identity Servers, All Policy Managers and Access Servers: You must update the LDAP bind password in the configuration files for each additional instance of the Identity Server, Policy Manager, and Access Server. In this case, you use many of the same parameters as you did for the first Identity Server instance. The exception is the -t parameter which, in this case, requires the *`file`* option.

```
$ modifyldapbindpasswd -i /opt/oam/access -c pm -t file
Please enter host machine for Configuration DS : YourConfigDSName
```

Different Host Names: You must repeat the procedure for each directory server host name variation. For example, if you are running a host named *computer1* that resides in domain *company.com*, you can configure the host name in Oracle Access Manager as both *computer1* and *computer1.company.com*. In this case, you must run the tool once for each configured host name. With chosen options, you are prompted for the host name of the directory server containing configuration data. For example:

```
$ modifyldapbindpasswd -i /opt/oam/identity -c is -t all
Please enter host machine for Configuration DS : YourConfigDSName
```

See the information on using host name variations in the *Oracle Access Manager Identity and Common Administration Guide.*

Modifying the Password in the Directory Server Itself: Both the old and new passwords are stored in the Oracle Access Manager configuration files. The old password continues to be valid until you have completed all updates. To perform this task, you must run the tool with the -x and -y options as shown here:

```
$ modifyldapbindpasswd -i /opt/oam/identity -c is -t all -x oldpassword -y
newpassword Please enter host machine for Configuration DS :  YourConfigDSName
```

In addition to the parameters that are required for each operation in the procedures provided here, you can add additional parameters and options to suit your needs.

> **See Also:** "Parameters for the ModifyLDAPBindPasswd Tool" on page 6-5

Generating the Encrypted Password File

Oracle recommends you use an encrypted file to provide the updated password. However, you can supply the password interactively when running the tool. During this procedure you are asked to provide information through a number of questions.

When you supply the name of the encrypted file to generate, the following specifications are acceptable:

```
-genpasswdfile test
-genpasswdfile test.xml
-genpasswdfile /home/oracle/test
-genpasswdfile /home/oracle/test.xml
```

In the filename, an .xml extension is provided automatically if you do not supply one. A separator after the filename (/ on UNIX or \ on Windows) produces an error.

Incorrect:

```
-genpasswdfile test/
-genpasswdfile /home/oracle/test/
```

Oracle recommends that you read and respond to all prompts presented during the procedure. The following procedure illustrates steps that you perform from a Windows platform.

> **Note:** On UNIX systems, the steps are the same; however, the tool name does not include the .exe extension. Path names might vary in your environment.

To generate the encrypted password file

1. Access the ModifyLDAPBindPasswd tool from the following directory:

   component_install_dir/oblix/tools/modbinpasswd/ModifyLDAPBindPasswd

   Where *component_install_dir* is the installation directory for the component for which you are updating the directory bind password.

2. Run the ModifyLDAPBindPasswd command with the following options:

   ```
   modifyldapbindpasswd.exe -i install_dir -c component -t target -genpasswdfile
   filename
   ```

   Here *file* is the name of a password file. For example, you might enter *myfile*, which generates myfile.xml.When you do not supply an .xml extension, it is supplied automatically. See also Table 6–1, " Required Parameters for the ModifyLDAPBindPasswd Tool".

3. Follow the prompts that appear to fill in the appropriate information for your deployment. For example:

```
Please enter bind password for Configuration DS : your_bind_pw

Is DS information to be updated same as Config DS? : 1(Y) 2(N) :1

Please enter New bind password for DS for which the password is updated :
new_bind_pw

Please confirm New bind password for DS for which the password is updated :
new_bind_pw
```

4. Review the confirmation message that identifies the location and name of the generated file. For example:

```
Password file created: /home/oracle/myfile.xml
```

Updating the LDAP Bind Password for Configuration Data

In this procedure, the steps that you must perform depend on whether policy data and configuration data are stored in separate directory servers or together. You can use the ModifyLDAPBindPasswd tool or create a script. Details about using a script are included in the following procedure.

> **See Also:** "About the ModifyLDAPBindPasswd Tool and Logs" on page 6-5

Oracle recommends that you read and respond to all prompts presented during the procedure. The following procedure illustrates steps that you perform from a Windows platform.

> **Note:** On UNIX systems, the steps are the same; however, the tool name does not include the .exe extension. Path names might vary in your environment.

To update the LDAP bind password for configuration data

1. **Using a Script**: Generate the password file as described in "To generate the encrypted password file" on page 6-9.

2. Access the ModifyLDAPBindPasswd tool from the following directory:

   IdentityServer_install_dir/oblix/tools/modbinpasswd/

   Where *IdentityServer_install_dir* is the installation directory for the first Identity Server for which you are updating the directory bind password.

3. **First Identity Server**: Run the command for one Identity Server instance using the following options:

   ```
   modifyldapbindpasswd.exe -i install_dir -c is -t all -options
   ```

   For other *options*, see Table 6–1 and Table 6–3.

