

Oracle® Access Manager

Identity and Common Administration Guide

10g (10.1.4.2.0)

B32419-01

August 2007

This book explains how to configure Identity System applications to display information about users, groups, and organizations; how to assign permissions to users for viewing and modifying the data that is displayed in the Identity System applications; and how to configure workflows that link together Identity application functions into a chain of automatically performed steps. This book also describes functions that are common to the Identity and Access Systems, for example, directory profile configuration, password policy configuration, logging, and auditing.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	xvii
Audience	xvii
Documentation Accessibility	xvii
Related Documents	xviii
Conventions	xix
 What's New in Oracle Access Manager?	xxi
Product and Component Name Changes	xxi
Globalization	xxiii
Password Policies and Lost Password Management	xxiii
Configuring Multiple Searchbases	xxiv
Configuring Groups to Accept Subscriptions	xxiv
Configuring Workflows	xxiv
Configuring Group Manager	xxiv
Configuring Org. Manager	xxiv
Auditing	xxv
Logging	xxv
Configuring the Directory Server	xxv
Active Directory	xxvi
Troubleshooting	xxvi
 Part I Introducing Oracle Access Manager Administration	
 1 Preparing for Administration	
Prerequisites	1-1
About Identity System Configuration and Administration	1-1
Identity System Components	1-2
Review of Identity System Installation and Setup	1-3
About Configuring the Identity System	1-3
About Managing the Identity System	1-5
Introduction to Using Oracle Access Manager	1-5
Login	1-6
Logging In to the Identity System	1-6
Logging into the Access System	1-7
Functional Areas on a page	1-8

Navigation Elements	1-8
Search Functionality	1-9
The Selector	1-10
Online Help	1-11
The About Page Link	1-11
Logging Out.....	1-11
To log out	1-11

2 Specifying Identity System Administrators

About Identity System Administrators.....	2-1
Specifying Administrators	2-4
Deleting Administrators	2-4
Delegating Administration	2-5
About Delegating Administration.....	2-5
Delegated Administration Models	2-6
Extranet Model	2-6
Intranet Model	2-7
ASP Model	2-8
Adding Delegated Administrators.....	2-8
Adding Substitute Administrators	2-10

Part II Configuring the Identity System

3 Making Schema Data Available to the Identity System

About Object Classes.....	3-1
About Sending Data to External Systems Using Template Objects.....	3-2
The Process for Configuring Schema Data.....	3-2
Objects Configured During Installation	3-3
Structural and Auxiliary Object Classes in the Identity System	3-3
Template Object Classes.....	3-4
Object Class Types	3-5
Viewing Object Classes.....	3-5
Modifying Object Classes	3-6
Selecting a Class Attribute	3-7
Changing the Structural Object Class	3-8
Adding Object Classes	3-8
How Auxiliary Classes Are Used	3-9
Deleting Object Classes	3-9
About Object Class Attributes.....	3-10
About Configuring Attributes.....	3-10
Attribute Data Types	3-10
Attribute Semantic Types.....	3-11
Semantic Types Defined During Setup.....	3-12
Semantic Types Used in Profile Pages	3-12
Semantic Types Used in the Group Manager	3-13
Location Coordinates Semantic Type	3-13

Semantic Types for Managing Lost Passwords	3-13
Other Semantic Types	3-14
Attribute Display Types	3-14
Viewing Attributes	3-16
Configuring Attributes	3-17
Using Rules and Lists	3-19
Defining a Rule.....	3-19
Defining a List	3-20
Localizing Attribute Display Names.....	3-20
Search Filters for the Object Selector Display Type	3-22
Creating a Search Filter for the Object Selector Display Type.....	3-22
Search Filters for Multiple Target Object Classes.....	3-23
Deleting a Search Filter.....	3-23
Usage of Rules and Filters.....	3-23
Static LDAP Search Filters	3-24
Static Searches Using Wild Cards.....	3-24
Static Searches Using Multiple Target Object Classes.....	3-24
Substitution Syntax: Returning Targets that Match the DN of the Logged In User	3-25
Examples of Dynamic LDAP Search Filters.....	3-25
Dynamic Searches Using Wild Cards	3-26
Dynamic Searches Using Multiple Values	3-26
Use of the Not Operator.....	3-26
Configuring Other Display Types	3-27
Configuring Derived Attributes: Matching Values from Different Attributes	3-27
Example of a Derived Attribute	3-28
Assigning a Derived Attribute to a User Manager Tab.....	3-29
Permissions for Derived Attributes	3-30
Attributes Configured for an Individual Application	3-30

4 Configuring User, Group, and Organization Manager

About User, Group, and Organization Manager	4-2
Configuring Tabs	4-2
Viewing and Modifying Tab Configuration Information	4-3
Localizing Tabs.....	4-5
Adding a Tab to the Organization Manager	4-6
Specifying the Search Attributes on a Tab.....	4-6
Viewing, Modifying, and Localizing Attributes that Appear in Search Results	4-7
Adding Auxiliary and Template Object Classes to a User or Org. Manager Tab	4-8
Adding Auxiliary and Template Object Classes to a Group Tab.....	4-9
Configuring Group Manager Tab Options.....	4-10
Deleting a Tab in Organization Manager	4-11
Ordering the Tabs in Organization Manager.....	4-11
Configuring Tab Profile Pages and Panels	4-12
Use of LDAP and Template Objects on a Panel.....	4-12
Configuring the Header Panel	4-12
Viewing Panels That You Have Configured in the End User Application	4-13
Adding, Modifying, Localizing, and Deleting a Panel.....	4-14

LDAP and Template Attributes on a Panel.....	4-14
Multiple Languages on a Panel.....	4-14
Creating a Panel	4-15
Viewing, Modifying, and Localizing a Panel.....	4-16
Ordering the Panels	4-18
Viewing Group Type Panels.....	4-18
Adding, Modifying, Localizing, and Deleting a Group Type Panel	4-19
Modifying and Localizing Attributes Displayed on a Panel	4-21
Allowing Users to View and Change LDAP Data	4-22
About the Searchbase	4-23
Guidelines for Setting the Searchbase.....	4-23
If You Need to Modify a Searchbase.....	4-24
Indexing and the Searchbase	4-24
Indexing Requirements for Oracle Internet Directory	4-24
Setting the Searchbase	4-25
If You Set a Searchbase for a Group	4-28
Configuring and Deleting Disjoint Searchbases.....	4-28
Writing LDAP Filters Using Query Builder.....	4-29
Methods for Retrieving Matches	4-30
Building Advanced LDAP Filters Using QueryBuilder	4-31
About View and Modify Permissions.....	4-32
Setting and Modifying LDAP Attribute Permissions	4-33
Keys for Selecting Multiple Attributes.....	4-35
Evaluation of LDAP Attribute Permissions	4-36
Examples of Configuring an Application.....	4-36
Displaying Photos in User Profiles.....	4-36
Importing and Storing Photos in a Directory	4-36
Referencing Photos in a File System	4-38
The Default Photo Image	4-38
Enabling the Location Tab in Organization Manager	4-38
About Assigning the Right to Create Groups in Group Manager	4-39
Configuring a Group to Accept Subscriptions	4-39
End-User Scenarios for the Group Manager.....	4-40
Managing Group Members in Group Manager	4-40
Searching for Group Members.....	4-41
Deleting Group Members	4-42
Adding Group Members.....	4-42
Managing Group Subscriptions	4-43
Subscribing to Groups.....	4-44
Configuring Auditing Policies	4-45
Viewing Auditing Policies	4-45
Modifying Auditing Policies	4-45
Generating Reports	4-46
Configuring Reports	4-46
Viewing, Modifying, Localizing, and Deleting Reports.....	4-48
Advanced Configuration	4-49
Expanding Dynamic Groups.....	4-49

Modifying the Default Searchbase Scope	4-50
Simplified Attribute Permissions for a Group	4-51
Implementing Simplified Permissions.....	4-51
Sample gscacparams.xml File	4-51
Simplified Permissions Reserved Words	4-52
Setting Container Limits in Organization Manager.....	4-53
Copying Container Limits	4-55
Modifying Container Limits.....	4-55

5 Chaining Identity Functions Into Workflows

About Workflows	5-1
How Workflows Are Initiated.....	5-2
Typical Workflow Examples	5-2
Advanced Workflow Options	5-2
Workflow Types.....	5-3
Creating Workflows.....	5-4
How Users Access Workflows in an Identity System Application.....	5-4
About Workflow Tickets.....	5-5
A Workflow Scenario	5-6
LDAP Versus Template Attributes in a Workflow	5-7
Workflow Types, Steps, and Actions	5-7
About Workflow Steps	5-8
About Step Actions	5-10
Descriptions of Step Actions.....	5-13
About Subflows	5-15
Using the QuickStart Tool	5-16
Creating a Self-Registration Workflow Using the Quickstart Tool	5-19
Using the Workflow Applet	5-19
Starting a New Workflow Definition	5-20
Defining an LDAP Target for Create Object Workflows.....	5-22
Defining the First Step in a Workflow	5-24
Defining Step Attributes	5-26
Defining Subsequent Steps	5-29
Committing Workflow Steps.....	5-30
Enabling a Workflow	5-30
Testing a Workflow.....	5-31
Example of Defining a Workflow	5-31
Defining a Subflow	5-32
Associating a Subflow with a Workflow	5-32
Approving Subflow Steps.....	5-33
Advanced Workflow Ticket Routing	5-33
Configuring Workflow Actions for Advanced Ticket Routing.....	5-34
About Notifying Newly Assigned Step Participants.....	5-34
Specifying Dynamic Participants.....	5-35
About Workflow Participants	5-35
About Workflow Ticket Routing	5-35
About Dynamic Participants	5-36

About Static Participants	5-36
About the Static Participants Not Available Button	5-37
Enabling Dynamic Participants	5-37
Specifying Surrogates	5-41
Enabling Time-based Escalation	5-43
Performing Asynchronous Operations	5-46
Notes on Asynchronous Workflows	5-47
Using a Workflow	5-47
Invoking a Workflow	5-48
Finding and Processing a Ticket	5-48
Deactivating and Reactivating Users	5-49
Reactivating a Deactivated User	5-50
Monitoring a Workflow	5-51
Archiving Requests.....	5-51
Deleting Requests.....	5-52
Preventing Other Administrators from Working on a Workflow Ticket	5-52
Managing Workflows	5-53
Viewing and Exporting a Workflow Summary	5-53
Copying a Workflow	5-54
Modifying a Workflow	5-55
Deleting a Workflow.....	5-55
Exporting Workflows	5-56
Viewing Workflow Panel Settings.....	5-56
Modifying the Appearance of Workflow Panels.....	5-57
Localizing Workflow Panels.....	5-58
Workflow Performance	5-59
The Identity Administrator's Modify Rights.....	5-59
Advanced Workflow Options	5-60
Pre and Post Actions.....	5-60
External Actions	5-60
Customization of Data and Actions in a Workflow	5-60
Adding Roles to a Workflow	5-61
Creating a Self-Registration Workflow.....	5-62
Creating a Location Workflow	5-64

6 Sending Non-LDAP Data to External Applications

About Configuring Non-LDAP Data	6-1
Summary of Using Non-LDAP Data in a Workflow	6-2
About Template Objects	6-3
About Template Object Data and Workflows.....	6-3
Object Template Configuration	6-4
Format of the Object Template File	6-4
How Template Objects Appear in the Identity System.....	6-5
Elements in an Object Template File	6-6
Sample Object Template File	6-8
Creating an Identity Event Plug-In for Template Attributes.....	6-9

7 Configuring Global Settings

Configuring Styles for Identity System Applications	7-2
Viewing a Style.....	7-2
Adding a Custom Style Directory	7-3
Deploying a Style	7-5
Changing a Style Name.....	7-6
Modifying a Style	7-6
Deleting a Style.....	7-6
Setting the Default Style.....	7-7
Configuring Multiple Languages for Oracle Access Manager.....	7-7
Selecting a Language for Administrative Pages	7-8
Language Evaluation Order for End-User Applications.....	7-8
Configuring Identity Server Settings.....	7-9
Configuring Session Timeout.....	7-10
Customizing Email Destinations	7-11
Configuring a Mail Server	7-12
Managing Caches	7-13
Managing Multiple Languages	7-14
Managing Identity Servers.....	7-14
Setting Up Multiple Identity Servers	7-14
Adding an Identity Server	7-15
Viewing and Modifying Identity Server Parameters.....	7-18
Deleting Identity Server Parameters	7-18
Managing an Identity Server Service from the Command Line	7-19
Managing Directory Server Profiles.....	7-19
About LDAP Directory Server Profiles.....	7-20
Locations for Directory Server Profiles.....	7-21
Creating an LDAP Directory Server Profile.....	7-22
Viewing an LDAP Directory Server Profile	7-28
Modifying an LDAP Directory Server Profile	7-29
Rerunning Setup Manually.....	7-29
Rerunning Identity System Setup.....	7-30
Rerunning Policy Manager Setup.....	7-30
Reconfiguring the Access Server	7-31
Adding Database Instances to LDAP Directory Server Profiles	7-31
LDAP Referrals.....	7-32
Deleting an LDAP Directory Server Instance	7-34
Working With Multiple Directory Searchbases.....	7-34
Managing RDBMS Profiles.....	7-36
Adding or Modifying an RDBMS Profile	7-36
Adding or Modifying an RDBMS Database Instance	7-39
Configuring WebPass	7-40
Viewing a Configured WebPass	7-40
Adding or Modifying a WebPass	7-41
Removing a WebPass	7-43
Modifying a WebPass from a Command Line.....	7-43
Managing Associations Between Identity Servers and WebPass	7-45

To view Identity Servers associated with a WebPass	7-45
To modify an Identity Server's connections to a WebPass	7-46
To associate an Identity Server with a WebPass	7-46
To configure the number of retries to a non-responsive Identity Server.....	7-46
Disassociating a WebPass from an Identity Server.....	7-47
Configuring the WebPass Poll Tracking Parameter	7-47
Configuring Password Policies	7-49
Order of Password Policy Evaluation.....	7-50
Managing Password Policies.....	7-51
Viewing Password Policies.....	7-51
Setting the Defaults or Global Settings for Different Types of Password Policies	7-51
Creating Password Policies for a Specific Domain	7-53
Modifying Password Policies.....	7-56
Deleting a Password Policy	7-57
Lost Password Management	7-57
Syntax for the Lost Password Management URL	7-59
About Presenting Challenge Phrases to Users	7-59
About Other Aspects of the Challenge and Response Page	7-59
How the User Experiences Lost Password Management with Multiple Challenges	7-59
Viewing and Configuring Lost Password Management Policies	7-60
Implementing Password Policies in the Access System.....	7-64
Modifying Authentication Schemes to Include a Password Policy.....	7-64
Configuring Password Redirect URLs	7-65
Configuring Redirection to a Password Reset Page After Password Expiry	7-65
Setting Up Password Expiry Warning Redirect URLs	7-67
Setting Up Redirect URLs for Account Lockout	7-68
Updates to the Access Server Cache.....	7-68
Configuring the Access Manager SDK for the Identity System	7-69
Cloned and Synchronized Components	7-70

Part III Performing Common Administrative Tasks

8 Changing Transport Security Modes

About Transport Security Modes	8-1
Transport Security Mode Between Components	8-2
About CA Certificates.....	8-4
Changing Transport Security for the Identity System	8-5
Transport Security Mode Changes for the Identity System.....	8-6
Changing to Simple Transport Security Mode	8-6
Changing to Cert Transport Security Mode.....	8-7
Changing Transport Security Modes for the Access System	8-9
Transport Security Mode Changes for the Access System.....	8-10
Changing to Open Transport Security Mode.....	8-12
Changing to Simple Transport Security Mode	8-13
Changing to Cert Transport Security Mode.....	8-15
Transport Security Changes for Directory Servers	8-19
Changing Transport Security Passwords.....	8-21

Importing Multiple CA Certificates	8-23
Changing Access Server Security Password	8-24

9 Reporting

About Reporting	9-1
Report Types	9-2
Data Sources.....	9-3
Data Output	9-3
Output Configuration.....	9-4
Data Uses.....	9-4
Summary of Reporting Features	9-4

10 Logging

About Logging, Log Levels, and Log Output	10-1
About Log Levels	10-2
About Capturing Stack Trace Data in the Log Files.....	10-4
About Log Output.....	10-4
About Log Configuration File Paths and Contents	10-4
Log Configuration File Paths and Names	10-5
Log Configuration File Contents	10-5
When Changes to the File Take Effect	10-5
About Comments in the Log File.....	10-6
About Directing Log Output to a File or the System File	10-8
Structure of the Log Configuration File	10-9
The Log Configuration File Header	10-10
The Initial Compound List.....	10-10
The Simple List, Logging Threshold, and Identity System Console Synchronization	10-10
The Second Compound List and Log Handlers	10-11
The List for Per-Module Logging	10-12
About XML Element Order	10-12
About Activating and Suppressing Logging Levels	10-13
About Log Handler Precedence.....	10-13
Log Configuration Parameters	10-14
Settings in the Default Log Configuration File	10-15
Description of the Settings in the Default Log Configuration File	10-16
Configuring Logs in the Identity System Console	10-17
Configuring Different Threshold Levels for Different Types of Data	10-19
About the MODULE_CONFIG Section	10-20
Location of the Per-Module Logging Section in the Log Configuration File.....	10-21
List of Modules That Can Be Logged.....	10-21
Configuring a Log Level Threshold for a Function or Module.....	10-24
Logging the Amount of Time to Process Requests	10-25
About the Calls Captured in the Request Timing Logs.....	10-26
About Interpreting Request Times in the Logs.....	10-26
Configuring the Log Files to Capture Time Data for External Requests	10-27

11 Auditing

About Auditing	11-1
Audit Output Considerations	11-2
Audit Security Considerations	11-2
Audit Performance Considerations	11-2
Static Audit Reports	11-3
Dynamic Audit Reports	11-4
Controlling Audit Output.....	11-4
About Audit Options.....	11-4
Auditing Requirements	11-7
Requirements for Auditing.....	11-7
Requirements for Auditing to a Database	11-8
Special Components for Database Auditing.....	11-8
Updates to Supported Versions and Platforms	11-9
Audit-to-Database Architecture	11-9
About OCI Settings	11-11
OCI Configuration for Oracle Access Manager 10.1.4.0.1.....	11-11
OCI Configuration for Oracle Access Manager 10.1.4.2.....	11-12
About ODBC Data Source Definitions	11-12
About ODBC Drivers.....	11-13
About the Windows ODBC Driver	11-14
About RDBMS Profiles for Database Auditing	11-14
About Profiles For Databases That Use an ODBC Connection Type.....	11-14
About Profiles For Databases That Use an OCI Connection Type	11-14
About the Audit Database	11-15
About the Crystal Repository.....	11-15
About Audit Reports	11-15
Setting Up File-Based Auditing	11-17
Setting Up Database Auditing	11-19
Setting Up Your System for Database Auditing.....	11-19
About installing SQL Server (Windows).....	11-20
Creating the Audit Database	11-21
Tuning the Audit Schema	11-22
Uploading the Audit Schema.....	11-23
Enabling Access and Identity Servers to Connect to the Audit Database	11-29
Configuring Auditing.....	11-35
Setting up Audit Reports	11-44

12 SNMP Monitoring

Prerequisites	12-1
About Oracle Access Manager SNMP Monitoring and Agents	12-1
The SNMP Agent	12-2
About the Oracle Access Manager MIB and Objects	12-3
MIB Index Fields	12-3
Identity Server MIB Objects.....	12-4
Access Server MIB Objects.....	12-8
About Integration with Oracle Enterprise Manager 10g Identity Management	12-12