   Using a Script: Pass the encrypted password file using the `-j` *`file_for_bind_password option.`*

4. **Remaining Identity Server, Policy Managers, Access Servers**: Locate the tool in the appropriate path for each remaining instance, and run the tool using these options:

   ```
   modifyldapbindpasswd.exe -i install_dir -c nn -t file -options
   ```

   Where *nn* represents the appropriate component ID from Table 6–1.

   Using a Script: Pass the encrypted password file using the `-j file_for_bind_password` option, as described in Table 6–3.

5. Repeat this command for every variant of the host name that you have configured in Oracle Access Manager. For example:

   ```
   modifyldapbindpasswd.exe -i install_dir -c is -t all
   Please enter host machine for Configuration DS : YourConfigDSName
   ```

   See the information on using host name variations in the *Oracle Access Manager Identity and Common Administration Guide*.

6. Update the bind password for the directory server that stores the configuration data. For example, you can provide details interactively when you include the -i, -c, and -t options:

   ```
   modifyldapbindpasswd.exe -i install_dir -c is -t all -x oldpassword -y
   newpassword
   Please enter host machine for Configuration DS :  YourConfigDSName
   ```

Updating the LDAP Bind Password for User Data

In this procedure, the steps that you must perform depend on whether policy data and configuration data are stored in separate directory servers or together. You can use the ModifyLDAPBindPasswd tool or create a script. Details about using a script are included in the following procedure.

Different steps require different parameters. Pay close attention to the parameters that are required for the step that you are performing. You can include additional options as long as you provide the required parameters.

> **See Also:** "About the ModifyLDAPBindPasswd Tool and Logs" on page 6-5

Oracle recommends that you read and respond to all prompts presented during the procedure. The following procedure illustrates steps that you perform from a Windows platform.

> **Note:** On UNIX systems, the steps are the same; however, the tool name does not include the .exe extension. Path names might vary in your environment.

To update the LDAP bind password for user data

1. **User Data and Configuration Data in Same Directory Server**: Follow the procedure "To update the LDAP bind password for configuration data" on page 6-10.

> **Note:** If you have already updated the LDAP bind password for configuration data, you are finished. If user data and configuration data are in different directory servers, see Step 2.

2. **User Data and Configuration Data in Different Directory Server**s: Access the ModifyLDAPBindPasswd tool from the following directory, and perform remaining steps:

   component_install_dir/oblix/tools/modbinpasswd/

   Where *component_install_dir* is the installation directory for the component for which you are updating the directory bind password.

3. **Using a Script**: Generate the password file as described in "Generating the Encrypted Password File" on page 6-9.

4. Run the command with the following parameters:

   ```
   modifyldapbindpasswd.exe -i install_dir -c nn -t ds -options
   ```

   For other *options*, see Table 6–1 and Table 6–3.

   Using a Script: Pass the encrypted password file using the `-j file_for_bind_password` option, as described in Table 6–3.

5. Repeat this command for every variant of the host name that you have configured in Oracle Access Manager. For example:

   ```
   modifyldapbindpasswd.exe -i install_dir -c is -t all
   Please enter host machine for Configuration DS : YourConfigDSName
   ```

   See the information on using host name variations in the *Oracle Access Manager Identity and Common Administration Guide.*

6. Update the bind password for the directory server that stores the configuration data. For example, to supply the password interactively:

   ```
   modifyldapbindpasswd.exe -i install_dir -c is -t all -x oldpassword -y
   newpassword
   Please enter host machine for Configuration DS :  YourConfigDSName
   ```

Updating the LDAP Bind Password for Policy Data

In this procedure, the steps that you must perform depend on whether policy data and configuration data are stored in separate directory servers or together. You can use the ModifyLDAPBindPasswd tool or create a script. Details about using a script are included in the following procedure.

> **See Also:** "About the ModifyLDAPBindPasswd Tool and Logs" on page 6-5

Oracle recommends that you read and respond to all prompts presented during the procedure. The following procedure illustrates steps that you perform from a Windows platform.

> **Note:** On UNIX systems, the steps are the same; however, the tool name does not include the .exe extension. Path names might vary in your environment.

To update the LDAP bind password for policy data

1. **Policy Data and Configuration Data in Same Directory Server**: Perform steps in "To update the LDAP bind password for configuration data" on page 6-10.

   > **Note:** If you have already updated the LDAP bind password for configuration data, you are finished. If policy data and configuration data are in different directory servers, see Step 2.

2. **Policy Data and Configuration Data in Different Directory Servers**: Access the ModifyLDAPBindPasswd tool from the installation directory of the component for which you are updating the directory bind password and perform remaining steps:

   component_install_dir/oblix/tools/modbinpasswd/

   Using a Script: Generate the password file as described in "To generate the encrypted password file" on page 6-9.

3. **First Identity Server**: Run the command following for the Identity Server instance:

   ```
   modifyldapbindpasswd.exe -i install_dir -c is -t ds -options
   ```

   For other *options*, see Table 6–1 and Table 6–3.

   Using a Script: Pass the encrypted password file using the `-j file_for_bind_password` option, as described in Table 6–3.

4. **Remaining Identity Servers, Policy Managers, Access Servers**: Run the following command for the remaining instances of the Policy Manager and the Access Server:

   ```
   modifyldapbindpasswd.exe -i install_dir -c nn -t file -options
   ```

   For other *options*, see Table 6–1 and Table 6–3.

   Using a Script: Pass the encrypted password file using the `-j file_for_bind_password` option, as described in Table 6–3.

5. Repeat this command for every variant of the host name that you have configured in Oracle Access Manager. For example:

   ```
   modifyldapbindpasswd.exe -i install_dir -c is -t all
   Please enter host machine for Configuration DS : YourConfigDSName
   ```

   See the information on using host name variations in the *Oracle Access Manager Identity and Common Administration Guide.*

6. Update the bind password for the directory server that stores the configuration data. For example:

   ```
   modifyldapbindpasswd.exe -i install_dir -c is -t all -x oldpassword -y
   newpassword
   Please enter host machine for Configuration DS :  YourConfigDSName
   ```

Changing the LDAP Bind Password When Running in ADSI Mode

If the Identity Server or Access Server uses an explicit bind, you can follow the procedures to change the LDAP bind password described in previous topics. In this case, you can skip this procedure.

If you are running the Identity Server or Access Server as a user in the Active Directory domain, you update the LDAP bind password as described in the following procedure.

To update the LDAP bind password for ADSI

1. Change the password for the user in the Active Directory domain.
2. Stop the Identity Server or Access Server.
3. From the command line enter the following:

   ```
   services.msc
   ```

   A dialog box appears that lists of all running services.
4. To change the credential of the user in the service, right-click the service to modify, then click Properties; click the Log On tab, then click This Account.
5. Restart the Identity Server or Access Server.

7

Synchronizing System Clocks Across Time Zones

Correct operation of Oracle Access Manager depends on synchronizing the system clocks for all of its main components.

This chapter includes the following topics:

- About Synchronization
- Synchronization With NTP
- Synchronization with a GPS-based System
- About Daylight Savings Time

> **Note:** This chapter provides a general discussion of NTP. It is provided for informational purposes only. Follow your own company's guidance for installing and configuring NTP.

About Synchronization

As discussed in *Oracle Access Manager Installation Guide*, if you plan to install Oracle Access Manager components across multiple computers, you must make sure all system clocks are synchronized. This is particularly important if you are running the software in Cert or Simple mode.

Synchronization is important for normal operations. Extremely accurate synchronization can also be a factor in security. For example, a time-based attack can be performed by changing the time on an expired cookie so that it appears to be earlier than the real time. Closely synchronized computers make it difficult to forge the timestamp on a cookie

Synchronization With NTP

The Network Time Protocol (NTP) is a commonly-used tool for synchronizing system clocks. The following URL provides information on time synchronization.

`http://www.ntp.org/`

Also, see the comp.protocols.time.ntp news group for information on time synchronization.

NTP can typically synchronize the time on computers to within a few milliseconds. The following example shows the output of an ntp command on a typical workstation in an uncontrolled office environment. The example shows the high degree of synchronization that is achieved with this command:

```
 ntpq -p
   remote           refid           st t when poll reach   delay   offset    disp
==============================================================================
-qa.mycompany.co    clock.via.net    2 u  228 1024  377     1.33    0.121    5.13
#palantir.mycomp    clock.via.net    2 u  254 1024  377     1.42   -1.518    5.12
-panacea.company    clock.via.net    2 u  244  256  377     0.91    0.551    3.31
+test.mycompany.    nist1.aol-ca.tr  2 u  175  256  376     0.96    3.760    5.41
+test.mycompany.    pra3a.mycompany  3 u  441  256  372     1.12    3.043   65.31
+test.mycompany.    pra3a.mycompany  3 u  232  256  377     0.81    3.736    2.85
+test.mycompany.    pra3a.mycompany  3 u   27  256  377     0.93    3.787    3.34
+test2.mycompany    nist1.aol-ca.tr  2 u  232  256  377     0.74    3.722    2.92
*nist1.abc-ca.tr .ACTS.           1 u  180  256  377    11.53    1.097    2.88
-ntp-cup.externa .GPS.            1 u   96  256  377    38.48   -0.694    4.45
```

The offset field is in milliseconds. Note that all of these computers are within 5 milliseconds of the same time. The nist1 workstation is about 1 millisecond slower (1.097 milliseconds) than the time that the U.S. National Institute of Standards provides. This compares favorably with some radio broadcasts, which can be limited to approximately 10 millisecond accuracy due to varying atmospheric propagation delays.

UNIX operating systems typically ship with a version of NTP. It takes a small amount of configuration to enable these shipped versions:

- **Solaris:** Ceate an ntp.conf file.