Enabling and Disabling SNMP Monitoring	12-13
Setting Up SNMP Agent and Trap Destinations	12-14
Changing SNMP Configuration Settings	12-15
Logging for SNMP	12-17
SNMP Messages	12-17
Discrepancies Between Netstat and SNMP Values	12-22
Configuring the Shutdown Interval	12-22

Part IV Appendices

A Deploying with Active Directory

Setting Up Directory Profiles and Searchbases	A-1
Defining Directory Server Profiles for Remaining Domains	A-2
Setting Up Disjoint Searchbases	A-2
About Deleting a Disjoint Searchbase	A-3
Configuring Group-Search Read Operations (Optional)	A-3
Authentication and Authorization with Active Directory	A-4
Parent-Child Authentication	A-4
Parent-Child Authorization	A-5
ObMyGroups Action Attribute	A-5
Configuring the credential_mapping Plug-In	A-6
Configuring Single Sign-On for Use with Active Directory	A-7
About Search Filters	A-8
About the Length of the SAMAccountName	A-9
Configuring for .NET Features	A-9
Troubleshooting	A-9
Microsoft Resources	A-9

B Configuring for ADSI

About ADSI with Oracle Access Manager	B-1
Recommendation	B-2
Identity System ADSI Configurations	B-2
Pure ADSI with ADSI Authentication	B-2
Mixed ADSI with LDAP Authentication	B-3
Bind Mechanisms for the Identity Server	B-3
Oracle Access Manager ADSI Configuration Files	B-4
About globalparams	B-4
About adsi_params	B-5
Access System ADSI Configurations	B-6
Pure ADSI with ADSI Authentication	B-6
Access System ADSI Configuration Files	B-7
Configuring ADSI for the Identity System	B-9
Enabling ADSI for a Default Directory Profile	B-9
Enabling ADSI for Other Directory Profiles	B-9
Configuring ADSI for the Access System	B-11
Changing the pageSize Parameter	B-12

Troubleshooting.....	B-12
C Configuring for Active Directory with LDAP	
Overview	C-1
Setting Up the Policy Manager for LDAP	C-2
Setting Up the Access Server for LDAP.....	C-3
Setting Active Directory Timeouts for LDAP.....	C-3
Enabling LDAP Authentication with ADSI	C-4
D Implementing .NET Features	
Resolving Ambiguous Names	D-1
About ANR Attributes, Searches, and Results	D-2
Configuring for ANR.....	D-2
Updating Configuration Data	D-3
Configuring ANR in Identity System Panels.....	D-3
Verifying ANR Attribute Access Control.....	D-4
Using ANR in Identity System Searches	D-5
Configuring for Dynamically Linked Auxiliary Classes	D-5
Adding Attributes Dynamically	D-6
Adding Attributes for a Group	D-7
Enabling Fast Bind for Access System Authentication.....	D-8
Enabling Impersonation	D-9
Setting Up Integrated Windows Authentication	D-10
Enabling IWA on the WebGate Web Server	D-11
Configuring the WebGate for IWA	D-12
Creating an IWA Authentication Scheme in Oracle Access Manager	D-12
Testing IWA Implementation.....	D-13
Using Access System Password Management.....	D-13
Using Managed Code and Helper Classes	D-13
Integrating with Authorization Manager Services.....	D-14
Integrating with Smart Card Authentication.....	D-14
Integrating the Security Connector for ASP.NET	D-14
Troubleshooting.....	D-15
Microsoft Resources.....	D-15
E Oracle Access Manager Parameter Files	
File Categories.....	E-1
For More Information on the Parameter Files	E-1
F Troubleshooting Oracle Access Manager	
Directory and Database Problems and Solutions	F-1
Problem Setting up SSL Between the Identity Server and Active Directory.....	F-1
Error Message to Check if the Directory Server is Running or Responding.....	F-2
Access Control and Searchbase Support for eDirectory 8.7.3.....	F-2
Problem	F-2
Solution.....	F-3

Unable to Save a Directory Server Profile	F-3
Problem	F-3
Solution.....	F-3
Active Directory: Adding Members Causes the Group Size to Shrink.....	F-3
ADSI Cannot Be Enabled for a Directory Profile	F-4
Problem	F-4
Solution.....	F-4
Database Validation Fails.....	F-4
Problem	F-4
Solution.....	F-4
Other Problems and Solutions	F-5
Auditing for the Identity System Ceases to Work.....	F-5
Problem	F-5
Solution.....	F-5
Identity Server Crashes if It Cannot Find a Stylesheet	F-5
Problem	F-5
Solution.....	F-5
Identity System Deletes a User Entry When an RDN Is Modified	F-6
Problem	F-6
Solution.....	F-6
JPEG Photo Images Are Not Updated	F-6
Problem	F-6
Solution.....	F-6
Memory Usage Rises for an Identity Server After Configuring a Directory Server Profile...	F-7
Problem	F-7
Solution.....	F-7
Pass Phrase for Simple Transport Security Has Been Lost	F-7
Performance of a Component Performance is Slow	F-7
Problem	F-7
Solution.....	F-8
Reports With Non-ASCII Characters Are Not Imported Correctly in Excel.....	F-8
Problem	F-8
Solution.....	F-8
Simple Transport Security Mode Expires After One Year	F-8
Problem	F-8
Solution.....	F-8
Stylesheet Validation Fails.....	F-9
Problem	F-9
Solution.....	F-9
User Creation Might Fail When You Have Multi-byte Charcters in the Password	F-9
WebPass Is Unable to Connect to Its Associated Identity Server	F-10
Problem	F-10
Solution.....	F-11
Error Messages and Recommendations for Handling Them.....	F-11
"Cannot Find xenroll.cab" Error Is Issued When Using a Workflow	F-11
Problem	F-11
Solution.....	F-11

"Enable Failed" Error Is Issued When Using a Workflow	F-11
Problem	F-11
Solution.....	F-12
"There is No Profile Configured for This Kind of Object" Error Is Issued.....	F-12
Problem	F-12
Solution.....	F-12
"Warning: Page Has Expired" Error Is Issued.....	F-12
Problem	F-12
Solution.....	F-12
Capturing Diagnostic Information	F-13
About Oracle Access Manager Diagnostics	F-13
Collecting Diagnostic Data	F-13
Interpreting Diagnostic Output	F-16
Interpreting Diagnostic Output When the list Parameter Is Used	F-16
Interpreting Diagnostic Output When the detail Parameter Is Used.....	F-17
Interpreting the Diagnostic Data for Caches.....	F-19
Automatically Writing Stack Traces to a Log File After a Core Dump.....	F-21
Manually Requesting a Stack Trace.....	F-21
Need More Help?	F-22

Index

Preface

There are two administration guides for Oracle Access Manager. This book, *Oracle Access Manager Identity and Common Administration Guide*, provides information on configuring Oracle Access Manager to read and make use of data in your directory, configuring Identity applications to display directory data, assigning read and write permissions to users, and defining workflows that link together Identity application functions into a sequence of automatically initiated steps. This guide also describes functionality that is common to both the Access System and Identity System. Common functionality includes configuring directory servers and password policies.

Note: Oracle Access Manager was previously known as Oblix NetPoint. Some items, such as schema objects, paths, and so on may still use the term "oblix" or "NetPoint."

This Preface covers the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This guide is intended for the administrators assigned during Oracle Access Manager installation and setup, as well as Master Identity Administrators and Delegated Identity Administrators. Administrators configure the rights and tasks available to other administrators and end users. A Master Administrator, the highest level administrator, is selected during Identity System setup. This administrator delegates responsibilities to other administrators, as described in this book.

This document assumes that you are familiar with your LDAP directory and Web servers.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see the following documents in the Oracle Access Manager Release 10g (10.1.4.0.1) 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 Application Server Release Notes*—Read these for the latest Oracle Access Manager updates. The release notes are available with the platform-specific documentation. The most current version of the release notes is available on Oracle Technology Network at:
<http://www.oracle.com/technology/documentation>.
- *Oracle Access Manager Patchset Notes Release 10.1.4 Patchset 1 (10.1.4.2.0) For All Supported Operating Systems*. It provides the system requirements and instructions needed to install or de-install the Patchset itself, a list of known issues related to the patchset, a list of the platform-specific bugs fixed in this Oracle Access Manager Patchset.
- *Oracle Access Manager List of Bugs Fixed Release 10.1.4 Patchset 1 (10.1.4.2.0)*. It supplements the Patchset notes document for this release. It provides a list of all generic (common to all operating systems) Oracle Access Manager bugs that have been fixed in this Patchset, sorted by component.
- *Oracle Access Manager Installation Guide*—Describes how to install and set up the Oracle Access Manager components.
- *Oracle Access Manager Upgrade Guide*—Explains how to upgrade earlier releases to the latest major Oracle Access Manager release.
- *Oracle Access Manager Access System 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 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 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 third-party products such as BEA WebLogic, Siebel 7, and IBM Websphere.
- *Oracle Access Manager Schema Description*—Provides details about the schema.
- *Oracle Access Manager Configuration Manager Installation and Administration Guide*—Provides information about pushing configuration data changes from one Oracle Access Manager 10g (10.1.4.0.1), or Oracle COREid Release 7.0.4, deployment to another. For example, when pushing changes from a development deployment to a pre-production deployment. Included are considerations, prerequisites, and step-by-step instructions to help ensure your success.

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	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 Oracle Access Manager 10g (10.1.4.2) and provides pointers to additional information within this book. Information from previous releases is also retained to help those users migrating to the current release.

The following sections describe the new features in Oracle Access Manager that are covered in this book:

- [Product and Component Name Changes](#)
- [Globalization](#)
- [Password Policies and Lost Password Management](#)
- [Configuring Multiple Searchbases](#)
- [Configuring Workflows](#)
- [Configuring Group Manager](#)
- [Configuring Org. Manager](#)
- [Auditing](#)
- [Logging](#)
- [Configuring the Directory Server](#)
- [Troubleshooting](#)

Note: For a comprehensive list of new features and functions in Oracle Access Manager 10g (10.1.4.2), and a description of where each is documented, see the chapter on What's New in Oracle Access Manager 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

Item	Was	Is
Product Name	Oblix SHAREid NetPoint SAML Services	Oracle Identity Federation
Product Name	OctetString Virtual Directory Engine (VDE)	Oracle Virtual Directory
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
Access System Service	AM Service State	Policy Manager API Support Mode

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

Globalization

- As part of the globalization support, some file formats have changed from the proprietary .lst format to .xml

Oracle Access Manager 10g Release 3 (10.1.4) has undergone a globalization process to provide multibyte support that enables processing of internationalized data and messages in the user's native language.

password.xml; globalparams.xml; obscoreboard; AppDBfailover.xml and AppDB.xml; ConfigDBfailover.xml and ConfigDB.xml; WebResrcDBfailover.xml -- now WebResrcDB.xml; snmp_agent_config_info.xml; obscoreboard_params.xml

See Also: References to these file names in this manual.

- Oracle Access Manager uses a locale-based case insensitive sorting method when you click the column heading (Full Name, for example) in the search results table.

See Also: ["Search Functionality"](#) on page 1-9.

- In the Identity System Console, some display names are displayed incorrectly if the locale of the browser is different from the locale of the characters used in the display name.

See Also: ["Attribute Display Types"](#) on page 3-14, ["Multiple Languages on a Panel"](#) on page 4-14.

- When generating a report for an Identity application, save the report file as .txt and re-import it for the characters to display correctly.

See Also: ["Viewing, Modifying, Localizing, and Deleting Reports"](#) on page 4-48.

Password Policies and Lost Password Management

Password policies and Lost Password Management have been enhanced.

- You can configure the minimum and maximum number of characters users can specify in a password.

See Also: ["Managing Password Policies"](#) on page 7-51.

- For lost password management, you can set multiple challenge-response pairs, create multiple stylesheets, and configure other aspects of the user's lost password management experience.

See Also: ["Lost Password Management"](#) on page 7-57.

- You can redirect users back to the originally requested page after resetting a password.

See Also: ["Configuring Redirection to a Password Reset Page After Password Expiry"](#) on page 7-65.

- You can enable users to access resources without re-authenticating after resetting a password.

See Also: ["Configuring Redirection to a Password Reset Page After Password Expiry"](#) on page 7-65.

Configuring Multiple Searchbases

- This book contains expanded information on configuring Oracle Access Manager for multiple directory searchbases, also called disjoint domains or realms.

See Also: ["Working With Multiple Directory Searchbases"](#) on page 7-34.

Configuring Groups to Accept Subscriptions

- Information has been added on enabling legacy groups to accepting user subscriptions.

See Also: ["Configuring a Group to Accept Subscriptions"](#) on page 4-39.

Configuring Workflows

- This book contains expanded information on configuring workflows for dynamic targets.

You can dynamically assign a user to a target on a create user workflow. For example, you can define a create user workflow that enables user A to log in under ou=users, invoke the workflow, and create user B whose entry is automatically determined to be in the same ou as user A. This ability always existed in the Identity System, and is now explicitly documented in the chapter on workflows.

- The section on the QuickStart tool now mentions that only Master Administrators can use the QuickStart tool.

See Also: ["Starting a New Workflow Definition"](#) on page 5-20, ["Defining an LDAP Target for Create Object Workflows"](#) on page 5-22, and ["Using the QuickStart Tool"](#) on page 5-16.

Configuring Group Manager

- You may receive an error when viewing or managing a static nested group. Administrators can now disable processing of nested static groups for performance reasons.

See Also: ["Managing Group Members in Group Manager"](#) on page 4-40, ["Searching for Group Members"](#) on page 4-41, ["Allowing Users to View and Change LDAP Data"](#) on page 4-22, ["Writing LDAP Filters Using Query Builder"](#) on page 4-29.

Configuring Org. Manager

- The section on configuring panel now notes that you should configure at least one panel for Org. Manager. This is required to enable users to save profile data.

See Also: ["Adding, Modifying, Localizing, and Deleting a Panel"](#) on page 4-14.

Auditing

- You can now audit to an Oracle Database as well as to Microsoft SQL Server. Support for MySQL is deprecated in this release.

The Crystal Reports package is no longer provided with the Oracle Access Manager package. You must obtain this product from the vendor.

See Also: ["Auditing"](#) on page 11-1.

Logging

- Changes to logging parameters take effect within one minute, rather than requiring you to restart the server where the changes were made.

See Also: ["Logging"](#) on page 10-1.

- If Oracle Access Manager experiences a core dump, a stack trace is automatically written to the log file if you have enabled logging.

See Also: ["Logging"](#) on page 10-1 and ["Capturing Diagnostic Information"](#) on page F-13.

- For keeping log output concise while diagnosing particular problems, you can configure different log level thresholds for different modules within a log configuration file. For example, if you want to diagnose slow response times for an Identity Server's LDAP directory, you would only be interested in detailed logs for LDAP operations.

See Also: ["Configuring Different Threshold Levels for Different Types of Data"](#) on page 10-19.

- For capacity planning and performance tuning purposes, you can log the time it takes to process calls to external components. For example, when performing capacity planning, you may want to know what calls to the directory server are taking the most time.

See Also: ["Logging the Amount of Time to Process Requests"](#) on page 10-25.

Configuring the Directory Server

- When you configure SSL mode for the directory server, only server authentication is supported. Client certificates are not supported.

See Also: ["Transport Security Mode Between Components"](#) on page 8-2.

- The default value for the Maximum Session Time of 0 (no maximum) can cause LDAP caches to become too large. The recommended value is 600 (10 hours).

See Also: ["Creating an LDAP Directory Server Profile"](#) on page 7-22.

Active Directory

- The `samAccountNameLength` parameter enables you to increase the number of characters permitted as a `SamAccountName` attribute value. For Active Directory environments that are running in native mode, you may want to increase the default value for this parameter.

See Also: ["About the Length of the SAMAccountName"](#) on page A-9.

Troubleshooting

- Information on troubleshooting that was dispersed throughout this manual has been consolidated in a separate appendix.

See Also: ["Troubleshooting Oracle Access Manager"](#) on page F-1.

- You can now write diagnostic information to a log file and collect stack traces.
The Access Server and Identity Server provide diagnostic tools to help you work with an Oracle Technical Support representative to troubleshoot problems. These tools are not for day-to-day administration. Their purpose is to help you investigate problems that require assistance from Oracle Technical Support.

See Also: ["Capturing Diagnostic Information"](#) on page F-13.

- New troubleshooting topics have been added.

See Also:

"Access Control and Searchbase Support for eDirectory 8.7.3" on page F-2

""There is No Profile Configured for This Kind of Object" Error Is Issued" on page F-12

"Unable to Save a Directory Server Profile" on page F-9

"Error Message to Check if the Directory Server is Running or Responding" on page F-2

"Active Directory: Adding Members Causes the Group Size to Shrink" on page F-3

"ADSI Cannot Be Enabled for a Directory Profile" on page F-4

"Stylesheet Validation Fails" on page F-9

"Identity Server Crashes if It Cannot Find a Stylesheet" on page F-5

"Identity System Deletes a User Entry When an RDN Is Modified" on page F-6

"WebPass Is Unable to Connect to Its Associated Identity Server" on page F-10

"Simple Transport Security Mode Expires After One Year" on page F-8,

"JPEG Photo Images Are Not Updated" on page F-6

""Enable Failed" Error Is Issued When Using a Workflow" on page F-11

""Cannot Find xenroll.cab" Error Is Issued When Using a Workflow" on page F-11

"Reports With Non-ASCII Characters Are Not Imported Correctly in Excel" on page F-8

"Auditing for the Identity System Ceases to Work" on page F-5

Part I

Introducing Oracle Access Manager Administration

Before you begin working with Oracle Access Manager, it is important to understand basic Oracle Access Manager administration concepts.

Part I provides an introduction to Oracle Access Manager administration and contains the following chapters:

- [Chapter 1, "Preparing for Administration"](#)
- [Chapter 2, "Specifying Identity System Administrators"](#)

Preparing for Administration

Before configuring and administering Oracle Access Manager, you may find it useful to preview the tasks that you perform as an administrator. It may also be useful to log in and view the user interface for the Identity System and Access System.

This chapter contains information you need before starting to configure and administer Oracle Access Manager, including these topics:

- [Prerequisites](#)
- [About Identity System Configuration and Administration](#)
- [Introduction to Using Oracle Access Manager](#)

Note: Although the product name has changed to Oracle Access Manager, in manuals and the product you may see the name NetPoint or Oblix. This is particularly true in file and path names.

Prerequisites

Oracle Access Manager should be installed and set up as described in the *Oracle Access Manager Installation Guide*. Read the *Oracle Access Manager Introduction* which provides an overview of Oracle Access Manager not found in other manuals.

This document focuses on Identity System administration as well as common configuration and administration tasks.

About Identity System Configuration and Administration

You use the Identity System and objects in the directory service to manage identity information about individuals, groups, organizations, and other objects. The Master Administrator can delegate authority to other administrators, allowing the Identity System to scale millions of users.

In addition to managing identity information, the Identity System enables you to manage read, write, and modify privileges for a user based on a specific user attribute, membership in a group, or association with an organization. You can link privileges together into a workflow.

For example, you can set up a self-registration workflow so that when a user self-registers, the registration request is forwarded to appropriate people for approval, and upon approval, the user is immediately and automatically granted access to all resources appropriate for his or her identity attributes.

Finally, the Identity System enables you to accurately manage user identities, group memberships, and organizational objects. This information can then be leveraged by the Access System to manage access privileges for users based on user attributes, group membership, or association with an organizational entity.

Identity System Components

The Identity System consists of these components:

- The Identity Server
- WebPass

Identity Server—A standalone server or several instances that manage identity information about users, groups, organizations, and other objects. The Identity Server provides the following applications:

- *User Manager*—If you are an administrator or a user, the User Manager enables you to add, modify, and delete user identities provided that you are a participant in a workflow that performs this function. User Manager data can be leveraged by the Access System to provide users with access privileges based on their directory profiles. The User Manager also has reporting capability.