  After you create this file using the Solaris conventions, xntp is started automatically when the computer is restarted.

- **HP-UX:** Use sam to start ntp.
- **AIX:** Create an ntp.conf file and enable or create a start script in /etc/init.d (or the equivalent directory on AIX).

For all versions of UNIX, you can also get a current (and more secure) version of the NTP daemon from `http://www.ntp.org/`.

All UNIX computers use UTC (the pedant's name for GMT) internally and convert to the local time for displaying the time to users.

Windows computers typically perform time synchronization automatically with their domain controller using a Microsoft version of NTP. While NTP can synchronize the times, you must also synchronize the domain controller with an official time source.

You can obtain a time service from many Internet Service Providers (ISPs). There is a list of open stratum-1 servers available from `http://www.ntp.org/`. Some of the servers that are listed at this site are open, for example, the servers at NIST. Other servers require an e-mail request before you use them to synchronize your network.

Windows computers keep the clock in local time, but the NTP synchronization programs compensate to convert to the appropriate time in each time zone.

Synchronization with a GPS-based System

If having the best possible time match is important to your organization, you can purchase GPS-based clocks. The less expensive ones require some assembly. These clocks can be used to set your entire network to the same time. GPS technology requires very accurate times. Each GPS satellite contains 3 atomic clocks with continuous corrections provided from the ground to compensate for relativistic effects.

In other words, an accurate estimate of the current time is developed as a side effect of determining where the GPS receiver is.

About Daylight Savings Time

As discussed in the *Oracle Access Manager Installation Guide*, Oracle Access Manager relies on synchronized time clocks and each host computers' Operating System to correctly manage time. When the Operating System time clock is operating properly, Oracle Access Manager operates properly. Usually, network time protocol (NTP) is used to manage and synchronize Operating System time clocks.

> **Note:** Time management includes changes for daylight savings time. Daylight savings time changes have no impact on Oracle Access Manager.

USA 2007 Daylight Saving Time (DST) Compliance for Oracle Database and Oracle Fusion Middleware Products: In calendar year 2007, the effective dates for daylight savings are going to change. In the United States, the Energy Policy Act of 2005 was signed into law to extend daylight saving time. Under the new rules, DST in the U.S. starts on the second Sunday in March and end the first Sunday in November. In the past, daylight savings time started on the first Sunday in April and ended the last Sunday in October.

Under the new rules for 2007, DST starts on March 11, 2007 and end on November 04, 2007. This change also affects Canada. Unless the required patches are applied, the database may report incorrect time zone data between March 11, 2007 and April 1, 2007 and between October 28, 2007 and November 4, 2007 (and on different dates in subsequent years). Mexico is still using the old DST rules.

For more information about the impact of USA 2007 DST compliance for Oracle Database and Oracle Fusion Middleware products, see Note: 397281.1 on the My Oracle Support (formerly MetaLink) Web Site: `https://metalink.oracle.com`.

US 2007 DSTChanges For Oracle Internet Directory and Oracle Application Server: only the database has potential DST issues with the 2007 DST change, and then only if timezones are set up. A compliant Operating System is needed. For more information, review the following notes on the My Oracle Support (formerly MetaLink) Web Site: `https://metalink.oracle.com`.

- Note 357056.1—Impact of changes to daylight saving time (DST) rules on the Oracle database
- Note 359145.1—Impact of 2007 USA daylight saving changes on the Oracle database
- Note 360803.1—AU Timezone Database and Fusion Middleware Recommendations
- Note 397281.1—USA 2007 Daylight Saving Time (DST) Compliance for Database and Fusion Middleware
- Note 401010.1—Western Australia Daylight Saving Time Changes Database and Fusion Middleware Recommendations

To locate knowledge base articles on My Oracle Support (formerly MetaLink)

1. Go to `https://metalink.oracle.com`.
2. Log in as directed.

3. Click the **Knowledge** tab.
4. From the Quick Find list, choose **Knowledge Base**, enter the *number* of the note, click the **Go** button.
5. From the results list, click the name of the note you want to view.

8 About Upgrading

The term *upgrade* refers to the process of installing the latest major product release over an earlier product release (whether the earlier release has been patched or not). This is known as an in-place upgrade.

The latest release provides significant enhancements and regulatory compliance over previous releases. For example, each major release provides new features and additional platform support, and may include changes to the schema, data, parameter, or message files.

The *Oracle Access Manager Upgrade Guide* provides detailed strategies and procedures for upgrading your deployment and ensuring backward compatibility with earlier custom plug-ins and AccessGates.

9

Oracle Access Manager Backup and Recovery Strategies

Oracle recommends that you periodically backup Oracle Access Manager files and data from time to time so that you can recover from any unforeseen event and restore your Oracle Access Manager system. Topics in this chapter include:

- About Backup and Recovery Strategies
- Backup Recommendations
- Back Up Strategies for Deployment Events
- Recovery Strategies

About Backup and Recovery Strategies

This section introduced concepts and strategies for back up and recovery.

The term recovery describes a process where you can perform certain steps to undo an event, or change, and return to earlier data or an earlier status. Recovery might be as simple as modifying an entry in the System Console. However when the System Console is not involved, recovery strategies can be successful only when you have performed appropriate backup tasks.

In any deployment, it is important to make a back up copy from time to time. However, it is your own company policies that determine the backup schedule for information within an Oracle Access Manager deployment. Depending on your deployment scenario (development, staging, or production, for example) and the business requirements for sustainability, you may be required to make monthly, weekly, or even daily backups of component filesystem directories or configuration and policy data.

Backing up Oracle Access Manager-related data helps you prepare for any unintended situation that may arise. For example, you can:

- Restore an LDAP directory snapshot if Oracle Access Manager data becomes inconsistent or is corrupted as a result of changes that are external to Oracle Access Manager.

  For information about using Oracle Access Manager Configuration Manager to create snapshots of the `oblix` tree of the LDAP environment, see the *Oracle Access Manager Configuration Manager Installation and Administration Guide*. Using LDAP tools to export data from and import it to an LDAP directory or database are outside the scope of this manual.

- Roll back to undo everything that you have done and return to the starting point (or to the last back up copy). For example, when you are upgrading to a later release, you can roll back all changes and return to your earlier Oracle Access Manager release.

  Oracle recommends that you create a back up copy of product directories, files, and configuration data before *and* after upgrading to a later Oracle Access Manager release. For more information, see the *Oracle Access Manager Upgrade Guide*.

- Revert (roll back) the changes made during data migration using Oracle Access Manager Configuration Manager. For more information about migrating data between Oracle Access Manager deployments, and data migration transactions, see the *Oracle Access Manager Configuration Manager Installation and Administration Guide*.

For specific backup recommendations, see the next topic. For additional information, see "Back Up Strategies for Deployment Events" on page 9-4.

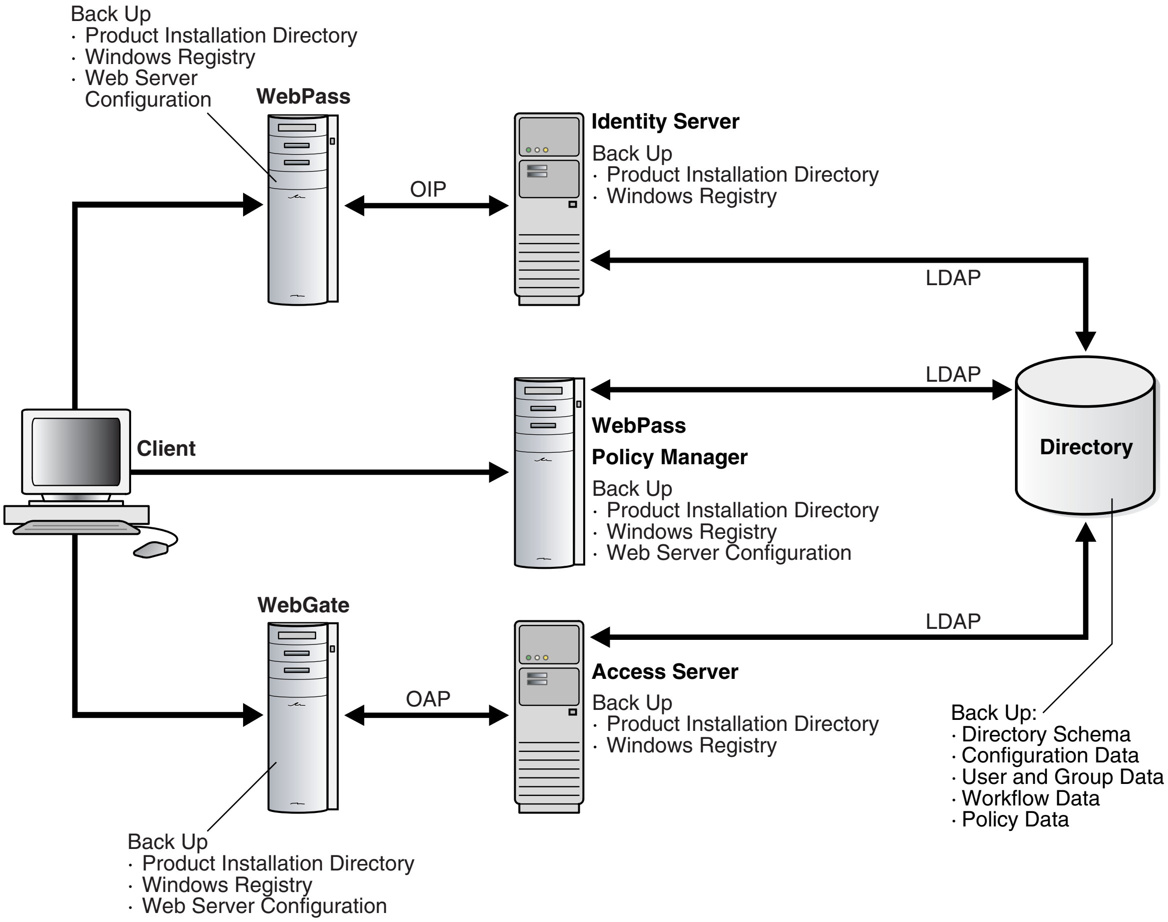
Backup Recommendations

When you consider that Oracle Access Manager is a distributed solution, there are multiple backup requirements. For instance:

- Every Oracle Access Manager component installation directory on each computer host should be backed up at a file level (Identity Server, WebPass Policy Manager, Access Server, WebGate)
- Any custom plug-ins should be backed up at a file level
- Critical configuration details stored in an LDAP directory or database should be backed up using vendor tools. Alternatively, you can use Oracle Access Manager Configuration Manager to create a snapshot of the `oblix` tree of the LDAP directory as described in the *Oracle Access Manager Configuration Manager Installation and Administration Guide*.

Figure 9–1 illustrates a simple Oracle Access Manager deployment and the data that Oracle recommends you back up. If you have only the Identity System installed, you can ignore details for the Policy Manager, Access Server, and WebGate.

Figure 9–1 Oracle Access Manager Deployment Back Up Strategy



As illustrated in Figure 9–1, each Oracle Access Manager component installation directory in the filesystem includes the following types of information:

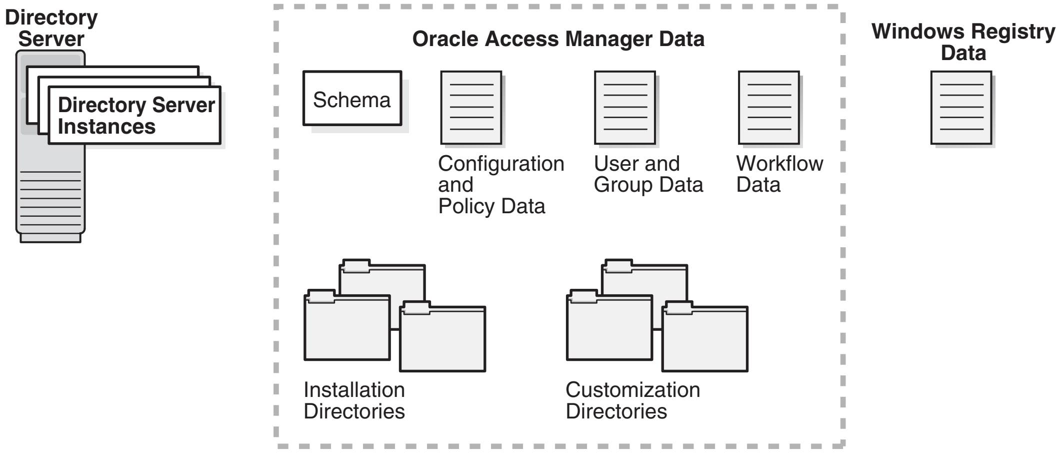
- Program and library files
- Message and parameter catalogs
- Component-specific configuration files, which may include:
  - Failover configuration files
  - Stylesheets
  - Software developer kit (SDK) configurations

In addition to backing up every Oracle Access Manager component installation directory in the filesystem, Oracle recommends that you backup:

- **Customizations**: Independent filesystem directories that contain customized Oracle Access Manager plug-ins and stylesheets
- **Updated Web Server Configurations**: Web server configuration files that were updated to operate with Oracle Access Manager Web components
- **Windows Systems**: The Windows Registry on each Windows system that is hosting an Oracle Access Manager component

Figure 9–2 further illustrates the Oracle Access Manager deployment data that should be backed up, which includes the directory server instance.

Figure 9–2 Oracle Access Manager Data to Back Up


Directory Server
Directory Server Instances
Oracle Access Manager Data
Schema
Configuration and Policy Data
User and Group Data
Workflow Data
Installation Directories
Customization Directories
Windows Registry Data

The directory server (or database) instance for an Oracle Access Manager deployment should be backed up. The information that is stored in the `oblix` tree of the directory server (or database) includes the Oracle Access Manager:

- Schema (directory objects and attributes specific to Oracle Access Manager)
- Configuration data
- User and group data
- Workflow data

  You can archive processed workflow instances and filter out transient data such as workflow tickets.
- Access policy data

Again, you can either use directory or database vendor tools to extract policy and configuration data in the `oblix` tree or use Oracle Access Manager Configuration Manager to create a snapshot of the `oblix` tree.

Back Up Strategies for Deployment Events

As part of the deployment planning process, Oracle recommends that your and your team review the following information to plan an appropriate backup strategy for specific deployment tasks.