The User Manager typically enables end users to view other users and to modify their own identity information. The users that a person can view and the identity information that someone can modify depends on the privileges granted by a Master Administrator.

- *Group Manager*—Enables administrators and users to create or delete groups, and enables users to subscribe or unsubscribe from groups. You must be a participant in a workflow that performs the desired function. The Group Manager also has reporting capability.

The Group Manager typically enables end users to view groups and to subscribe to membership in a group. The groups that a person can view and subscription rights are granted by a Master Administrator.

- *Organization Manager*—If you are an administrator or a user, the Organization Manager enables you to create and delete organizations and other objects (such as floor plans and assets) that do not belong in the User Manager or Group Manager. You must be a participant in a workflow that performs the desired function. The Organization Manager also has reporting capability.

The Organization Manager enables end users to view organizational entities such as floor plans. The organizational entities that a person can view depend upon the rights granted by a Master Administrator.

- *Identity System Console*—Enables administration and configuration of the Identity System. Using the System Console, you also create Administrators and assign the right to delegate administrative tasks.

The Identity Server stores user information on a directory server. The Identity Server keeps the directory current so that the Access Server gets the right information.

WebPass—WebPass is a Web server plug-in that passes information between the Web server and the Identity Server. WebPass can talk to multiple Identity Servers.

Details here include:

- [Review of Identity System Installation and Setup](#)
- [About Configuring the Identity System](#)

- [About Managing the Identity System](#)

Review of Identity System Installation and Setup

Installation and setup includes the following events:

- At least one Identity Server and one WebPass were installed and the resulting Identity System was set up.
- A transport security mode was chosen to protect communication between the Identity Server and WebPass.
- The Identity Server was configured to communicate with an LDAP directory server or virtual directory.

You are prompted regarding automatic setup of your directory server schema. If you chose not to automatically update your schema, you are prompted to do so manually during configuration. Instructions on manual updates of your directory server schema are provided in this manual.

- Each expected application was installed with the Identity Server.

When you log in to the Identity System, you see a series of tabs on the top navigation bar that match your applications. From these tabs you can configure the look and functionality of the User Manager, Group Manager, and Organization Manager applications.

- Required attributes for the User and Group object classes were set up.

Other attributes may also have been configured.

- At least one Master Administrator was selected.

This is the highest-level administrator. You must have at least one administrator defined to begin working with Oracle Access Manager. These are the people who configure the System. The Master administrator creates lower-level administrators called Master Identity Administrators.

[Table 1-1](#) provides a review of Identity System installation and setup.

For more information, see the *Oracle Access Manager Installation Guide*.

About Configuring the Identity System

The Identity System consists of an administrative console and three end-user applications discussed earlier:

- Identity System Console (includes User Manager Configuration, Group Manager Configuration, Org. Manager Configuration, Common Configuration, and System Configuration)
- The User Manager application
- The Group Manager application
- The Organization Manager application

People use the Identity System end-user applications for tasks such as changing personal information, resetting passwords, adding other users, and looking up organizational information. This identity data originates in your LDAP directory. To configure the Identity System applications, you need to know what attributes in the directory you want to display, and what attributes you want to be able to modify.

After configuring the Identity System to work with data in your directory, you configure the Identity System application profile pages. These profile pages display

the directory data. For example, you can display a user's name, title, address, and phone number on a profile page in the User Manager application. You can also improve the efficiency of your organization by using Identity workflows. Identity workflows enable you to automate Identity System application-related activities, for example, creating a user and assigning email and other accounts to that user.

Finally, you use the Identity System to create Identity workflows. Identity workflows are definitions for a set of actions and the steps you perform to complete the actions. For instance, you can create workflow definitions for the way new employees are added to your various corporate information systems.

[Table 1–1](#) provides an overview of configuring the Identity System:

Table 1–1 Overview of Identity System Configuration

To perform this task	Description	Read
Specify additional structural object classes for the Organization Manager and auxiliary object classes for all applications	During setup, you configure one structural object class each for the User Manager, Group Manager, and Organization Manager. You can define additional structural objects classes for the Organization Manager. For instance, you may want the Organization Manager to display assets. You can also add auxiliary object classes to provide the Identity System applications with data.	"About Object Classes" on page 3-1
Configure attributes	You can determine what attributes are available to the User, Group, and Organization Manager applications. You also can configure rules for how to display attribute values on an Identity System application profile page. For example, you may want employees to be able to select their department name from a list.	"About Object Class Attributes" on page 3-10.
Configure User, Group, and Organization application tabs	In the User Manager, you configure what the user sees on the My Identity tab. In the Group Manager, you configure what the user sees on the My Groups tab. In the Organization Manager, you configure what the user sees on the Location tab and, optionally, additional tabs.	"Viewing and Modifying Tab Configuration Information" on page 4-3.
Configure User, Group, and Organization profile pages	Tabs contain one or more profile pages. A profile page contains a set of panels. A panel is a collection of attributes. For example, on a profile page for a user, you can define an Identity panel to display values for attributes such as Name, Photo, Title, and so on.	"Configuring Tab Profile Pages and Panels" on page 4-12.
Set the searchbase	The searchbase determines the entry point in the directory tree for a search.	"About the Searchbase" on page 4-23.

Table 1–1 (Cont.) Overview of Identity System Configuration

To perform this task	Description	Read
Configure view and modify permissions for attributes	You need to determine who can find what, at what point in the searchbase, and with what filter. These decisions affect who can read or write to data and who receives email notification when an attribute has been modified.	"Allowing Users to View and Change LDAP Data" on page 4-22.
Define workflows	A workflow is a series of steps for creating, deleting, and modifying attributes in the Identity System. For example, in the User Manager, you may want to define a workflow for creating a user that includes collecting information about the new user from several people in your organization.	"Chaining Identity Functions Into Workflows" on page 5-1.
Configure password policies	You can determine the length of passwords, frequency of password change, and so on.	"Configuring Password Policies" on page 7-49.
Delegate administration	To scale your installation, you need multiple administrators, each overseeing a subset of users.	"Specifying Identity System Administrators" on page 2-1.

About Managing the Identity System

You can extend your Identity System by adding servers, and expanding your network of Identity System administrators. You can configure audits and logs and perform other administrative functions. [Table 1–2](#) provides an overview of managing the Identity System:

Table 1–2 What to Read for More Information on the Identity System

To perform this task	Read
Add more Identity Servers	<i>Oracle Access Manager Installation Guide</i> . To ease this process, you may choose to add more Identity Servers through silent installation or cloning, as described in the <i>Oracle Access Manager Installation Guide</i> .
Add more WebPasses	<i>Oracle Access Manager Installation Guide</i> . To ease this process, you may choose to add more WebPasses through silent installation or cloning, as described in the installation manual.
Add other Identity System components	<i>Oracle Access Manager Installation Guide</i> describes how to install most components. Information on how to install the Access Manager SDK is located in the <i>Oracle Access Manager Developer Guide</i> .
Configure container limits for Organization Manager	"Setting Container Limits in Organization Manager" on page 4-53.

Introduction to Using Oracle Access Manager

Commonly used functions in the Oracle Access Manager user interface the following:

- Login
- Functional Areas on a Page

- Online Help
- Logout

Login

Oracle Access Manager logs people in based on the roles they have been assigned. As described in ["Specifying Identity System Administrators"](#) on page 2-1, you can specify the following roles for users:

- **End User:** An end user can perform searches, view profile data, and modify profile data, depending on access permissions set for individual attributes.
- **Delegated Access Administrator:** A Delegated Administrator is a user who can perform all of the same tasks as an end user and can create user, group and organization objects, depending on the level of permissions he or she has been granted. A Delegated Administrator can also view requests.
- **Delegated Identity Administrator:** A Delegated Identity Administrator is a user who has been delegated the right to view configuration tabs for the User Manager, Group Manager, and Organization Manager applications. This person can set attribute access controls, define workflows, and so on.
- **Identity Administrator:** An Identity Administrator can view the User Manager, Group Manager, and Organization Manager applications, and use Identity System configuration functions in the Identity System Console.

For example, if you log in as an Identity Administrator, you can view every screen in every application. But if you log in as an end user, you may only see a subset of the User, Group, and Organization Manager applications, and you cannot access Identity System administrative functions.

By default, single sign-on is configured between the Identity and Access Systems. If you log in to one system, you should not be prompted to log in to the other system.

If you use the Access System to protect the Identity System applications, you can bypass the default login form and implement your own custom form. For details about protecting resources with policy domains, see the *Oracle Access Manager Access System Administration Guide*.

Logging In to the Identity System

The procedure for logging in to the Identity System depends on whether you customized the login screen, made it available as a portal insert, or protected it with the Access System.

This section covers the default login screen that ships with the Identity System, as well as the impact of the default user type on login. See the *Oracle Access Manager Customization Guide* and the *Oracle Access Manager Developer Guide* for more information on customization.

You must configure an attribute with a semantic type of Login before users can log in to the Identity System. You can either automatically configure this attribute during installation, or manually configure it from the Identity System Console. See ["Making Schema Data Available to the Identity System"](#) on page 3-1 for more information.

Note: Only Master Identity Administrators and Delegated Identity Administrators have access to the Identity System Console. See ["Specifying Identity System Administrators"](#) on page 2-1 for more information about configuring these administrators.

To log in to the Identity System

1. In your browser, type the path to the Identity System and press Return.

For example:

`https://hostname:port/identity/oblix`

where *hostname* is the name of the computer on which the WebPass is installed and *port* is the Web server port for the WebPass. You can log in using the HTTP or HTTPS protocol.

The main product page appears. This page will have links to one or more applications, including the User Manager, Group Manager, and Org Manager.

See the *Oracle Access Manager Customization Guide* for more information about changing this default.

2. Select the desired application.

A login page appears.

3. Enter your user name and password.

For Active Directory users, if the Domain field is present, select the domain in which this installation of the Identity System operates.

By default, when you log in to the Identity System, you see all of the functions available to an Identity System Administrator. For example, in the User Manager, you will see functions such as "My Identity," "Reports," and the search function.

Logging into the Access System

By default, a user is not required to log in to the Access System if he or she is already logged in to the Identity System, and vice versa. Session information is stored in a cookie called the ObTEMC cookie. You may choose to protect the Identity System applications in a policy domain, in which case a different authentication can be used. For details about protecting resources with policy domains, see the *Oracle Access Manager Access System Administration Guide*.

You must configure an attribute with a semantic type of Login before users can log in to the Access System. You can either automatically configure this attribute during installation, or manually configure it from the Identity System Console. See "[Making Schema Data Available to the Identity System](#)" on page 3-1 for more information.

This section covers the default login screen that ships with the *Access System*.

Note : Only Master Administrators and Master Access Administrators have access to the Access System Console. For details about configuring Master Access Administrators, see *Oracle Access Manager Access System Administration Guide*.

To log in to the Access System

1. In your browser, type the path to Access System and press Return.

Example: `https://hostname:port/access/oblix`

where *hostname* is the name of the computer on which the Policy Manager is installed and *port* is the Web server port for the Policy Manager. You can log in with the HTTP or HTTPs protocol.

The main product page appears. This page will have links to one or more applications, including the Identity System, the Policy Manager, and the Access System Console.

2. Select the application you want.

Policy Manager—Only Delegated Access Administrators will see any policy domains. For details about delegating administration in the Policy Manager, see the *Oracle Access Manager Access System Administration Guide*.

Access System Console—Only Master Administrators and Master Access Administrators can access its functions. For more information about configuring Master Access Administrators, see the *Oracle Access Manager Access System Administration Guide*.

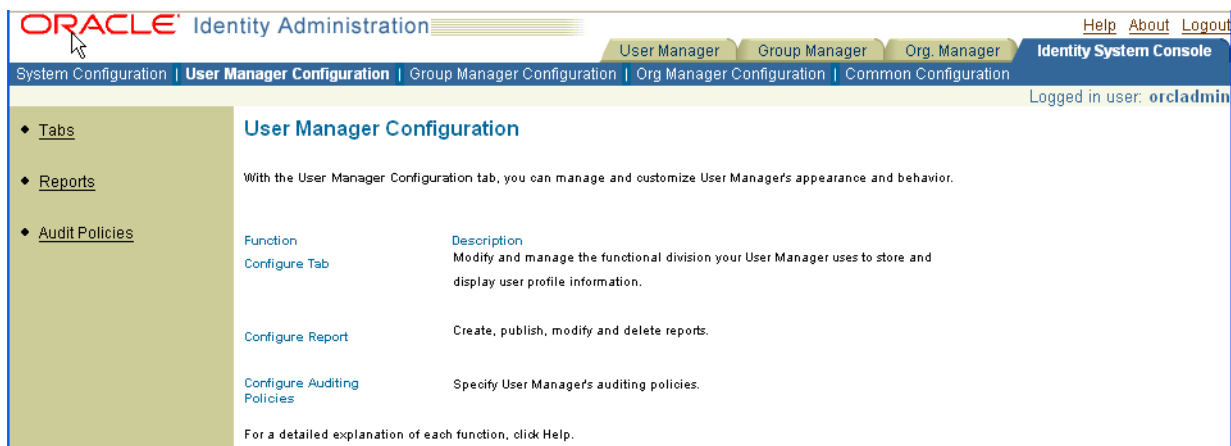
3. A login page appears.

Functional Areas on a page

The following identifies the main components of an Identity System page.

Navigation Elements

The following is a portion of an Identity System Console page. This page appears when you access the Identity System landing page, click the Identity System Console link on that page, then click the User Manager Configuration sub-tab.



All pages have the following functional areas:

- **Application Tabs:** A set of tabs that show the Identity System applications: the User Manager, Group Manager, Organization Manager (abbreviated to Org. in the user interface), and the Identity System Console.
- **Application Sub-Tabs:** A set of tabs that show the main functions for the Identity System applications. For example, the Identity System Console contains modules for System Configuration, User Manager Configuration, Group Manager Configuration, and Common Configuration.
- **Help, About, and Logout links:** These links appear at the top of the page.
- **Left Navigation Pane:** The Identity System Console uses a left navigation pane. This pane contains a list of links to functions that are applicable to the tab or sub-tab that has been selected. The user applications use sub-tabs and panels instead of a left navigation pane.

- **Main Body:** The main body displays a description of the currently selected function or the fields to be completed.

Search Functionality

The user interface contains search fields to search for users or groups. These search fields appear on most Identity application pages. The number of fields available to you and the items that you can search on depend on how an administrator has configured the search function.

You query for users or groups by filling in the search criteria and clicking Go. Optionally, you can store the results of a query by clicking the Reports tab and selecting Generate Report. Oracle Access Manager uses a locale-based case insensitive sorting method when you click the column heading (Full Name, for example) in the search results table.

ORACLE Identity Administration

User Manager Group Manager Org. Manager Identity System Console

My Profile | Reports | Create User Identity | Deactivate User Identity | Substitute Rights | Requests | Configuration

Search Full Name That Contains All 8 Results Go Advanced

To use the search function

1. Enter your search criteria.

For the simplest kind of search, type a text string in the Search entry field.

By default, you need not enter a minimum number of characters. However, to help users narrow their search criteria you can control the minimum number of characters that users must enter in the search field by setting the `searchStringMinimumLength` parameter in `oblixadminparams.xml`. See the *Oracle Access Manager Customization Guide* for details.

2. Click Go.

Users or groups matching your search criteria appear on the screen.

By default, 8 results are displayed on a page. This applies to both Selector and Query Builder. If you perform a search or query that results in more than 20 hits, you receive truncated results. For instructions on changing this search cap, refer to the `cookieBustLimit` parameter documentation in the *Oracle Access Manager Customization Guide*.

ORACLE Identity Administration

User Manager Group Manager Org. Manager Identity System Console

My Profile | Reports | Create User Identity | Deactivate User Identity | Substitute Rights | Requests | Configuration

Search Full Name >= All 8 Results Go Advanced

Table View Custom View

Search Results

Previous Next Email Link Customize

Full Name

Silvia Matson

Susan Wacheski

3. In the search results, click the links for the users or groups to select them.

Note: If you receive a "Bad request" message when you click Done, your search string is too long for your browser. Browsers handle the search parameters as URLs, and they generate an error if the search exceeds their maximum URL length.

4. In the search results, click the column heading to sort the list.

Note: Oracle Access Manager uses a locale-based case insensitive sorting method.

The Selector

The selector provides search functionality and the ability to aggregate the results of a search. For example, if you want to create a group, after clicking the Group Manager tab and the Create Group sub-tab, a page appears with selection buttons.

If you click a selection button (Select Member in this example), the Selector appears.

The Selector landing page is a blank search page with Done and Cancel buttons and an empty list of selected items. If you use the search functionality on this page, the Selector enables you to move retrieved items to the Selected list.

Search: Full Name >= sil All 8 Results Go Advanced

Done Cancel

Table View Custom View

Search Results

Previous Next

Add All

Full Name
ADD Silvia Matson
ADD Susan Wacheski
ADD Skipper McQuaig

Selected

Name

No Selections Currently

Online Help

A Help link is located at the top right of Identity System screens, and in the side navigation bar of Access System screens. To access online Help, click this link.

You can perform the following tasks in an online Help window:

- Scroll to view the entire Help topic.
- Click Contents to display a list of topics.
- Click Back or Forward to see other Help topics.
- Click Exit to close the window.

The About Page Link

A link to the About page is located at the top right of Identity and Access System pages. Click the About link to display the Oracle address, telephone numbers, and other contact information, and copyright information.

The **View System Info** button displays the server platform and version, and contact information for Oracle.

Logging Out

A Logout link is located at the top of the Identity and Access System pages. By default, if you log out of the Identity System, you are automatically logged out of the Access System and vice versa.

When you finish using Oracle Access Manager, to prevent unauthorized people from accessing your information you should log out and close your browser.

By default, sessions expire after three hours. To change the timeout, see "[Configuring Session Timeout](#)" on page 7-10 for details.

Note: On Firefox, users are prompted to manually close their browser window after logging out.

To log out

1. Click Logout in the upper right-hand corner of the page.
2. Click OK when prompted to close your browser.

Specifying Identity System Administrators

This chapter explains how to specify Identity System administrators.

This chapter contains the following topics:

- [About Identity System Administrators](#)
- [Specifying Administrators](#)
- [Delegating Administration](#)

About Identity System Administrators

The Identity System manages user, group, and organization identity information, as described in the *Oracle Access Manager Introduction*.

Administering the Identity System involves a broad range of tasks that are designed to help you manage your data, enhance performance, and control the appearance and functionality of Identity System applications. For details about these tasks, see "[Configuring Global Settings](#)" on page 7-1.

The responsibility of administering the Identity System is shared between Master Administrators and Master Identity Administrators:

Master Administrators: At least one Master Administrator is specified when the product is set up. This is the highest level administrator. This administrator can specify other Master Administrators and Master Identity Administrators.

Master Identity Administrators: Master Identity Administrators can delegate specific responsibilities to administrators called Delegated Identity Administrators.

Delegated Identity Administrators: Delegated Identity Administrators are assigned by Master Identity Administrators and created in User Manager.

See [Table 2-1](#) for a description of the types of Identity System administrators and their privileges

Table 2–1 Types of Identity System Administrators

Administrator	Becomes an Administrator When	Tasks Performed
Master Administrator	Assigned when Oracle Access Manager is installed	■ Assigns other Master Administrators and Master Identity Administrators ■ Assigns self as a Master Identity Administrator ■ Manages all System Configuration and System Management functions of the Identity System Console ■ Configures Identity Server ■ Specifies administrators ■ Configures styles ■ Configures directory server profiles ■ Configures WebPass ■ Configures Password policies ■ Manages Identity Server settings ■ Imports photos ■ Manages log files and audit files

Table 2–1 (Cont.) Types of Identity System Administrators

Administrator	Becomes an Administrator When	Tasks Performed
Master Identity Administrator	Assigned by the Master Administrator	■ Assigns Delegated Identity Administrators ■ Manages all three Identity System applications: User Manager, Group Manager, and Organization Manager ■ Manages Common Configuration as well as application-specific configurations in the Identity System Console ■ Common Configuration Tasks: - Configures object classes - Configures workflow panels - Configures master audit policies - Configures logging and auditing policies ■ User Manager Configuration Tasks: - Configures tabs - Configures reports - Configures logging and auditing policies ■ Group Manager Configuration Tasks: - Configures tabs - Configures reports - Configures group types - Configures Group Manager options - Configures logging and auditing policies - Manages the group cache ■ Organization Manager Configuration Tasks: - Configures tabs - Configures reports - Configures logging and auditing policies
Delegated Identity Administrator	Assigned by Master Identity Administrators	■ Assigns other Delegated Identity Administrators ■ Manages assigned tasks ■ Delegates administration ■ Configures attribute access control ■ Defines workflows ■ Monitors workflow status ■ Sets searchbase ■ Expands dynamic groups ■ Sets container limits

Specifying Administrators

You use the Identity System Console to assign Delegated Identity Administrators and Master Identity Administrators. As mentioned earlier, you must be a Master Administrator to complete this task.

To specify Master Administrators and Master Identity Administrators

1. Log in to the Identity System as a Master administrator, and from the landing page for the Identity System, click the Identity System Console link.

If you are already logged in, click the Identity System Console tab.

2. Click the System Configuration sub-tab.

The System Configuration page appears.

3. Click Administrators in the left navigation pane.

The Configure Administrators page appears, displaying two options: Master Identity Administrators and Master Administrators.

See [Table 2-1](#) for a list of the tasks performed by each type of administrator.

4. Click the category of administrator you want to add.

A *Modify type_of_administrator* page appears.

where, *type_of_administrator* is either a Master Administrator or a Master Identity Administrator.

5. Click Select User to add an administrator.

See "[The Selector](#)" on page 1-10 for information about using this feature.

6. Select a user and click Add.

The name you select in the Selector page appears in the *Modify type of administrator* page, where *type of administrator* is a Master Administrator or a Master Identity Administrator. You can specify multiple administrators.

7. Click Done to leave the Selector page.

8. Click Save to add the administrator.

Deleting Administrators

When you delete an administrator, you remove administration rights from the user, but you do not remove or deactivate the user from the LDAP directory.

To delete an administrator

1. From the Identity System Console, click the System Configuration sub-tab.
2. Click Administrators.
3. In the Configure Administrators page, click the link for the type of administrator that you want to delete.

The *Modify type of administrator* page appears, where *type of administrator* is a Master Administrator or a Master Identity Administrator.

Click Select User.

4. Clear the DEL button next to the administrator who you want to delete.
5. Click Done to confirm the deletion.

Delegating Administration

You can delegate your rights and responsibilities to other administrators. The tasks delegated are specific to the delegated right, the target, and the tree path.

This section covers:

- [About Delegating Administration](#)
- [Delegated Administration Models](#)
- [Adding Delegated Administrators](#)

About Delegating Administration

Delegating administration allows the Master Administrator and Master Identity Administrator to delegate their responsibilities to other, more local administrators. This is particularly useful in large organizations, where it may be necessary to administer thousands or millions of users.

When you delegate administration, you determine what rights you want to grant to another user. Rights include the ability to configure the following:

- Read access for attributes
- Write access for attributes
- Notification by email of attribute modifications
- Setting the searchbase
- Monitoring requests
- Defining workflows
- Containment limits

In addition, you can designate people to act as your substitute. People who are granted substitution rights can temporarily perform any of the functions that you are permitted to perform.

After you have delegated a right to another user, that user becomes a Delegated Identity Administrator. By delegating administration, you determine who can configure or access which feature, at what level, and with which filters. Configuration or access authority may be for a specific user or group of users, a role, or a rule. The resource that can be configured or accessed may include a searchbase, an attribute access control, a workflow definition and so forth. The level is the starting DN.

Task overview: Delegating administrators

1. Start the delegation procedure for the desired application.

Note: All activities here are described in "[Adding Delegated Administrators](#)" on page 2-8.

2. Select the right that you want to grant (for Read, Write, and Notify permissions only).
3. Identify the attribute associated with the right.
4. Specify the level of access control for that attribute, thus setting the scope of the directory tree to which the rights apply.

5. Select the person to whom you are delegating the rights.

For example, as the Master Identity Administrator, you can grant one or more users the ability to set Read access control for the Title attribute. You can specify whether you want the Delegated Identity Administrator to be able to delegate access control to others.

For more information, see "[Adding Delegated Administrators](#)" on page 2-8.

Delegated Administration Models

The Identity System enables you to set access controls and delegate administration for directory tree structures that represent different business models. These models include an extranet model, an intranet model, and an ASP model. These models are described in the following sections:

- [Extranet Model](#)
- [Intranet Model](#)
- [ASP Model](#)

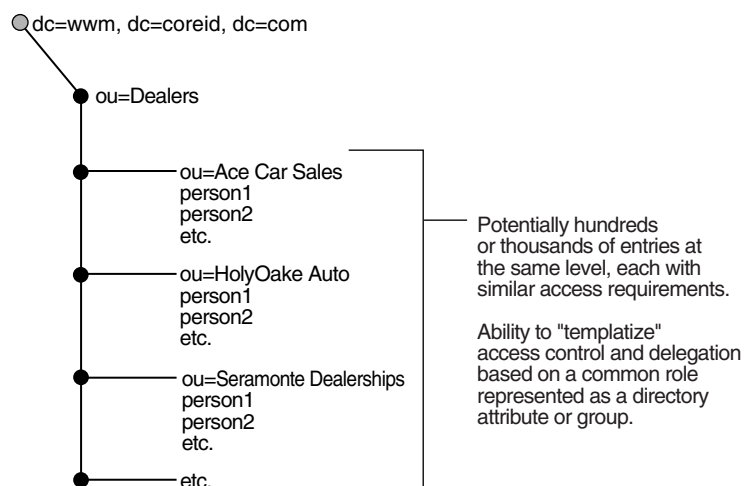
Extranet Model

A typical business-to-business extranet might have 500 or more extranet organizations using a site. These organizations represent customers, partners, and suppliers, each having between 1 and 100 users.

The goal in the extranet model is to have the Master Identity Administrators push out administrative responsibilities to each of the partners. But because there are so many partners, it would be a burden to define new roles and responsibilities each time a partner joined. Therefore, the Directory Administrator must define a fixed set of roles and responsibilities that are leveraged across all customers, existing and new. The Master Identity Administrator can then define access controls and create delegated administrator policies that are symmetric across all organizations.

The Delegated Identity Administrator at each partner site is typically a line-of-business person who has a fixed, well-defined set of tasks and rights, such as creating users and changing attribute access permissions. Delegated Identity Administrators can only give others in their organization administrative privileges by adding and deleting people from a set of pre-defined roles.

For example, the Delegated Identity Administrator creates a new user with an attribute of admin=yes. This new user then inherits the ability to change attribute access control permissions, create new users, and other well-defined tasks, as illustrated in [Figure 2-1](#).

Figure 2–1 Extranet Delegated Administration Example

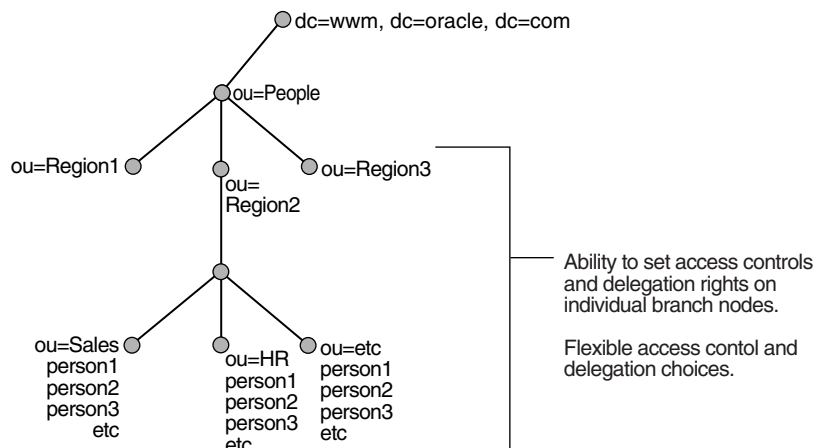
Intranet Model

In a typical intranet model, the directory tree is generally organized according to a logical separation of users, such as by geography (North America and Europe) or function (Marketing and Engineering).

The directory might be characterized by only a few branches at each OU, but may be several layers deep in branching. The branches may be very different from each other and may have several thousand users in each branch. At a given node, a European branch might have 500 users under Sales and Marketing, while a North American branch might have 10,000 users under East, Central, and West.

The Master Identity Administrator may choose to delegate administration centrally or at the OU level, depending on where the technical and business process knowledge resides. Or additionally, the Master Identity Administrator may choose to delegate administration across specific tasks; for example, you might delegate the task of provisioning phone numbers—but not managing access permissions or creating new users.

Figure 2–2 illustrates the intranet model:

Figure 2–2 Intranet Delegated Administration Mode

ASP Model

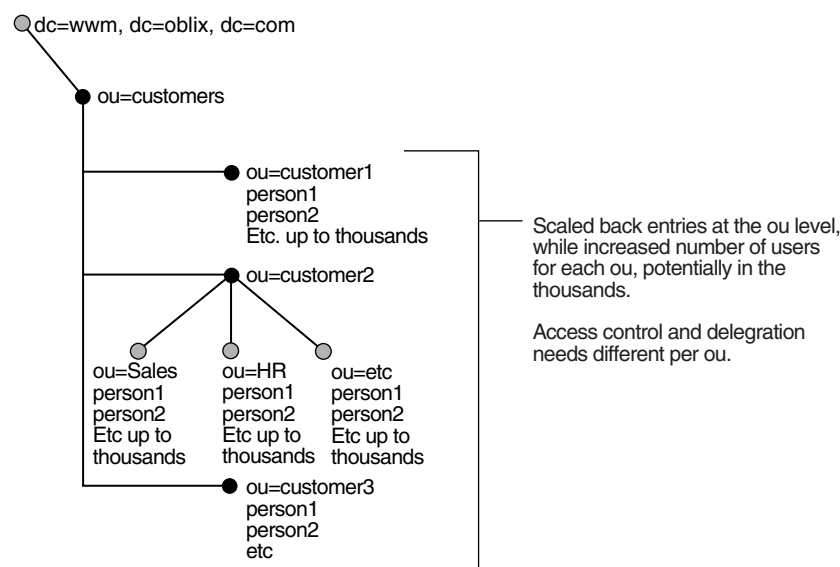
Some business-to-business extranet sites may follow the application service provider (ASP) model more closely than the extranet model described earlier.

In an ASP model, there are fewer extranet partners but significantly more users at each partner site. For example, there may be only approximately 50 partners but each partner may have 1000 users.

ASPs provide hosted services. Different customers may need different sets of services. This means the scope of data that needs to be managed, such as access rights, may differ for each OU. Further, the directory structure of each OU may be substantially different. Under each OU may be all the complexity of an intranet directory tree as in the intranet model, yet the structure of the tree could be completely different between the OU for Customer 1 and the OU for Customer 2.

The ASP model needs a flexible delegation model similar to the intranet model. The Master Identity Administrators at the ASP site performs some top-level configuration, such as setting the searchbase, and configures an initial delegation model similar to the extranet model. However, each customer site requires the flexibility to create a customized delegated administration model, either by a technical Delegated Identity Administrator at the customer site or by the Master Identity Administrators at the ASP site.

Figure 2-3 ASP Delegated Administration Model



Adding Delegated Administrators

Delegating administration allows the Master Identity Administrator or a Delegated Identity Administrator to further delegate responsibility to other local administrators.

To delegate administration

1. Log in to the Identity System, and from the landing page select the link for the User Manager, Group Manager, or Organization Manager.

If you are already logged in, select the tab for the application.

2. Click the Configuration sub-tab.

The Configuration page appears.

3. Click the Delegated Administration link.

On some browsers you may receive a prompt asking if you trust the certificate of the application. If you receive this prompt, select the Trust Always option.

The Delegate Administration page appears.

4. In the Management Domain box, specify the scope of the DIT that this right applies to.

Initially this field displays the searchbase defined during setup. The searchbase is usually defined at the highest (company-wide) level. Depending on the level of your delegated rights, you can specify access control at any level, from the lowest level (an individual user), through intermediate levels (departments, divisions, partners), and then to the highest level (company-wide). For example, if you select the Full Name attribute and select a department such as Sales, you are setting an access control that applies to all full names belonging to the Sales department.

The selection appears in the field beneath the Management Domain box.

5. Optionally, use the Filters field to specify either a variable substitution or LDAP rule to filter the DIT level you selected.

For more information, see ["Usage of Rules and Filters"](#) on page 3-23.

6. Optionally, in the Add Filter field, enter another filter, then click Save.

The new filter appears in a field beneath the previous filter.

7. In the Grant Right list, select the right that you want to grant to the delegated administrator:

- **Read:** Allowed to set read (view) permission for the selected attribute
- **Modify:** Allowed to set modify permission for this attribute
- **Notify:** Allowed to set notify permission when user requests attribute value change
- **Set Searchbase:** Allowed to specify the searchbase
- **Monitor Requests:** Allowed to monitor requests and manage deactivated users
- **Define Workflow:** Allowed to define workflows
- **Substitute Rights:** Allowed to designate other people as your substitute

8. Give the new administrator the authority to further delegate this right to other administrators by selecting the Delegate Right check box.

Note: Selecting Delegate does not automatically assign Grant rights. You must define Delegate and Grant rights separately.

9. In the Attribute box, select an attribute to associate with the right.

10. Select a trustee to whom you want to assign one or more rights with one or more of the following methods:

- **Rule:** Click Build Filter and use the Query Builder to create a rule. See ["Writing LDAP Filters Using Query Builder"](#) on page 4-29 for details.
- **Person(s):** Click Select User and use the Selector to specify one or more users.

- **Group(s):** Click Select Group and use the Selector to specify one or more groups.
The Rule, Person(s), and Group(s) fields have an or relationship. A user specified in any of these fields is assigned the right.
11. Use the Copy and Paste buttons to copy users and groups from one attribute to another.
Click Copy, click Reset, select another attribute, and click Paste. The users and groups appear in their respective boxes.
 12. Click one of these buttons:
 - **Save:** Saves and implements your changes
 - **Reset:** Clears all selections
 - **Delete:** Clears all rule, group, and user specifications
 - **Report:** Generates a report of all attributes and their access permissions across the domain

Adding Substitute Administrators

As an Identity System Administrator, if you have been granted substitute rights, you can designate other people to temporarily assume your rights. After your substitute logs into the Identity System, they can assume your identity. When your substitute views the My Identity page, your information is shown rather than the other person's information.

By assigning substitute rights, you allow someone else temporarily to assume your identity. For example, suppose you are a Delegated Identity Administrator. Before leaving for vacation you assign substitute rights to J. Smith. When J. Smith logs in, he assumes your identity. Later, when J. Smith wants to perform his own duties, he reverts the delegated rights. Although the substitute appears to be you while assuming your identity, the Identity System logs all activities with both the substitute's and your identities. All logs and alarms show duplicate entries using both identities.

To assign or remove a substitute

1. From the Identity System landing page, log in and select the link for the User Manager.

If you are already logged in, click the User Manager tab.

2. Click the Substitute Rights link.

The Substitution Rights page appears.

If you have been granted substitute rights, this page contains a Select User button. If you have designated people to be your substitute, these people are listed in the Substitute(s) field. This page also contains a Substitute for field with a list of people who have designated you as their substitute. If no people are listed, no one has designated you as their substitute.

3. Assuming that you have been granted substitute rights, click Select User.

The Selector page appears. See "[The Selector](#)" on page 1-10 for details.

4. Select the user and click Add.

The user is added to the Selected list.

5. Select a user and click Delete to remove the user.

The user is removed from the Selected list.

6. Click Done to leave the Selector page.
7. Click Save to save your changes.

To assume an identity

1. From the Identity System landing page, log in and select the link for the User Manager.

If you are already logged in, click the User Manager tab.

2. Click Substitute Rights.
3. In the Substitute for User section of this page, select the user whose rights you wish to assume.

This user must already have assigned you to be a substitute.

4. Select Assume Right and click Save.

To revert to your own identity

1. From the User Manager, select Substitute Rights.
2. Select Revert

See "[Configuring Global Settings](#)" on page 7-1 for details about configuring styles for Identity System applications, configuring multiple languages for the Identity System, configuring and managing Identity Servers and WebPass, and configuring password policies and the Access Manager SDK for the Identity System.

Part II

Configuring the Identity System

You use the Oracle Access Manager Identity System to manage user data, configure Identity applications (User Manager, Group Manager, and Organization Manager), define workflows, and send non-LDAP data to external applications.

Part II explains how to configure the Identity System and contains the following chapters:

- [Chapter 3, "Making Schema Data Available to the Identity System"](#)
- [Chapter 4, "Configuring User, Group, and Organization Manager"](#)
- [Chapter 5, "Chaining Identity Functions Into Workflows"](#)
- [Chapter 6, "Sending Non-LDAP Data to External Applications"](#)
- [Chapter 7, "Configuring Global Settings"](#)

Making Schema Data Available to the Identity System

The Identity System applications (the User, Group, and Organization Manager) enable users to view and modify data about themselves, other users, groups, and other objects. The items that users see on the Identity System applications consist of LDAP directory attributes that you have configured in the Identity System Console. In order for the Identity System applications to display data, you use the Identity System Console to configure objects and attributes from a directory schema that the application is to work with.

The Identity System applications also enable users to send data to back-end applications. For example, users can enter data in a workflow, and that data can be sent to an application that creates new user email accounts. To prepare the Identity System applications to be used in this manner, you use the Identity System Console and configure objects and attributes from a template schema. A generic schema file is supplied with the Identity System.

This chapter discusses the following topics:

- [About Object Classes](#)
- [Viewing Object Classes](#)
- [Modifying Object Classes](#)
- [Adding Object Classes](#)
- [Deleting Object Classes](#)
- [About Object Class Attributes](#)
- [Viewing Attributes](#)
- [Configuring Attributes](#)
- [Configuring Derived Attributes: Matching Values from Different Attributes](#)
- [Attributes Configured for an Individual Application](#)

About Object Classes

As an Identity System administrator, you configure three applications for your users (and other administrators). These applications are the User Manager, Group Manager, and Organization Manager.

[Figure 3-1](#) shows a portion of a User Manager Profile page.

Figure 3–1 Sample User Manager Profile Page

The screenshot shows the Oracle Identity Administration console. The top navigation bar includes 'ORACLE Identity Administration' and tabs for 'User Manager', 'Group Manager', and 'Org. Manager'. Below the navigation bar is a search section with a dropdown menu set to 'Full Name', a search box, and a 'That Contains' dropdown. To the right of the search box are radio buttons for 'All' and '8' (selected), a 'Results' label, and 'Go' and 'Advanced' buttons. A 'Logged in' link is also present. Below the search section are 'View Panels' and 'Modify' buttons. The main content area is titled 'User Profile' and displays the following information:

Title	Director
Full Name	Silvia Matson
Car License	WKQTW2H
Department Number	20P556

The Identity System applications obtain most of the data that they display from entries in an LDAP directory. For instance, the User Manager may show a person's name, email, and so on. This data is taken from attribute values stored on a person object in the directory. These attributes and their values are displayed on profile pages in the User Manager. In [Figure 3–1](#), the name displayed in the user profile is based on the name attribute for the person object in the directory. The actual name being displayed is a value stored with the attribute. The title is based on the title attribute for the person object.