Oracle recommends that you make a full and complete backup of specific directories and data in the following situations:

- Immediately before installing Oracle Access Manager in a production environment. For more information, see "Backing Up Before Oracle Access Manager Installation" on page 9-5.
- Immediately after installing and setting up Oracle Access Manager in a production environment. For more information, see "Backing Up After Oracle Access Manager Installation" on page 9-5.
- Immediately before, and after, applying policy changes to Oracle Access Manager deployments. For information about using Oracle Access Manager Configuration Manager to create snapshots of the `oblix` tree of the LDAP environment, see the *Oracle Access Manager Configuration Manager Installation and Administration Guide*.

- Immediately after customizing Oracle Access Manager, as described in "Backing Up After Customizing Oracle Access Manager" on page 9-5.
- Immediately before upgrading from an earlier Oracle Access Manager release component to a later release. For more information, see "Backing Up Before Upgrading" on page 9-6.
- Immediately after upgrading from an earlier Oracle Access Manager release component to a later release. For more information, see "Backing Up After Upgrading" on page 9-6.

Backing Up Before Oracle Access Manager Installation

To assist with recovery strategies, Oracle recommends that you back up critical information immediately before installing and setting up Oracle Access Manager.

To back up critical information before installing Oracle Access Manager

1. **Existing Web Server Configuration**: Back up the existing Web server configuration file before installing Oracle Access Manager Web components. Use instructions from your Web server vendor.
2. Back up any LDAP directory server instances before you start installing Oracle Access Manager. Use instructions from your directory server vendor to accomplish this task.
3. **Windows**: Back up existing Windows Registry data.

Backing Up After Oracle Access Manager Installation

To assist with recovery strategies after installing Oracle Access Manager, Oracle recommends that you back up critical information immediately after installing and setting up Oracle Access Manager and verifying that it is operating properly.

To back up critical information after installing a new component

1. Back up the newly installed Oracle Access Manager component directory in the filesystem and store the back up copy in a new location.
2. **Updated Web Server Configuration**: Back up the updated Web server configuration file for Oracle Access Manager Web components using instructions from your Web server vendor.
3. **Windows**: Back up the Windows Registry for each component.
4. Copy configuration and policy data in the `oblix` tree (or use Oracle Access Manager Configuration Manager to create a snapshot of the `oblix` tree).

Backing Up After Customizing Oracle Access Manager

Oracle recommends that you back up customization information (plug-ins, stylesheets, and the like) immediately after verifying that it is operating properly.

To back up customizations

1. Create a backup your customization filesystem directory and store it in a different location.
2. Copy all customization files and sub directories, as follows:

   From: *customizations_dir*

To: *backup_customizations_dir*

Backing Up Before Upgrading

Oracle recommends that you perform certain back up activities before upgrading from an earlier Oracle Access Manager release component to a later release. This enables you to restore an earlier environment in the unlikely event that you want to do this following an upgrade. Table 9–1 provides more information; full details are provided in the *Oracle Access Manager Upgrade Guide*.

Table 9–1 Backup Strategies Before Upgrading

Back Up the Following	As Described in The Following Sections of the *Oracle Access Manager Upgrade Guide*
Oracle Access Manager Schema	Backing up the Earlier Oracle Access Manager Schema
Oracle Access Manager Configuration and Policy Data	Backing up Oracle Access Manager Configuration and Policy Data
Oracle Access Manager User and Group Data	Backing Up User and Group Data
Oracle Access Manager Workflow Data	Backing Up Workflow Data
Processed Workflows	Archiving Processed Workflow Instances
Existing Directory Instances	Backing Up Existing Directory Instances
Earlier Installed Component Directory (and any Customization Directories)	Backing Up the Existing Installed Directory
Web Server Configuration Files	Backing Up the Existing Web Server Configuration File
Windows Registry	Backing Up Windows Registry Data

Backing Up After Upgrading

After you have completed and verified each component upgrade, Oracle recommends that you back up the upgraded information as outlined in Table 9–2. This enables you to restore an upgraded environment to the newly upgraded status should this be needed. For specific details, see the *Oracle Access Manager Upgrade Guide*.

Table 9–2 Backup Strategies After Upgrading

Back Up the Following	As Described in the *Oracle Access Manager Upgrade Guide*
Existing Directory Instances	Backing Up Existing Directory Instances
Earlier Installed Component Filesystem Directory (and any Customization Directories)	Backing Up the Existing Installed Directory
Web Server Configuration Files	Backing Up the Existing Web Server Configuration File
Windows Registry	Backing Up Windows Registry Data

Recovery Strategies

The following topics provide information about recovery strategies to use in various situations:

- Recovery Strategies After Installation
- Recovery Strategies During Upgrades

Recovery Strategies After Installation

If you encounter a problem during installation and want to roll back to your original installation before trying again, you can perform the following tasks.

To recovery critical information after installing Oracle Access Manager

1. Uninstall Oracle Access Manager as described in the *Oracle Access Manager Installation Guide*.
2. **Web Server Configuration**: Restore your original Web server configuration file using instructions from your Web server vendor.
3. Restore the original LDAP directory server instances that were backed up before you started installing Oracle Access Manager. Use instructions from your directory vendor to accomplish this task.
4. **Windows**: Restore the original Windows Registry.