All of the Identity System applications—the User Manager, Group Manager, and Organization Manager—display attribute values for specific objects on profile pages.

About Sending Data to External Systems Using Template Objects

In addition to configuring objects and attributes from an LDAP directory, the Identity System enables you to define template schemas. Using the Identity System Console, you configure the objects and attributes from a template schema in the same way that you configure LDAP data. However, LDAP data and template data are used in different ways. You configure LDAP data to populate Identity System applications. Users enter values for LDAP data from either an Identity System application profile page or from a workflow. The LDAP data is displayed on the profile page. In contrast, you can only enter template data during a workflow step. The data is not displayed on the profile page. Instead, it is sent to a back-end application that needs the data.

Unlike LDAP data, you can use only template object data for sending data to back-end systems. For example, you can create an object template schema with attributes that an email system can understand. You would then configure the attributes from this schema in the Identity System, define a workflow that uses these attributes, and use the Identity Event API to send this data to the back-end system.

Note: For instructions on how to configure the template object file, see ["Sending Non-LDAP Data to External Applications"](#) on page 6-1. The template object file must be finalized before you can configure the template objects in the Identity System Console, as described in ["Sending Non-LDAP Data to External Applications"](#) on page 6-1.

The Process for Configuring Schema Data

When you first install and set up Oracle Access Manager, the User Manager, Group Manager, and Organization Manager applications are empty. You need to configure these applications with information. For example, you may want the User Manager to

display information about a user such as their name, title, phone number, email, and so on. Before configuring the appearance of these applications, you must first use the Identity System Console to configure the LDAP objects and attributes that you want to display on the application profile page. You must also define how each attribute is to be displayed, for example, if it is a string value, a selection list, or a radio button. You primarily use LDAP directory data to identify in the Identity System Console which objects and attributes you want to display to users on application profile pages.

Once you have configured objects and attributes in the Identity System Console, you can configure Identity System applications to display these attributes and their values. Configuring the Identity System applications is discussed in ["Configuring User, Group, and Organization Manager"](#) on page 4-1. Finally, you assign View and Modify rights to determine who can view and modify these attributes.

If you are using the Identity System as the entry mechanism for an external application, you use the Identity System Console to configure template objects and attributes. As with LDAP data, you define how each attribute is to be displayed on a profile page. The primary difference between LDAP and template data is that users may only enter values for template attributes during a workflow step, and you must use the Identity Event API to pass this data to a back-end application.

Objects Configured During Installation

During installation and setup of the Identity System, you configured the following object classes:

- **User Manager:** A required person object class
- **Group Manager:** A required group object class
- **Organization Manager:** A predefined location object class

These object classes are taken from the LDAP directory that the Identity System communicates with.

In addition to the object classes that are configured during the installation process, you may want to configure additional LDAP and template-based objects and attributes. The following sections discuss how you configure objects and attributes to provide the Identity System applications with data.

Note: Configuring an attribute does not ensure that the attribute is displayed on an Identity System application page. You must associate specific attributes with an Identity System application page, and assign View and Modify rights to these attributes. See ["Configuring User, Group, and Organization Manager"](#) on page 4-1 for details.

Structural and Auxiliary Object Classes in the Identity System

The Identity System works with structural and auxiliary LDAP object classes. When you install the Identity System, the User, Group, and Organization Manager applications are associated with one structural object class each. A structural object class describes the basic aspects of an object. Examples of structural object classes include person and groupOfNames. The person object class may contain attributes such as name, department, employee ID, and email address.

The User Manager and Group Manager are always associated with only one structural object class.

The Organization Manager can be associated with any number of structural object classes of a generic or location object class type (see ["Object Class Types"](#) on page 3-5 for details). During installation and setup, a location structural object class is associated with the Organization Manager. In the Organization Manager, each structural object class is represented as an option in the menu in the top right corner of the page. The work area for a particular object class in the Organization Manager is referred to as a tab.

Figure 3–2 Organization Manager Tabs



You use an auxiliary object class to add a set of related attributes to an entry that already belongs to a structural class. Auxiliary object classes are mix-in LDAP object classes that can be added to any structural class. Items such as a billing address, a challenge phrase, a response to a challenge phrase, and so on may be useful for definition in an auxiliary object class.

You must configure attributes for each object class that you want to manage through the Identity System Console. See ["About Object Class Attributes"](#) on page 3-10 for more details.

The inheritance of all objects is based on the premise of a common super class for both the structural object class and the auxiliary class. Otherwise, object class extension is not feasible. For example, if you don't choose Top as the inherited Object Class in eDirectory, NDS sets the inherited object class to None. When you configure the object class as an auxiliary class in the Identity System, no problems are evident initially. However, if you execute a Create User Workflow that contains attributes of this auxiliary class, the Enable step fails when trying to commit because the schema is incompatible and the auxiliary class cannot be added to the entry's object class attribute due to a schema violation.

Note: Before users can see values for attributes that you configure, you must set up the User Manager, Group Manager, and Organization Manager tabs and provide view and modify permissions, as appropriate, to the users. See ["Configuring User, Group, and Organization Manager"](#) on page 4-1 for details.

Template Object Classes

In addition to structural and auxiliary object classes, the Identity System recognizes template object classes. Template object classes function in part like auxiliary object classes, in that they are used to augment the functionality on an Identity System application tab, and they cannot be used as the foundation for a tab. However, template objects are not defined in an LDAP directory, and you do not use template objects for configuring the profile pages shown on a tab.

You define template objects in a schema file. The template objects are only used in Identity System workflows, for sending data to back-end applications. Template object classes differ from the other kinds of object class in several respects:

- Users interact with the template object class attributes to send data to back-end applications.
- Users interact with template object class data attributes only in the context of an Identity System workflow.

Template attribute values are only visible when a user invokes a workflow instance and enters the data. Once the data is submitted, it cannot be displayed again in the Identity System. See ["Chaining Identity Functions Into Workflows"](#) on page 5-1 for details. This is different from LDAP data, which you use to configure the fields, labels, and other items displayed on the Identity System application pages.

Note: Template attribute values cannot currently be displayed because the flow of data from the Identity System to the back-end system is one-way. This limitation will be removed in a future release.

- Template object data resides a template object file, not an LDAP directory.

For information on defining template objects and the complete process for using them with back-end applications, see ["Sending Non-LDAP Data to External Applications"](#) on page 6-1.

Object Class Types

When configuring your object classes, you are asked to specify an object class type. The term object class type refers to how an object class is used within the Identity System. [Table 3–1](#) lists the object class types supported in the Identity System.

Table 3–1 Object Class Types

Type	Description
Person	This type contains information about a person. Examples of this type include companyOrgPerson and customerOrgPerson. When you install the Identity System, the oblixOrgPerson type is created. This is an important auxiliary object class. It provides the obUserAccountControl attribute, which you should never modify. This attribute is written to the profile of any user you deactivate.
Group	This type contains information about a group. Examples include groupOfUniqueNames and mailGroups. When you install the Identity System, the oblixGroup and the oblixadvancedgroup type auxiliary classes are created to help you configure useful information on the Group manager.
Location	This type contains information about locations. The Organization Manager uses this object class to store and display location information.
Generic	Any object class that does not fit in the other categories. An example includes the organizationalUnit object class that is managed by the Organization Manager.

Viewing Object Classes

When you install and set up the Identity System, several object classes are already configured. You can view and modify these object classes, and you can create additional object classes.

Note: By default, the Identity System does not display every object and attribute in the directory. This parameter `excludeOCsForTreeInApplet` in `Identity_install_dir/apps/common/bin/globalparams.xml` enables you to expose object classes in the Identity System applications that would otherwise be hidden.

See the parameter reference chapter in *Oracle Access Manager Customization Guide* for details.

To view configured object classes

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

The Common Configuration page appears.

2. Click the Object Classes link in the left navigation pane.

The Configure Object Classes page displays object classes that are configured in your LDAP directory and object templates, along with the following information:

Column	Description
Object Class	Name of the object class.
Object Class Type	How the object class is used by the Identity System. See " Object Class Types " on page 3-5 for details.
Object Class Kind	If you have configured an LDAP object, the kind can be Structural, Auxiliary, or other object. See "Structural and Auxiliary Object Classes in the Identity System" on page 3-3 for details. An object class kind of Other indicates that the object class kind is undefined. Text can be Don't care or Unknown. If you have configured a template object, the kind can only be Template. See "Template Object Classes" on page 3-4 for details.
Object Class Attribute	This is used by the Identity System for attribute access and it is also the attribute that the Identity System uses to link search results to a profile page. See " Selecting a Class Attribute " on page 3-7 for a description.

Modifying Object Classes

From the Identity System Console, you can change the class attribute and the type for an object class. It is important to specify a class attribute for your structural object classes.

You can change the structural object class. However, it is best if you plan your configuration so that this task is not necessary.

Note: Using the application-specific Tabs function, you can provide a different display name or display type for an attribute on that application-specific Configuration tab only (different than what is configured at the object class level). This will override the information configured for the attribute at the object class level. For details, see "[Modifying and Localizing Attributes Displayed on a Panel](#)" on page 4-21.

To modify an object class type

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

The Common Configuration page appears.

2. Click the Object Classes link in the left navigation pane.
3. Click the link for the object class you want to modify.

The View Object Class page appears.

4. Click Modify.

The Modify Object Class page appears.

5. Select a new class type.

See ["Object Class Types"](#) on page 3-5 for more information on object class types.

6. Save your change.

Selecting a Class Attribute

In the User Manager, Group Manager, and Organization Manager, each tab is associated with a structural object class. Within the structural object class, you select an attribute to be the class attribute. The class attribute is used for attribute access. Users who do not have read access to a class attribute do not have access to the entire entry.

Note: It is not required to set a class attribute for a template object class. You determine user access to template objects and attributes when you configure a workflow, as described in ["Chaining Identity Functions Into Workflows"](#) on page 5-1.

The Identity System also uses the class attribute when displaying search results on a profile page. When a user conducts a search, the Identity System returns a list of results. Each returned item has one value that is displayed as a link. The linked value is taken from the class attribute of the returned object. When the user clicks the link, the Identity System displays the profile associated with that link.

For example, if you specify User Name as the class attribute for the orgPerson object class, when someone searches in the User Manager, the list of search results displays user names as links. Clicking a link displays the profile for that user.

Class attributes are usually selected as follows:

- User Manager uses a class attribute for a person's name.
- Group Manager uses a class attribute for a group's name.
- Organization Manager uses one class attribute for each tab. For the location structural object class, the attribute would typically be a location name.

To select the class attribute

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

The Common Configuration page appears.

2. Click the Object Classes link in the left navigation pane.
3. Click the link for the object class you want to modify.

The View Object Class page appears.

4. Click Modify.

The Modify Object Class page appears.

5. Select the class attribute from the list of attributes.
Note that you can only select a class attribute for a structural object class.
6. Click Save.

Changing the Structural Object Class

Changing the user or group structural object class requires you to rerun Identity System setup.

To change user or group structural object classes

1. Shut down all but one Identity Server.
2. Locate *IdentityServer_install_dir*/identity/oblix/config/setup.xml, and change the status parameter value from "done" to "incomplete," as described in ["Rerunning Setup Manually"](#) on page 7-29.
3. Depending on which application you are modifying, delete the top node for the structural object class, as follows:

User Manager node: obapp=userservcenter,o=Oblix,o=company,c=us

Group Manager node: obapp=groupservcenter,o=Oblix,o=company,c=us
4. Restart the Identity Server and navigate to the Identity System Administration Console to initiate and complete the setup process as you reconfigure the structural object classes.

When you restart the Identity Server, the other Identity Servers should pick up the new structural user or group object class from the updated directory tree.

Adding Object Classes

There are two basic methods for adding an object class in the Identity System Console:

- Configure each attribute manually.
- Select the autoconfigure object class option

This method configures the object class using settings from the Identity System. This option is faster than manual configuration. You cannot view or modify the Identity System-provided attributes before importing them.

With either option, you can later modify the attributes from the System Console. See ["About Object Class Attributes"](#) on page 3-10.

To add an object class

1. From Identity System Console, click Common Configuration, Object Classes.
2. Click Add.

The Add Object Class page appears.

The default schema domain is LDAP. If you have not defined any template object classes, LDAP is the only choice. If you have configured additional template object classes and want to configure objects from the additional class, select the class from the Schema Domain list.

3. In the Schema Domain list, select the type of schema that you want to work with, if applicable.
4. From the Object Class list, select the object class to add.

This allows the Identity System to manage the object class. The list contains object classes that were defined in your LDAP directory prior to installing the Identity System.

5. In the Class Type field, select what type of Identity System application can manage this object class.

See ["Object Class Types"](#) on page 3-5 for details.

6. In the Class Kind field, select Structural, Auxiliary, Template, or Other.

If the Identity System can determine the Class Kind from the LDAP directory, these radio buttons are hidden.

7. Select Auto Configure object class to populate this object class with attributes from an Identity System-provided file.

8. Click Save.

When a template object class is saved, it is saved in fully qualified form. For example:

```
obclass=person.,o=oblix,o=company,c=us
```

This format is taken from the .tpl file that contains the template object class definition. See ["Sending Non-LDAP Data to External Applications"](#) on page 6-1 for details.

How Auxiliary Classes Are Used

You can use auxiliary object classes as mix-ins with structural object classes. This can be helpful when you configure the User Manager, Group Manager, and Organization Manager applications. The more object classes you have at your disposal, the more items you can display on the tabs for these applications, and the more information you can configure for users of those applications.

An object assigned to an auxiliary object class must be associated with a structural class. For example, you can add inetOrgPerson as your structural object class and associate it with the tab in the User Manager application. You can then add auxiliary object classes with attributes for particular kinds of people, such as customers, partners, and so on.

To associate one or more auxiliary object classes with the structural object classes chosen for the Identity System applications, see ["Adding Auxiliary and Template Object Classes to a User or Org. Manager Tab"](#) on page 4-8.

Deleting Object Classes

You can delete auxiliary object classes. You also can delete template object classes that have not yet been added to a tab for a user or group. You cannot delete a structural object class. You can only substitute a new structural object class for an existing one. See ["Changing the Structural Object Class"](#) on page 3-8 for details. When you delete an object class, you should also remove any searchbases that you have configured for that object class. See ["Setting the Searchbase"](#) on page 4-25 for details.

To delete an auxiliary object class

1. From the Identity System Console, select Common Configuration, Object Classes.
2. Click a link for the object class.

The object class Kind must be Auxiliary.

3. From the View Object Class page, click Delete.

About Object Class Attributes

When installing the Identity System, you configure required structural object classes and their attributes. After completing Identity System setup, you may want to add object classes, configure additional attributes, and modify existing attributes. When adding an object class, you can have the Identity System automatically configure attributes in that object class, as described in ["Adding Object Classes"](#) on page 3-8. Use the Modify Attributes feature to change attributes and to configure additional attributes.

The following sections describe attribute configuration:

- [About Configuring Attributes](#)
- [Attribute Data Types](#)

Note: For Active Directory installations, there is a subset of attributes that are unavailable to schema definition by default. To make these attributes visible to the Identity System for configuration, you must remove their entries from the following three files in the *IdentityServer_install_dir/identity/oblix/data/common* directory: *exclude_attrs_config.xml*, *exclude_attrs-ad.xml*, and *ad_exclude_attrs.xml*. Restart the Identity Server for your changes to take effect.

About Configuring Attributes

When configuring an object in the Identity System, you select a class attribute, as described in ["Selecting a Class Attribute"](#) on page 3-7. In addition, you need to decide how the Identity System will display and work with other object attributes. For instance, you need to decide what facts about a person you want the User Manager to display. You also need to decide how you want to display data. For instance, you may want to display lists of printers on the Organization Manager. Or you may want to display a list of preferred travel agents based on a user's geographical location.

You can configure the Identity System to use any attributes stored in your LDAP directory. Having a well-structured set of attributes to work with allows the Identity System to display the data you want to display and to provide fine-grained access controls for your users.

After configuring an attribute, you must perform additional steps to display the attribute on a profile page in the User, Group, or Organization Manager. For more information, see ["Configuring Tab Profile Pages and Panels"](#) on page 4-12.

After configuring an attribute you must set view and modify privileges to allow users to see the attributes you are displaying. For information about providing users with read and modify privileges, see ["Allowing Users to View and Change LDAP Data"](#) on page 4-22.

Before you configure attributes, you need to understand the relationships between an attribute's data type, semantic type, and the ability to perform searches. These topics are discussed in the following sections.

Attribute Data Types

When you modify an attribute as described in ["About Object Class Attributes"](#) on page 3-10, a data type for that attribute is displayed. A data type is the format of the

attribute's value. For instance, a name attribute may have a data type of a single text line. Every LDAP attribute has an associated data type. In the Identity System, six data types are supported. Data types have corresponding display types, described in ["Attribute Display Types"](#) on page 3-14. You cannot configure the data type for a template or LDAP attribute in the System Console. You configure the data type in the .tpl file or the LDAP schema. Supported data types are shown in [Table 3-2](#):

Table 3-2 Supported Data Types

Data Type	Description	Allowed Display Type
String	A case-insensitive or case-sensitive string.	Boolean, check box, date, email address, filter builder, GIF image URL, multi-line text, numeric string, postal address, radio button, selection menu, single line text
Distinguished Name	Distinguished names are how you refer to entries. A distinguished name (DN) is like the path name for a file, except that the DN is read in the opposite order of a path, from the bottom of the directory.	Object Selector, Location (LDAP data only)
Integer	An integer	None, Boolean, check box, date, email address, filter builder, GIF image URL, multi-line text, numeric string, password, postal address, radio button, selection menu, single line text
Telephone	Telephone number	Any display type
Binary	A binary file, such as a GIF file	GIF image, media, password, S/MIME certificate
Postal Address	This is a compound string consisting of one to six sub-strings concatenated with the dollar sign (\$) as the delimiter. Each sub-string can have a maximum of 30 characters.	Postal address

Attribute Semantic Types

A semantic type is an optional characteristic that governs the behavior of the attribute within an Identity System application. For example, the value of an attribute assigned to the semantic type Photo appears in the header area of a profile page in an Identity System application. You can only assign a semantic type to one attribute. However, an attribute can have more than one semantic type assigned to it. For example, you can assign the Login and DNPrefixed semantic types to the cn attribute.

Once a semantic type is assigned, it cannot be assigned to another attribute within a domain unless you disassociate it from the first attribute. For example, only one LDAP attribute can be assigned the semantic type of Password. If you have configured any other schema domain, you could assign the semantic type of Password to only one attribute in that domain. See ["Sending Non-LDAP Data to External Applications"](#) on page 6-1 for details.

To disassociate a semantic type from an attribute, you must first specify a semantic type of None for the attribute, and then assign a new semantic type to it.

Each semantic type is associated with one or more display types, as described in ["Attribute Display Types"](#) on page 3-14.

Semantic Types Defined During Setup

[Table 3–3](#) describes semantic types that are required during Identity System setup:

Table 3–3 Semantic Types

Semantic Type	Description	Allowed Display Type
Full Name	Required for the person and group structural object classes and for all structural object classes in Organization Manager. Typically assigned to the cn attribute. The cn attribute is required for most schemas.	Check box, Date, Email address, Multi-line text, Numeric string, Radio button, Selection menu, Single line text
Login	Required for the person object class. Specifies the user credentials during login.	Single line text, Email address
Password	Required for password management for the person object class. It is also required for Active Directory. Specifies the user password for password management. Note: If you use Sun's iPlanet directory, passwords cannot use UTF-8 characters. If the user supplies UTF-8 characters, the iPlanet directory default 7-bit plug-in fails the operation. By default the 7-bit plug-in requires the uid, mail, and user password attribute values to be 7-bit. To resolve this problem, turn off the plug-in or remove the user password attribute from the configuration.	Password
DN prefix	Required for the person and group structural object classes and for all structural object classes in Organization Manager. 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. The attribute with this semantic type must be in the initiating step of a workflow.	Check box, date, email address, multi-line text, numeric string, radio button, selection menu, single line text

Semantic Types Used in Profile Pages

[Table 3–4](#) shows semantic types used in profile header panels. For more information about profile panels, see "[Configuring User, Group, and Organization Manager](#)" on page 4-1:

Table 3–4 Semantic Types in a Profile Header Panel

Semantic Type	Description	Allowed Display Type
Photo	Specifies a GIF or JPEG image. The Photo semantic type displays the image in the header of the profile page.	GIF image, GIF image URL

Table 3–4 (Cont.) Semantic Types in a Profile Header Panel

Semantic Type	Description	Allowed Display Type
Title	Displays the attribute value in the header of the profile page. Must be associated with a structural class.	Check box, date, email address, multi-line text, numeric string, radio button, selection menu, single line text
Full Name	Is used in a profile header panel and to personalize the Identity System. Users see their name in the Identity System application user interface.	Check box, date, email address, multi-line text, numeric string, radio button, selection menu, single line text

Semantic Types Used in the Group Manager

Table 3–5 shows semantic types used in the Group Manager:

Table 3–5 Semantic Types Used in the Group Manager

Semantic Type	Description	Allowed Display Type
Group Owner	Specifies the attribute where a group owner is stored. The Identity System uses this information primarily as a role in attribute access and delegated administration. Also, group owners can be notified when a user subscribes or unsubscribes from their groups.	Object Selector
Group Dynamic Member	Specifies the attribute storing the dynamic filter that defines the dynamic membership of a group. If you are configuring the Group Manager, you must assign this semantic type to an attribute. The attribute must also belong to the group object class.	Object Selector
Group Static Member	Specifies the attribute where static members of a group are stored. If you are using the Group Manager, you must assign this semantic type to an attribute. The attribute must also belong to the group object class. For NetScape installations, this attribute is uniqueMember. For Active Directory, it is Member.	Object Selector

Location Coordinates Semantic Type

The Location Coordinates semantic type is used to track location. It specifies the position of the location GIF image and is used with the obRectangle attribute. It has no allowable display type because it is used internally by the Identity System.

Semantic Types for Managing Lost Passwords

Two semantic types are used for lost password management. The Identity System provides lost password functionality for both the Identity System and the Access System. Once you configure attributes with these semantic types, end users can enter a challenge-and-response phrase that can later be used to recover their lost passwords:

Table 3–6 Semantic Types Used for Lost Password Management

Semantic Type	Description	Allowed Display Type
Challenge	Displays a challenge phrase when an end user initiates lost password management. The end user must type the correct response phrase.	Single line text
Response	The end user must type the correct response to a challenge phrase when implementing the lost password management feature.	Password

Other Semantic Types

[Table 3–7](#) describes other semantic types:

Table 3–7 Other Semantic Types

Semantic Type	Description	Allowed Display Type
Preferred Email address	Used to send email notifications	Email address
Map	This semantic type is used with the location feature in the Organization Manager. When an object is configured to be a location, you should configure a binary attribute to be a map semantic type. The binary attribute stores a GIF or JPEG of a map for the location feature.	GIF image
None	This is a place holder and is not a true semantic type. Select None when you do not want to associate an Identity System business rule with an attribute.	N.A.

Attribute Display Types

The display type specifies the appearance of an attribute value, for instance, whether the possible values are True or False or an email address. The display type determines whether the attribute can be used when users conduct a search. Only certain display types such as Date and Multi-Line Text are searchable, as indicated in the following table.

Once you have assigned an attribute to an Identity System application panel as described in ["Configuring Tab Profile Pages and Panels"](#) on page 4-12, if you want to change the attribute display type or semantic type you must delete the panel, change the attribute display type, and re-create the panel.

[Table 3–8](#) describes the attribute display types:

Table 3–8 Attribute display types

Display Type	Description	Configurable Characteristics
None	A place holder.	N.A.
Boolean	Displays a True or False choice that the user must make. This display type is not searchable.	N.A.

Table 3–8 (Cont.) Attribute display types

Display Type	Description	Configurable Characteristics
Check Box	Provides a check box. This display type only supports multiple values, and it requires you to specify a rule or a list. See "Using Rules and Lists" on page 3-19 for details. This display type is not searchable.	Rule (LDAP filter and attribute), List (display name and other features)
Date	Allows users to enter month, day, and year. This display type supports single or multiple values. This display type is searchable.	Data type, data separator
Email	Displays a link to an end user's email address. This display type is searchable.	N.A.
Filter Builder	Creates a button that launches the Filter Builder. The Filter Builder allows users to design custom LDAP queries. This display type is not searchable.	Target object class list
GIF Image	Allows users to find an image. Some Identity System applications also support JPEGs. This display type is not searchable.	Photo style, height, width
GIF Image URL	Allows you to specify an external location for the GIF image. This enables you to display the image on a profile page. This display type is searchable.	Photo style, height, width
Location	Creates a link in the associated profile page to the Locations page. This display type is used internally in the Identity System.	Target object class list
Media	Used for binary media files. This attribute must have the binary data type. This display type is not searchable.	Description, MIME type
Multi-Line Text	Two or more lines of text, such as an address. This display type supports single or multiple values. This display type is searchable.	N.A.
Numeric String	Provides a field for specifying a number. This field does not accept non-numeric characters. This display type is searchable.	N.A.
Object Selector	Use the Object Selector display type when you want users to modify an attribute using the Selector. This display type is only valid for attributes of type DN. This display type supports single and multiple values and is not searchable. For more information on the Object Selector display type, see "Search Filters for the Object Selector Display Type" on page 3-22.	List of object classes, LDAP filter
Password	Lets users type a password. The password characters appear as asterisks, and the user is prompted to enter the password twice. This display type is not searchable.	Text size and length
Postal Address	Six data entry fields in which a user can specify a postal address. Each field can contain a maximum of 30 characters. This display type supports single and multiple values.	N.A.

Table 3–8 (Cont.) Attribute display types

Display Type	Description	Configurable Characteristics
Radio Buttons	Provides a set of radio buttons that allows the user to select one value from the list of radio buttons. This display type requires you to specify a rule or a list. See "Using Rules and Lists" on page 3-19 for details. This display type is not searchable.	Rule (LDAP filter, attribute), list (display name, storage name)
Selection Menu	Provides a list. This display type supports single or multiple selectable values. This display type requires you to specify a rule or a list. See "Using Rules and Lists" on page 3-19 for details. This display type is not searchable. Do not configure DN attributes using the Selection Menu display type. This display type does not support order, which can be important in a DN. For example, if there are two o's in a DN, the order is important.	Rule (LDAP filter, attribute), list (display name, storage name)
Single Line Text	Displays information in a single line of text. There is no maximum number of characters for this field. This display type supports either single or multiple values. This display type is searchable.	N.A.
S/MIME Certificate	Stores certificate data in the configured attribute rather than on disk. This display type is not searchable.	N.A.

Note: In the Identity System Console, the display names that appear as values for items in the list of display types (radio button, checkbox, and so on) may be corrupt due to a known limitation with Java Applets and internationalized characters. The browser's JVM displays only those characters that are in the current locale. Internationalized characters are displayed correctly in applets only if you have set the browser to the same locale.

Viewing Attributes

You view attributes on the Modify Attribute page.

To view the Modify Attribute page from the System Console

- From the Identity System landing page, click the link for the Identity System Console.
If you have already logged in, click the Identity System Console tab.
- Click the Common Configuration sub-tab, then click Object Classes in the left navigation pane.
- Click the link for an object class.
The View Object Class page for the selected class appears.
- Click Modify Attributes.
The Modify Attributes page appears.

To view an application-specific Modify Attribute page

1. From the Identity System landing page, click the link for the Identity System Console.

If you are already logged in, click the Identity System Console tab.

2. In the System Console, click the User Manager Configuration sub-tab.

You can also click the sub-tab for Group Manager Configuration or Organization Manager Configuration.

3. In the left navigation pane, click Tabs.

The Configure Tab page appears. The structural object class for that tab is displayed as a link under the heading "Existing Tabs."

4. Click the link under the Existing Tabs label.

The View Tab page appears.

5. Click the Modify Attributes button.

The Modify Attributes page appears.

Configuring Attributes

When installing the Identity System, you configure all required attributes for your structural object classes. After installation, you can modify attributes to resolve conflicts among configured attributes and to configure additional attributes.

For Active Directory installations, some attributes are unavailable by default. To make these attributes available for configuration in the Identity System, remove their entries from the following three files in the *IdentityServer_install_dir*/identity/oblix/data/common directory: *exclude_attrs_config.xml*, *exclude_attrs-ad.xml*, and *ad_exclude_attrs.xml* and restart the Identity Server.

Note: In the object class *oblixadvancedgroup*, the attribute *obgroupsubscribenotification* has a display type of Check Box and a List sub-type. The list contains two values:

- One value for subscribing (*NotifyUponSubscription*)
- A second value for unsubscribing (*NotifyUponUnsubscription*)

If you want to modify the values for this attribute, note that you can change only the values in the Display Name field. Do not change the value in the Storage field.

To configure an attribute

1. Open the Modify Attributes page as described in "[Viewing Attributes](#)" on page 3-16.

Note: Using the application-specific Tabs function, you can provide a different display name or display type for an attribute on that application-specific Configuration tab only (different than what is configured at the object class level). This will override the information configured for the attribute at the object class level. For details, see ["Modifying and Localizing Attributes Displayed on a Panel"](#) on page 4-21.

2. In the Attribute list, select an attribute you want to modify.

The attribute's data type appears after the list. This is a read-only field.

Note: Novell Directory Server (NDS) maps the attribute and object class names from the native directory server to the LDAP layer of NDS. Some of these attributes or object classes will have multiple mappings (aliases) in the LDAP layer. For example, the native NDS object class is group, while the LDAP layer of NDS maps two aliases called GroupofNames and GroupofUniqueNames. For the Identity System to work correctly, make sure the object class or attribute name that you provide during configuration is the one that occurs ahead of the other mappings for the same object class or attribute. You can check the mapping order through consoleOne.

3. In the Display Name field, enter a user-friendly display name for this attribute.

The display name appears on an Identity System application page, for instance, the User Manager. For example, for the departmentNumber attribute, you might enter Department Number as the display name.

For template object attributes, the Display Name should indicate the template being used. As noted earlier, users will not be able to see the data values for these attributes. As a result, you should identify these "special case" fields so that users can be advised that it is normal for data not to be shown. For example, in an object template for application ABC, you might want all ABC-related attributes to be identifiable by their display names, such as assistant.person.abc.

The Data Type field displays the attribute's data type, as described in "Attribute Data Types" on page 3-10. This is a read-only field.

You cannot use attributes with binary, distinguished name, or postal address data types as report criteria or in search attributes.

4. In the Semantic Type list, you can optionally select one or more semantic types.

See ["Attribute Semantic Types"](#) on page 3-11 for details on semantic types.

5. In the Attribute Values field, specify whether the attribute can have single or multiple values.

Depending on the attribute, data type, and display type, this option may not be available.

6. In the Display Type list, select the attribute's display type.

If you select a date attribute for the display type, you must select a date type to indicate how you want the date to appear on the Identity System application

profile page. Do not change the date type after you select it because this may display existing data incorrectly.

Several display types allow you to specify a rule or list. See ["Using Rules and Lists"](#) on page 3-19 for details.

Other display types allow you to specify a photo or text. See ["Configuring Other Display Types"](#) on page 3-27 for details.

Using Rules and Lists

The display types of Selection Menu, Radio Buttons, and Check Box require that you specify a rule or a list. For instance, you may assign a data type of string and a display type of radio button to a Title attribute. To display a list of titles on a User Manager profile page, you need to build a list.

A *list* is a static set of values. A *rule* is an LDAP filter that queries the directory before building a list. For example, if you create a filter to find every instance of objectClass=dialUpConnection with an attribute of TelephoneNumber, a list of phone numbers is shown in the selection menu.

For more information about LDAP filters, refer to ["Search Filters for the Object Selector Display Type"](#) on page 3-22 for information. Also, the Internet Engineering Task Force's RFC 2254 defines a string representation of LDAP search filters.

Defining a Rule

You can define a static list to display on an Identity System application page, or you can define a rule. For instance, you can provide a static list of available printers, or you can construct this list from entries for printers in your directory. When you configure a rule, you create a dynamic list based on entries in your directory. A rule is a directory query based on an attribute that you specify in the rule. The query returns a set of records from the directory. Using the rule, you cause the Identity System to build the list to be displayed on the application page by doing the following:

- Querying the directory for a specific object or attribute
- Building a list of hits
- Selecting one attribute from each directory hit
- Building a list showing the values for each attribute

The advantage of a rule over a list is that what is displayed as a result of the rule filter is updated whenever the directory is updated.

To define a rule

1. Open the Modify Attributes page as described in ["Viewing Attributes"](#) on page 3-16.
2. Select an Attribute from the list.
3. From the Display Type list, select the attribute's display type.

To configure a rule for the attribute, the display type must be Selection Menu, Checkbox, or Radio Buttons.

A Rule button, Add Filter text box, and Attribute field appear.

4. Select the Rule button.

A rule *must* be a valid LDAP filter.

5. In the Add Filter box, type an LDAP filter.

For example:

```
(objectclass=printer)
```

Suppose you invoked the Modify Attribute page for an attribute called Printer, with a display name of Printers. The rule shown in this step would be appropriate for displaying a list of printers.

For examples of filters, see ["Search Filters for the Object Selector Display Type"](#) on page 3-22.

6. In the Attribute field, type the LDAP name of the attribute that you want to associate with the filter.

In the printer example, you might type PrinterName. This rule causes an LDAP query on the printer object class and builds a list of values taken from the PrinterName attribute of each returned entry.

7. Continue with ["Defining a List"](#) on page 3-20.

Defining a List

A list is a static set of values presented to a user.

To define a list

1. On the Modify attributes page, click the List button.

List-related Display and Storage fields become active.

2. In the Display field, type the list item's Display Name.

This is a name that the user sees when this attribute is displayed on an application page, for instance, the User Manager.

3. In the Storage field, type a storage name for the attribute.

This value is saved in the database. It can be the same as the display name, or it can follow your own database-naming conventions.

When you click Add, if you omit a storage name, the display name is used as the storage name.

If you want to change a storage name, delete the entry in the Storage field, and retype the Display and Storage names.

4. Click Add.

The information is added to the List field.

Items in the list appear on the Identity System application page in the order they appear on this page. To rearrange items in the list, or to remove an item, use the Move Up, Move Down, and Delete buttons.

Localizing Attribute Display Names

You can localize attribute display names to present to Identity System applications to end-users in their native language. See ["Configuring Multiple Languages for Oracle Access Manager"](#) on page 7-7 for information on managing multiple languages.

In order to localize object class attributes, you must manually enter the attribute display names in the Identity System Console for each language that you installed.

After you have localized object class attributes, you can view and modify them in the Identity System Console.

The process for localization is the same for LDAP and template objects.

[Table 3–9](#) lists the attributes that can be localized for each object class.

Table 3–9 Items You Can Localize

Items	What You Can Translate
Object classes configured for tabs	Name Description Mouseover
Object classes configured for panels	Name Description Mouseover
Attributes	Display name
Attributes with a media display type	Display name
Attributes with a choice display type	Display name
Workflow definitions	Workflow name Note: You specify a translation for your workflow name when you create or modify a workflow definition, as described in "Chaining Identity Functions Into Workflows" on page 5-1.

To create, view, or modify localized attribute display names

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

The Common Configuration page appears.

2. Click the Object Classes link in the left navigation pane.

3. Click the link for the object class you want to modify.

The View Object Class page appears.

4. Click Translate.

Note: The Translate button appears only if more than one language has been installed.

The Summary of Attribute Display Names page appears. All language-specific attribute display names appear on this page. Display names that have not been configured are marked as Not Configured and appear in blue text.

5. Click Modify to enter or modify a display name.

The Configure Attribute Display Names page appears. This page lists links for installed languages and the localized display names for attributes. Display names that have not been configured appear in blue text and are not flush left.

6. Click the language for which you want to modify attribute display names.

7. Enter display names in the Display Name fields.

8. Click Save to save your changes (or Cancel to exit the page without saving your changes).

Note: If a display name has not been configured for a language, a localized "Not Configured" message is displayed in the display name field.

Search Filters for the Object Selector Display Type

The object selector display type associates a Selector with the LDAP attribute assigned to this display type. (This does not apply to template attributes.) Users invoke the Selector to search for users or groups. The Selector is available when you view, create, or modify a profile or a workflow.

Use the object selector display type to create a search filter for the Selector. See "[The Selector](#)" on page 1-10 for details. You can write search filters to help people conduct a Selector search during the following operations:

- Create profile
- Modify profile
- Modify workflow
- Delete workflow

These filters do not apply to the Query Builder.

When a user invokes the Selector, they perform a directory search. When you create a filter for the Selector, the filter is used in an "and" relationship with the search criteria that the user provides.

A filter can be static. For example, you can restrict Selector searches so that the search results only contain people with an organizational unit of Corporate in their directory profile.

A filter can also be dynamic. For example, you can restrict a Selector search to return only people whose organizational unit matches that of the person being displayed on the Modify Profile page where the search was invoked. When using a dynamic filter for a Modify Profile page, the Selector search is based on the directory profile of the person being displayed. In the case of creating profiles, a dynamic filter produces Selector search results based on the login profile of the person performing the task.

Creating a Search Filter for the Object Selector Display Type

A filter helps the user narrow down a search. A filter narrows down the place in the directory tree where a search may be conducted.

To create a filter

1. From the Identity System Console, click Common Configuration, then click Object Classes.
2. Click the link for the object class for which you want to create a filter.

For example, to develop a search filter for a sales person, you might navigate to the Modify Attribute page for the Person object class.
3. Click Modify Attributes.
4. On the Modify Attribute page, in the Attribute list, choose the attribute for which you want to define a Selector search.

You must choose a DN attribute. For instance, if you want Selector searches for sales people, you might select a DN attribute called salesPersonDN.

5. On the Display Type list, select Object Selector.

If the attribute you chose in the previous step is a DN attribute, the Object Selector option appears in the list. The Target Object Class list and the Add Filter input box are also displayed.

6. In the Target Object Class list, select an object class to be used as the primary key in the search filter.

The target object class determines what is displayed on the Selector search page. For instance, if you want the Selector to help users find sales people, you might use Person as the target object class. If you select more than one Target Object Class on the Modify Attribute page, the Selector application will contain a tab for each object class.

7. Type a valid LDAP filter in the Add Filter input box.

The filter determines what is displayed on the Selector search results. For examples of the types of LDAP filters you can write, see ["Static LDAP Search Filters"](#) on page 3-24 and ["Examples of Dynamic LDAP Search Filters"](#) on page 3-25.

Note that your filter can use only attributes that are contained in the definition of the target object class.

Note: the Identity System treats white spaces literally. Be aware of extra trailing spaces or carriage returns in your filters.

8. Save your changes.

This creates a filter that is used in an And relationship with any other criteria the user specifies on the Selector search.

Search Filters for Multiple Target Object Classes

If you select more than one target object class on the Modify Attribute page, the Selector application will contain a tab for each object class. Your search filter will need to contain an Or operator (|) and separate selection criteria for each object class you selected. An example of this type of search filter is provided in ["Static Searches Using Multiple Target Object Classes"](#) on page 3-24.