Recovery Strategies During Upgrades

Should something unlikely occur and you find that an upgrade process did not complete successfully, you may use the strategies in Table 9–3 to recover. For specific details, see the *Oracle Access Manager Upgrade Guide*

Table 9–3 Upgrade Recovery Strategies

Task	What to do If the Task Fails
Backing Up Existing Oracle Access Manager Data	Retry this task using instructions in Chapter 5, "Preparing for Schema and Data Upgrades" in the *Oracle Access Manager Upgrade Guide*.
Backing Up Existing Directory Instances	See your directory vendor documentation.
Adding An Earlier Identity System to Use as a Master (against Read/Write master directory instances, not against read-only replicas) Note: You use this additional earlier setup as a master when upgrading the schema and data to ensure that your existing installation is not affected should any issues arise.	Retry this task using instructions in Chapter 5, "Preparing for Schema and Data Upgrades" in the *Oracle Access Manager Upgrade Guide*.
Adding an Earlier Access Manager to Use as a Master (against Read/Write master directory instances, not against read-only replicas) Note: You use this additional earlier setup as a master when upgrading the schema and data to ensure that your existing installation is not affected should any issues arise.	Retry this task using instructions in Chapter 5, "Preparing for Schema and Data Upgrades" in the *Oracle Access Manager Upgrade Guide*.
Upgrading Identity System Schema and Data	Restore the directory instance you backed up before starting this upgrade (see "Backing Up Existing Directory Instances" in the *Oracle Access Manager Upgrade Guide*.). Locate your backup copy of the earlier master Identity Server installation directory (made before the upgrade) and make another backup copy. You retain one to use as a backup and use the other when you retry the upgrade. See "Backing Up Directories, Web Server Configurations, and Registry Details" in the *Oracle Access Manager Upgrade Guide*. Retry the upgrade of the master Identity Server using instructions in Chapter 6, "Upgrading Identity System Schema and Data" in the*Oracle Access Manager Upgrade Guide*.

Table 9–3 (Cont.) Upgrade Recovery Strategies

Task	What to do If the Task Fails
Enabling Multi-Language Capability when upgrading the master Identity Server from a starting release of 6.1.1. Note: This process does not occur when your starting release is 6.5 or 7.x because those releases automatically supported multi-language capability.	Restore the directory instance you backed up before starting this upgrade (see "Backing Up Existing Directory Instances" in the *Oracle Access Manager Upgrade Guide*.). Locate your backup copy of the earlier master Identity Server installation directory (made before the upgrade) and make another backup copy. You retain one to use as a backup and use the other when you retry the upgrade. See "Backing Up Directories, Web Server Configurations, and Registry Details" in the *Oracle Access Manager Upgrade Guide*. Retry the upgrade of the master Identity Server using instructions in Chapter 6, "Upgrading Identity System Schema and Data" in the*Oracle Access Manager Upgrade Guide*.
Upgrading Access System Schema and Data	Restore the directory instance you backed up before starting this upgrade (see "Backing Up Existing Directory Instances"). Locate your backup copy of the earlier master Access Manager installation directory (made before the upgrade) and make another backup copy. You retain one to use as a backup and use the other when you retry the upgrade. See "Backing Up Directories, Web Server Configurations, and Registry Details" in the *Oracle Access Manager Upgrade Guide*. Retry the upgrade of the master Access Manager using instructions in Chapter 7, "Upgrading Access System Schema and Data" in the*Oracle Access Manager Upgrade Guide*.
Uploading Directory Server Index Files	Retry this task using instructions in a "Uploading Directory Server Index Files" in the *Oracle Access Manager Upgrade Guide*.
Upgrading Components: Upgrading an earlier version of any Oracle Access Manager component (Identity Server, WebPass, Policy Manager (formerly known as the Access Manager component)), Access Server, or WebGate). Note: Schema and data upgrades occur only when upgrading master components added for this purpose.	Locate your backup copy of the earlier component installation directory (made before the upgrade) and make another backup copy. You retain one to use as a backup and use the other when you retry the upgrade. See "Backing Up Directories, Web Server Configurations, and Registry Details" in the *Oracle Access Manager Upgrade Guide*. Retry this step and specify the earlier component installation directory when asked for the installation directory. See Part III, "Upgrading Components" in the *Oracle Access Manager Upgrade Guide*.
Upgrading Your Identity System Customizations	Retry this task using instructions in Chapter 12, "Upgrading Your Identity System Customizations" in the *Oracle Access Manager Upgrade Guide*.
Upgrading Your Access System Customizations	Retry this task using instructions in Chapter 13, "Upgrading Your Access System Customizations" in the *Oracle Access Manager Upgrade Guide*.

Additional information on recovering from an upgrade failure can be found throughout the *Oracle Access Manager Upgrade Guide*.

Index