Deleting a Search Filter

Remove a filter by erasing the text in the Filter text box.

Usage of Rules and Filters

This section covers important topics related to rules and filters:

- Creating basic static filters
- Creating dynamic filters using substitution syntax
- Use of wild cards in a search
- Proper use of the Not operator when writing a filter

Static LDAP Search Filters

When you implement a static search filter, all search results must match a fixed value. For example, you can restrict a search to return only people whose directory profiles show an organizational unit of Sales.

As an example of a simple static filter, suppose you want to provide Selector searches for the seeAlso attribute. The filter will return search results that show only people whose directory profiles contain a businessCategory value of dealership.

To create a static filter

1. Open the Modify Attributes page as described in ["Viewing Attributes"](#) on page 3-16.
Navigate to the Modify Attribute page for the object type that triggers this search filter (for instance, Person).
2. Select the seeAlso attribute in the Attribute list.
3. From the Display Type list, select Object Selector.
4. In the Target Object Class, select the object class that will be the primary key for the filter (for instance, Person).
5. In the Filter text box, input the following:

```
(businessCategory=dealership)
```

Static Searches Using Wild Cards

As an example of a static filter that uses wild cards, suppose you want only people with the word Manager in their title to be returned on a search using the Selector. You can create a filter that searches for the string Manager.

To create a static search filter using a wild card

1. Navigate to the Modify Attribute page for the object class containing the DN attribute to be associated with the Selector (for instance, the Person object class).
2. In the Attribute list, select the DN attribute (for instance, the Manager attribute).
3. On the Display Type list, select Object Selector.
4. In the Filter text box, input the following:

```
(attribute=*value*)
```

For example:

```
(title=*manager*)
```

Static Searches Using Multiple Target Object Classes

You can build a static filter that searches for more than one target object class. For example, suppose you want to build a filter for the uniqueMember attribute so that a search using the Selector returns one of two items:

- On a search of people, the search results show only full time employees
- On a search of groups, the search results show only mail groups

In this example, to create an LDAP filter for both characteristics, you need to select people with employee=fulltime in their directory profiles, and you need to select groups with the object class of MailGroup in their directory profile. Each attribute in

the filter must be associated with an appropriate object class. Finally, you need to join the two searches with an Or operator (`|`), as follows:

```
( | (&(objectclass=inetOrgPerson) (employeeType=FullTime))
  (&(objectclass=groupOfUniqueNames) (objectclass=MailGroup)))
```

Substitution Syntax: Returning Targets that Match the DN of the Logged In User

You can enter substitution syntax using the Advanced button of the Query Builder. See ["Writing LDAP Filters Using Query Builder"](#) on page 4-29 for details. When using substitution syntax, the attribute value for the source DN (the person logged into the application) is substituted in the rule and evaluated against the target DN (the entry you are trying to view or modify).

Note that setting a searchbase can present significant administrative overhead. A filter-based approach that is provided by substitution syntax can provide the same functionality in a more scalable and simplified design.

Using substitution syntax, you can create a function that starts searches higher in the directory structure, but filters the search data by comparing an attribute of information from the search initiator's record (for example, using the substitution `$ou$`) to an attribute of data on each possible result (for example, `ou=`). You can use substitution syntax for attribute access control and searchbases. For example, by placing a filter on the class attribute `Login` for `inetOrgPerson`, the ability of a user to view any records outside their scope is removed.

Substitution syntax allows a rule to be evaluated dynamically, according to the person executing a task. The syntax is as follows:

```
attribute=$attribute$
```

For example:

```
(ou=$ou$)
```

This rule filters all those in the same organizational unit as the person logged into the application.

You can use operators such as And and Or in a filter. For example:

```
( | (ou=$ou$) (ou=people) )
```

Note: For the selected searchbase, users can search only for entries from the same ou as their own. This applies only to the attribute on the person's record, not the ou of the branch of the directory in which they reside. Additionally, users from `ou=people` can search for entries within the selected searchbase.

Examples of Dynamic LDAP Search Filters

In addition to specifying a conventional LDAP search filter, you can use the Identity System's filter substitution syntax to create a dynamic filter. A dynamic filter allows a search to return results that are based on a user profile. For instance, suppose you create a search filter for the `orgPerson` attribute that contains the following:

```
(ou=$ou$)
```

Using this search filter, if you conduct a Selector search on a Modify Profile page for a person, your search results contain only people whose directory profiles match the organizational unit in the profile of the person you are modifying. For example, if you

invoke the Modify Profile page for John Smith and invoke the Selector to choose John Smith's manager, the search results show only people in John Smith's organizational unit.

When you use filter substitution, the profile of the search target is substituted. In the example (ou=\$ou\$), the value of John Smith's ou is substituted. If a target is not present, for instance, in the Create Workflow function, the search filter substitutes the value from the login profile of the person creating the workflow.

For example, suppose you are creating a workflow for a group named Corporate whose organizational unit is not yet defined in the directory server. In this case, the Identity System uses the ou value of the logged-in participant who is creating the group. The logged-in participant's ou value carries over in the workflow until you commit this group in the directory server. At that point, the ou value in the dynamic filter (ou=\$ou\$) changes to the group's ou value.

As another example, suppose you want people who have the Secretary attribute in their directory profile to receive search results containing only people who have the same manager that they do. From the Configure Attribute page for the Secretary attribute, you would specify the following:

```
(manager=$manager$)
```

Dynamic Searches Using Wild Cards

You can use wild cards in a dynamic filter. For example, suppose you want to supply a contactPerson attribute in an organizationalUnit object. The contactPerson attribute should return people in same Zip code as the organizationalUnit object. If the organizationalUnit profile contains an attribute called zipCode, and the Zip code is specified at the end of a postalAddress directory attribute, you would specify the following in the filter:

```
(postalAddress=*$zipCode$)
```

Dynamic Searches Using Multiple Values

Finally, you can supply multiple values in a dynamic search filter. For example, suppose you want the seeAlso attribute for business objects to select the businessCategory of dealership, and specifically, dealerships in the same state as the search target. You would specify the following filter for the seeAlso attribute:

```
(&(businessCategory=dealership)(state=$state$))
```

Use of the Not Operator

You use And (&) and Not (!) operators when constructing a filter. For example:

- **Example of an And Operation**--(&(sn=white)(objectclass=personOC))
- **Example of a Not Operation**--(&(!(sn=white))(objectclass=personOC))

In the example of the Not filter in the previous example, you might expect the following filter to be valid:

```
(!(sn=white))
```

However, when specifying a Not operation, you need to embed it within an And and specify the person object class. A filter such as (!(sn=white)) is not allowed because a search of this type would be conducted on the entire domain before targeting the reduced domain set specified on the filter. This is costly steps in terms of performance. The optimized algorithm that the Identity System uses causes the search to be

conducted on the reduced domain set. The proper use of the Not operation is as follows:

```
( & ( ! ( sn=white ) ) ( objectclass=personOC ) )
```

The optimized algorithm causes the filter `!(sn=white))` to not give the expected result.

Configuring Other Display Types

This section discusses configuration of text-based display types, radio buttons, check boxes, the GIF image display type, and display types that use target object classes.

You can configure default values for a display type that contains text fields, for example, single line text and multi-line text attributes, radio buttons and checkboxes. You use XSL stylesheets to configure the defaults. This is also true for setting columns and rows. See the *Oracle Access Manager Customization Guide* for details.

When you select an attribute with a display type that uses target object classes, you specify one or more objects for association with this attribute. This display type supports single or multiple values.

- If you want to set a target object class, select one or more required object classes in the Target Object Class list.
- If you want to set the MIME type, select the kind of media file you want to attach from the MIME Type list.

To configure a GIF image display type

1. On the Modify Attributes page, select the Attribute Photo.
2. Using the buttons under the Display Type list, select the photo style:
 - **True Size**—Select True Size to display the GIF in its actual size.
 - **Fixed Size**—Select Fixed Size to specify the height and width for the image.

Configuring Derived Attributes: Matching Values from Different Attributes

Derived attributes are virtual LDAP object class attributes. Derived attributes are generated by comparing the contents of one object class's attribute with an attribute in the same or a different object class. The main purpose of a derived attribute is reverse lookup.

For example, someone's profile may contain the name of their administrative assistant, but administrative assistant profiles rarely contain the names of all of the people they administer. A derived attribute allows the administrative assistant's profile to refer back to all of the people who have the administrative assistant attribute in their profile. In this example, the administrative assistant's Self attribute value is compared with the AdminAssistant attribute value of the other people in the organization. Similarly, a groupOfUniqueNames object may contain a uniqueMember attribute with links to members of the group. But an object that contains data for a person might not link back to the groupOfUniqueNames object. The derived attribute feature would allow group members to refer back to the group they belong to.

To create derived attributes, you specify two attributes whose values are to be compared. All matches are added to the derived attribute.

Use derived attributes to provide information in profiles that otherwise would require an LDAP filter, search, or report.

Attributes associated with template objects cannot be configured as derived attributes.

Note: Using derived attributes can result in slower response times, especially if you have multiple attributes or if your attribute references multiple object classes (such as a Group Manager derived attribute performing a lookup on a User Manager attribute). If you experience performance issues with your derived attributes, Oracle recommends you modify your index file to include the corresponding attribute index and reimport it.

Example of a Derived Attribute

Suppose the administrative assistants of your organization have asked to view all the managers they support in the User Profiles in User Manager. To do this, you can create a derived attribute that takes the attribute of each administrative assistant and compares it with the value for Secretary in everyone else's profile. When an administrative assistant views user identities in User Manager, only people who match the derived attribute are displayed.

Here is a summary of the steps required to create a derived attribute.

Task overview: Configuring a derived attribute

1. Add a derived attribute in the Identity System Console, giving your new attribute a descriptive display name, such as My Direct Reports.
2. Specify Self as the Match Attribute.

This indicates that the administrative assistants' DNs are the search criteria.

3. Because you are looking for your administrative assistant's direct reports, specify Person object class as the searched object class.

Be sure you are looking for people and not groups or other kinds of objects.

4. Specify secretary as the Lookup Attribute.

You are searching the user identities for users with matching values in the secretary attribute.

5. Save your new attribute and associate it with the Employee tab in User Manager.

Now, whenever an administrative assistant views the User Profile page, the Identity System takes the value of the Self attribute (DN) and compares it against the values of everyone's secretary attribute. Wherever there is a match, the Identity System lists that manager's name in the administrative assistant's My Direct Reports section of the User Profile page.

Note: The attributes displayed in the profile are determined by the selected object class's Object Class Attribute. To change this value, you must modify the object class.

To configure a derived attribute

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

The Common Configuration page appears.

2. Click the Object Classes link in the left navigation pane and click the named link of an object class.

The View Object Class page appears.

3. Click the link for the object class whose derived attribute you want to modify.
4. Click Modify Derived Attributes, then click Add.
The Create Derived Attribute page appears.
5. In the Attribute Name field, specify a name for your new derived attribute.

Note: Since a derived attribute is a virtual attribute, the attribute name must not exist in the schema.

6. In the Display Name field, type the name of the derived attribute as it will appear in the Identity System pages.
7. In the Match Attribute field, select the attribute in the current object class whose values you want to match.
8. In the Object Class field, select the object class you want to search.
9. In the Lookup Attribute field, select an attribute in the specified object class whose values you want to compare against.
10. Click Save to save your changes (or Cancel to exit without saving).

Assigning a Derived Attribute to a User Manager Tab

Before you can use your derived attribute, you must assign it to an Identity System application tab.

To add a derived attribute to an application tab

1. From the Identity System Console, click User Manager Configuration (or Group Manager Configuration or Organization Manager Configuration).
2. Click Tabs.
3. Under the Existing Tabs label, click the link for the application tab that you want to modify, then click View Object Profile.

The page that appears depends on which application you are using. In User Manager, the Configure User Profile page appears. In Group Manager, the Configure Group Profile page appears. In Organization Manager, the Configure Object page appears.

4. If you are working with User Manager Configuration or Organization Manager Configuration, click Configure Panels.

For the Group Manager, the link name is Configure Group Profiles Panel.

The Configure Panels (or Configure Group Profile Panels) page displays the panels currently configured to appear in a User Manager profile.

5. Click the name of the panel to which you want to add the derived attribute.
The View Panel page appears.

6. Click Modify.
The Modify Panel page appears.

7. In the Attributes menu, select the derived attribute you want to add.
8. In the associated text box, type the name as you want it to appear in the Identity System pages.

9. If you need additional Attribute fields, click Add.
10. Click Save to save your changes (or Cancel to exit without saving).

Permissions for Derived Attributes

You can assign Read permissions to a derived attribute. See ["Setting and Modifying LDAP Attribute Permissions"](#) on page 4-33 for information about assigning rights to an attribute.

Attributes Configured for an Individual Application

When you configure attributes as described in this chapter, these attributes are available to any Identity System application. In other words, the attribute can be used within the User Manager, Group Manager, or Organization Manager.

However, you may want to make only certain attributes available to certain applications. For example, you may want an attribute for a person's address to only be available to the User Manager application. If this is the case, you can access the object and attribute configuration functionality described in the preceding sections from the relevant application. In the case of the Organization Manager, you can configure objects and attributes for individual tabs.

For more information on configuring the User, Group, and Organization Managers, see ["Configuring User, Group, and Organization Manager"](#) on page 4-1.

Configuring User, Group, and Organization Manager

The User Manager, Group Manager, and Organization Manager are Identity System applications that enable end users to view and modify information about themselves, other people, groups, inventory, and any other item that you, the administrator, choose to make available.

The chapter on ["Making Schema Data Available to the Identity System"](#) on page 3-1 describes how to make the Identity System aware of objects and attributes in your directory and in object template files, and how to configure the way that attributes are displayed on an application page. This chapter explains how to configure the Identity applications—the User Manager, Group Manager, and Org. Manager—by placing attributes on specific application pages and enabling users to view and modify the attributes. This chapter also touches on end use of these applications.

You must be a Master Identity Administrator or Delegated Identity Administrator to configure the User Manager, Group Manager, and Organization Manager applications. See ["Delegating Administration"](#) on page 2-5.

This chapter covers the following topics:

- [About User, Group, and Organization Manager](#)
- [Configuring Tabs](#)
- [Configuring Tab Profile Pages and Panels](#)
- [Allowing Users to View and Change LDAP Data](#)
- [Examples of Configuring an Application](#)
- [End-User Scenarios for the Group Manager](#)
- [Configuring Auditing Policies](#)
- [Generating Reports](#)
- [Advanced Configuration](#)

Note: There are a number of considerations for performance when managing groups in the Group Manager application. See the chapter on performance tuning in the *Oracle Access Manager Deployment Guide* for details.

About User, Group, and Organization Manager

The User, Group, and Organization Manager are the primary Identity System applications:

- People use the User Manager to view information about their identity, to modify information such as their home phone number, and to find information on other people.
- People use the Group Manager to view groups, subscribe to groups, and to manage group subscriptions.
- People use the Organization Manager to manage other objects.
- Popular uses of the Organization Manager include viewing organization maps and searching for inventory items.

You control who is allowed to see what attributes and values on these applications. You also control what portion of the directory tree is accessed when users conduct searches. You can add filters so that when users search, the results conform to criteria specified on the filters.

When you first install, set up, and configure objects and attributes in the Identity System, the Identity System application pages are empty. You make information available on an Identity System application as follows.

Task overview: Displaying information on an application

1. Configure the objects and attributes to be used by the Identity System applications, as described in ["Making Schema Data Available to the Identity System"](#) on page 3-1.
2. Configure the Identity System application pages, or tabs, as described in ["Configuring Tabs"](#) on page 4-2.
3. Configure profile pages on each tab by arranging groups of attributes into panels, as described in ["Configuring Tab Profile Pages and Panels"](#) on page 4-12.
4. Optionally, set the searchbase to control what portion of the directory tree is included in a search for LDAP attributes only, as described in ["About the Searchbase"](#) on page 4-23.
5. Set permissions for users to view and modify the attributes you are displaying on the application tabs, as described in ["About View and Modify Permissions"](#) on page 4-32 for details.

Configuring Tabs

The Identity System applications each have one or more tabs, which are configured as follows:

- When you work with the Group Manager Configuration tab in the Identity System Console, you are configuring the My Groups tab in the Group Manager application.
- Similarly, the User Manager Configuration tab in the Identity System Console is used to configure the My Identity tab in the User Manager application.
- Unlike the User Manager and the Group Manager, you can configure multiple tabs in the Organization Manager.
- When you install the Identity System, a default structural object class Location is defined for the Organization Manager. The Organization Manager can manage

objects defined in the Identity System as having a Generic or Location data type. See "[Object Class Types](#)" on page 3-5 for details.

The tab in the User Manager is associated with the person structural object class. The tab in the Group Manager is associated with the group structural object class. The Organization Manager can have multiple tabs, each associated with a different object class. All tabs may have auxiliary LDAP object classes and template object classes associated with them.

Viewing and Modifying Tab Configuration Information

You can view and modify characteristics of the tabs that are displayed on the User, Group, and Organization Manager pages.

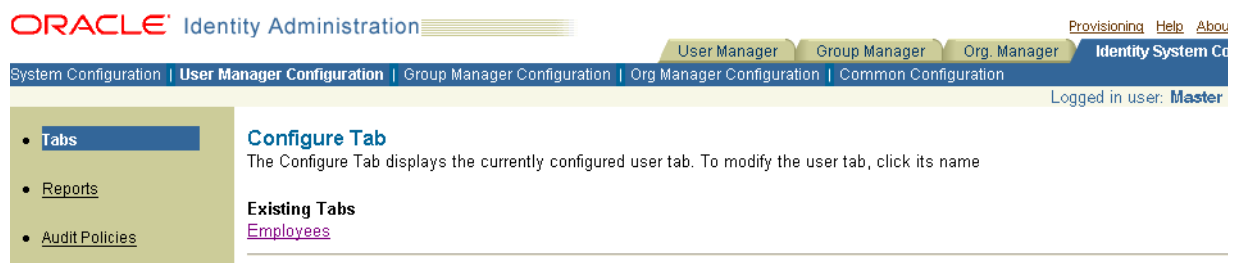
To view or modify tab configuration information

1. Navigate to the Identity System Console and click User, Group, or Org. Manager Configuration.

The User, Group, or Organization Manager Configuration page appears.

2. Click Tabs.

The Configure Tab page appears, showing the name of the tab for the application. The Organization Manager may have more than one tab.



3. Click the link for the tab.

Since there is only one tab for User Manager and Group Manager, there can only be one link. For Organization Manager, there can be more than one tab.

The View Tab page appears.

ORACLE Identity Administration [Provisioning](#) [Help](#) [About](#)

User Manager Group Manager Org. Manager Identity System Console

System Configuration | **User Manager Configuration** | Group Manager Configuration | Org Manager Configuration | Common Configuration

Logged in user: **Master**

- Tabs**
- [Reports](#)
- [Audit Policies](#)

View Tab

The View Tab screen displays information about the User Tab. To change any information about the User Tab, click Modify. By clicking View Search Result Attributes, you can view the attributes that will be displayed in the table view of the search results. To configure the Profile Page for this object, click View Object Profile. To select/deselect the attributes that can be searched in the User Manager, click View Search Attributes. To modify display name, semantic type, display type and attribute cardinality for the attributes of this tab, click Modify Attributes.

Tab ID	Employees
Tab Name	Employees
Tab Image	(no image)
Depressed Tab Image	(no image)
Mouse Over Message	
Class Type	
Object Class(es)	gensiteorgperson OblixOrgPerson
Tab Filter	
Tab Search base	

[Modify](#) [View Search Result Attributes](#) [View Object Profile](#) [View Search Attributes](#) [Modify Attributes](#) [Back](#)

Table 4–1 describes the information on this page:

Table 4–1 Tab Information

Field	Description
Tab ID	Unique identifier for the tab.
Tab Name	The name displayed on the application tab. You can localize this field.
Tab Image	GIF image for the tab. The GIF must be stored in <i>WebPass_install_dir/identity/oblix/lang/langTag/style0</i> where <i>WebPass_install_dir</i> is the directory where you installed WebPass and <i>langTag</i> is the folder that contains the specific language that you are using. Enter only the name of the GIF file, not the full path.
Depressed Tab Image	GIF image displayed when a user clicks the tab image.
Mouse Over Message	Text displayed when the user passes the cursor over the tab. You can localize this field.
Class Type	The type associated with the structural class for this tab. See "Object Class Types" on page 3-5 for details.
Object Class(es)	The structural, auxiliary, and template object classes used by this tab. Template object classes are shown in fully qualified format, for example, <i>miis.person</i> . The format is read from the <i>.tpl</i> file where the class is defined. See "Sending Non-LDAP Data to External Applications" on page 6-1. You cannot change the structural object class through Identity System Console. You can associate auxiliary object classes with the structural object class, as described in "Adding Auxiliary and Template Object Classes to a User or Org. Manager Tab" on page 4-8. Note that some object classes may appear in a non-editable list on this page, and other object classes may appear in a text box on this page. The object classes in the text box have not yet been added to the tab.

Table 4–1 (Cont.) Tab Information

Field	Description
Tab Filter	An LDAP filter that queries the directory and returns objects qualified by the filter. For examples of the types of LDAP filters you can write, see "Static LDAP Search Filters" on page 3-24 and "Examples of Dynamic LDAP Search Filters" on page 3-25. Tab filters do not support filter substitution. Tab filters affect searches, viewing and modifying profiles, and creating reports on the tab. The filter is used in an "and" relationship (with criteria specified during a search) and when creating reports. That is, the criteria from both the filter and the search are applied. View and modify operations use this filter to qualify the target object.
Tab Searchbase	Starting point in the directory tree (DIT) for user searches. See "About the Searchbase" on page 4-23 for details.

4. Click Modify.
5. Make the desired changes and click Save.

If you do not see your changes reflected in the Identity System application, go to Identity System Console, System Configuration, View Server Settings, and click Clear Cache to flush and reload the cache.

Note: When you modify a tab image, depressed tab image, and so on, these elements are immediately available for users to view. This is different from adding attributes to a panel, which requires setting permissions before users can view the information.

Localizing Tabs

If you have installed more than one language pack, you can localize tab names to display them in those languages. You create, view, and modify localized tab names in the Administration Console.

See ["Configuring Multiple Languages for Oracle Access Manager"](#) on page 7-7 for information on managing multiple languages.

Note: The Translate button appears only if more than one language has been installed.

To create, view, and modify localized tab configuration

1. Log in to the Identity System Console and click User Manager Configuration (or Group Manager Configuration or Org. Manager Configuration).

The User, Group, or Organization Manager Configuration page appears.

2. Click the Tabs link in the left navigation pane.

The Tab page appears, showing the name of the tab or tabs for the application.

3. Click the link to an existing tab to view its details.

The View Tab page appears. Tab details such as the ID, name, class type, and object classes are displayed on this page.

4. Click Translate.

If this button does not appear on the page, you have only one language installed and you cannot localize the display names. Click Modify to change the display name for the single language that you have installed.

The Summary of Tab Label Display Names page appears. Display names, if any, that have been configured for the following language-specific fields appear on the page:

- Tab Name
- Mouse Over Message

Display names that have not been configured for a particular language are marked as Not Configured.

5. Click Modify to enter a tab display name or to modify an existing one.

The Tab Display Names page appears. This page contains fields for the tab display names and links for all the installed languages.

6. Click the language for which you want to localize the tab.
7. Enter the display names in the Tab Name and Mouse Over Message fields.
8. Click Save to save your changes.

Adding a Tab to the Organization Manager

The Organization Manager can contain more than one tab.

To add a tab

1. From the Identity System Console, click Org. Manager Configuration, then click Tabs.

The Configure Tabs page appears.

2. Click Add.

The Create Tab page appears.

3. Complete the fields in this page, as described in "[Viewing and Modifying Tab Configuration Information](#)" on page 4-3.
4. Click Save.

Specifying the Search Attributes on a Tab

At the top of the application page for the User Manager, Group Manager, and Organization Manager there are search fields. See "[Adding Auxiliary and Template Object Classes to a Group Tab](#)" on page 4-9 for an example. You specify the attributes that you want to appear in the search function list. Note that search attributes can only be taken from an LDAP directory. Template attributes cannot be used as search attributes.

Note: You must configure attributes before they can appear on a tab. For more information, see "[About Object Class Attributes](#)" on page 3-10.

To specify what attribute can be used in a search

1. From the Identity System Console, click User Manager Configuration (or Group Manager Configuration or Org. Manager Configuration), and click the Tabs link in the left navigation pane.
2. Click the link for the tab.
3. Click View Search Attributes.
The View Search Attributes page appears.
4. Click the Modify button.
5. Select an attribute check box to make the attribute searchable.
6. Click Save.

Viewing, Modifying, and Localizing Attributes that Appear in Search Results

You choose what attributes are to appear in the results of a search. If you have installed and configured multiple languages, you can localize search result attributes. This enables you to display search results in multiple languages.

To view the search result attributes

1. From the Identity System Console, click User Manager Configuration (or Group Manager Configuration or Org. Manager Configuration).
2. Click the Tabs link in the left navigation pane
3. Click the link for the tab.
4. Click View Search Result Attributes.

The View Search Result Attributes page appears.

This page shows the attributes that appear when the results of a user's search are displayed. If you have configured the Identity System for more than one language, those languages are displayed on the page.

5. Click Modify to change the attributes.

The Modify Search Result Attributes page appears.

The first attribute is always the Class Attribute.

You cannot modify the class attribute Name from this page. It is displayed in bold and is not editable on this page. If you want to modify a class attribute, see ["Selecting a Class Attribute"](#) on page 3-7 for details.

6. From the attribute lists, select new attributes for each search field you want to change.

The attribute's Display Name appears in the editable field to the right of the attribute list. This name appears in the Identity System user application. See ["About Object Class Attributes"](#) on page 3-10 for details.

7. Click Add if you need additional attribute fields.
8. Click Save.

To localize search results

1. In the Identity System console, select User Manager Configuration (or Group Manager Configuration or Org. Manager Configuration).

2. In the left navigation pane, click Tabs then click a link.
The View Tab page appears.
3. Click View Search Result Attributes to display the View Search Result Attributes page.
4. Click Translate.

Note: The Translate button appears only if more than one language has been installed.

The Summary of Search Results Attribute Display Names page appears. Existing display names in all the locales are listed on this page. Display names that have not been configured for a language are marked as Not Configured.

5. Click Modify to configure a display name for a language.
The Search Results Attribute Display Names page appears.
6. Click the language for which you want to configure a display name.
7. Enter the name in the Display Name field.
8. Click Save to save your changes.

Adding Auxiliary and Template Object Classes to a User or Org. Manager Tab

You can use auxiliary object classes as mix-ins with structural object classes. For instance, if you have an auxiliary class for a person, and the auxiliary contains an attribute for the person's badge number, you might want to associate this auxiliary class with your structural object class. When you configure the User Manager, Group Manager, and Organization Manager applications, the more object classes you have at your disposal, the more information you can configure for users of those applications.

If you have created workflows as described in "[Chaining Identity Functions Into Workflows](#)" on page 5-1, there are issues when associating an auxiliary object class with a tab:

- If the tab has associated workflows with pending requests, you cannot attach an auxiliary object class.
- If the auxiliary object class you are attempting to attach has any required attributes, you must edit all associated workflows to include those attributes.

You can also associate template objects with a tab. This is required if you plan to configure a workflow that makes use of the template object.

Note: You cannot remove an auxiliary object class you have added to a User Manager or Organization Manager tab. In Group Manager, under Group Types, you can remove an auxiliary class.

To add an auxiliary or template object class to a User or Org. Manager tab

1. Ensure that you have configured the object class that you want to add in the Common Configuration tab.

See "[Adding Object Classes](#)" on page 3-8 for details.

2. From the Identity System Console, select User Manager Configuration (or Org. Manager Configuration), and click Tabs.

The Tab page appears.

3. Click the link for the tab.

The View Tab page appears.

4. Click Modify.

The Modify Tab page appears.

5. In the menu on the far right of the Object Class(es) label, select an auxiliary or template object class or classes to associate with the tab.

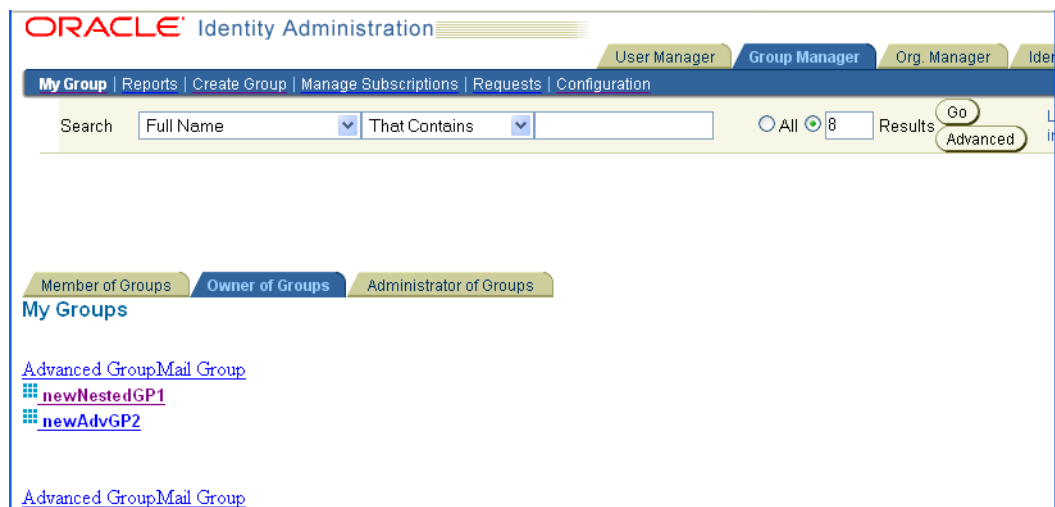
6. Click Save.

The select object class or classes are added to the list on the left of the selection box when you save your changes.

Adding Auxiliary and Template Object Classes to a Group Tab

Use Group Types to associate auxiliary object classes with the Group Manager. Oracle Access Manager provides the oblixAdvancedGroup auxiliary object class to enable you to configure attributes for subscribing members to groups, and for creating and expanding dynamic groups.

The following shows page that appears when you click the Group Manager tab and then click the My Group sub-tab. My Groups has been configured with multiple group type panels.



When you create a Group Type panel, the attributes from the associated object class are available in the Group Manager user application.

To add auxiliary and template object classes to the Group Manager

1. Ensure that you have configured the object class or classes that you want to add in the Common Configuration tab.

See ["Adding Object Classes"](#) on page 3-8 for details.

2. From the Identity System Console, select Group Manager Configuration, then click Configure Group Type.

3. Click Configure Group Type Panels, then click Create.
4. In the topmost menu, select the object class that you want to add.
5. In the Panel Label field, enter the label that you want to display to end users when they view elements from this object class in the Group Manager.
6. Select the Panel Information Is Complete check box.
7. Click Save.

The object class is added. You can view this new object class by clicking the Tabs link in the left navigation pane for Group Manager Configuration.

To delete auxiliary and template object classes from the Group Manager

1. From the Identity System Console, select Group Manager Configuration, then click Configure Group Type.
2. Click Configure Group Type Panels.
3. Click the link for the group type that you want to delete.
4. Click the Delete button.

Configuring Group Manager Tab Options

Use the Group Manager Options feature to select what users see in the My Groups and View Members Profile pages of the Group Manager application. This feature enables you to turn off expensive operations. This can be useful if you need to enhance Identity System performance.

To select what users see in My Groups and View Member Profiles

1. From Identity System Console, click Group Manager Configuration, and click Group Manager Options.

The Group Manager Options page appears.

2. Click Modify to display the Modify Group Manager Options page.

[Table 4–2](#) describes each option.

Table 4–2 Group Manager Options

Option	Description
Show static group	Displays or hides groups consisting of individual members. Applies to the My Groups page.
Show nested groups	Displays or hides groups containing individual members and other groups. Applies to the My Groups page.
Show dynamic groups	Displays or hides groups with members that are determined by a filter. Applies to the My Groups page.
Show groups you are a member of	Displays the Member of Groups attribute on the My Groups page. You must also enable the Show static group, Show nested group, and Show dynamic group options to enable this function.
Show groups you are an owner of	Makes the Owner of Group attribute available on the My Groups page. You must also configure an attribute to be a Group Owner semantic type to use this feature.

Table 4–2 (Cont.) Group Manager Options

Option	Description
Show groups you are an administrator of	Makes the Administrator of Group attribute available on the My Groups page. You must also configure an attribute to be a Group Administrator semantic type to use this feature.
Show static members of this group	Applies to the View Members page. You must configure an attribute to be a Group Static Member semantic type to use the static membership feature.
Show nested members of this group	Applies to the View Members page.
Show dynamic members of this group	Applies to the View Members page. You must configure an attribute to be a Group Dynamic Member semantic type to use the dynamic membership feature.
Allow users to override the defaults through URL parameters	Specifies whether or not the user can enter URL parameters to customize the Group Manager display options. Applies to the View Members page and My Groups page.

3. Select each option you want to apply to Group Manager.
4. Click Save.

Deleting a Tab in Organization Manager

If you have more than one tab in Organization Manager, you can delete a tab.

To delete a tab

1. From the Identity System Console, click Org. Manager Configuration, then click the Tabs link in the left navigation pane.
2. Click the link for the tab that you want to delete.
The View Tab page appears. If you have more than one tab defined for Organization Manager, a Delete button appears on this page.
3. Click Delete.
You are prompted to confirm your decision.
4. Click OK to delete the tab and all associated information.

Ordering the Tabs in Organization Manager

You can change the order in which tabs appear in the Organization Manager when there is more than one tab listed.

To order the tabs in the Organization Manager

1. From the Identity System Console, click Org. Manager Configuration, Tabs.
2. Click the Order Tabs button at the bottom of the list of tabs.
The Order Tabs page appears listing Tab 1, Tab2, and so on. Beside each tab number is a list that contains the names of existing tabs.
3. Use the list beside each tab number to specify the order you would like, for example:

Tab 1: Site

Tab 2: Location

4. Click Save.

Configuring Tab Profile Pages and Panels

A *profile page* is a Web page that shows information about an object in an Identity System application. For example, when you search for information about a user in the User Manager, a profile page for that user is displayed. The profile page may contain data such as the user's:

- Name
- Address
- Department
- Manager
- Phone number
- Email

The information on a profile page is based on objects and attributes in the LDAP directory that the Identity System communicates with, or it can be based on information in an object template file.

You can assemble profile pages from a collection of *panels*. For example, the profile page for a person may contain panels for personal, location, and project information. If you have configured an object template file for provisioning purposes, you may want to place the attributes from the template file on one particular panel.

Users can display profile pages in one of two ways:

- A panel view organizes the data on the profile page into panels.
- A page view organizes the data on the profile page into one long list.

Use of LDAP and Template Objects on a Panel

When you configure LDAP attributes on a panel, the attribute labels and values are shown on the profile pages that use the panel. In contrast, template attributes do not actually appear on the profile page. Template attributes only appear on Modify Profile pages, and then only if you have defined a workflow that uses the attributes.

See ["Sending Non-LDAP Data to External Applications"](#) on page 6-1 for details.

Configuring the Header Panel

The header panel appears at the top of a profile in the User Manager or Organization Manager. The header displays attributes with the semantic types of Full Name, Title, and Photo from the structural object class for the tab. You can turn the header off so that it is hidden from a user identity profile page.

Here is a sample header panel for a user:

User Profile

Title Principal Engineer
Full Name Angela Silver
Photo



Note: You can configure only LDAP attributes from the structural object class for a tab in header panels.

To configure the header panel

1. From the Identity System Console, click User or Org. Manager Configuration, Tabs.
The Configure Tab page appears. The Organization Manager may have multiple tabs.
2. Click a tab link, then click the View Object Profile button.
3. Click Header, which is listed across the top of the page.
The Header Panel page displays the attributes that appear in the Profile header. For example, Map Image, Location Name, Location Title.
4. Click the Modify button, then select each attribute to appear in the header panel.
5. If you want to display the Header Panel in user profiles, click Show Header Panel in User Manager.
6. Click Save.

Viewing Panels That You Have Configured in the End User Application

Panels that you configure in the Identity System Console appear to the user as a collection of attributes on the User Manager, Group Manager, and Organization Manager pages.

The following table shows some examples of panels for a user profile:

Table 4–3 User Profile Panel Attributes

Panel	Attributes
Telecommunications	Telephone number
	Fax number
	Cellular phone number
Location	Room
	Floor number
	Building number
Personal	Organization name
	Type
	Manager

Before configuring a panel, be sure the object class for the attribute that you want to place on the panel is configured with the appropriate object class type. See "[Object Class Types](#)" on page 3-5.

To view a panel in an end user Identity System application

1. From the User, Group, or Organization Manager, conduct a search for a user, group, or organization object.
2. Click one of the links returned on the search.

The profile page for that object appears.

If the application displays the profile in a page view, click the View Panels button.

Adding, Modifying, Localizing, and Deleting a Panel

You can create panels using the attributes configured during setup and when you performed the tasks described in "[Making Schema Data Available to the Identity System](#)" on page 3-1. You can use an attribute once on a panel, and you can usually use the same attribute in more than one panel.

When a user saves new or modified information in an object profile in the Org. Manager application, some of the data may be lost if the Org. Manager tabs do not contain any panels.

To ensure that there is no loss of data when modifying object profiles in Org. Manager, you should configure at least one panel for the tab. This panel should contain the same attributes as the Header Panel for the tab.

In earlier releases, the challenge phrase and response attributes were allowed on different panels of the User Profile page. In 10g (10.1.4.0.1), however, both the challenge phrase and response attributes must be on the same panel. In 10g (10.1.4.0.1), challenge phrases and responses are displayed one after the other even though these are not configured one after the other in the panel.

If a panel contains only the challenge attribute, it will be displayed in the User Profile page without a response. If the panel contains only the response (without the challenge attribute), the response will not be displayed in User Profile Page.

LDAP and Template Attributes on a Panel

You will probably want to configure one or more LDAP attributes or a combination of LDAP and template attributes on a panel. Since template attributes appear only in the context of workflow execution, a panel that consisted only of template attributes would appear to be empty.

Multiple Languages on a Panel

If you have configured the Identity System for more than one language, you can view or modify the panel fields for each language. You can localize display names for the following panel fields:

- Panel Label
- Description
- Attributes
- Mouse Over Message

Note: In the Identity System Console, the display names that appear as values for items in the list of display types (radio button, checkbox, and so on) may be corrupt due to a known limitation with Java Applets and internationalized characters. The browser's JVM displays only those characters that are in the current locale. Internationalized characters are displayed correctly in applets only if you have set the browser to the same locale.

Set the browser to the locale used when setting the display name value.

Creating a Panel

The following procedure describes how to create a panel.

To create or add a panel

1. From in the Identity System Console, click User, Group, or Org. Manager Configuration, then click Tabs.
2. Click the link to the tab.
The View Tab page appears.
3. Click View Object Profile.
4. Click the appropriate button at the top of the page:
 - For the User Manager and Organization Manager, click Configure Panels.
 - For the Group Manager, click Configure Group Profile Panels.
 The Panels page appears. Currently defined panels are displayed.

5. Choose an operation:
 - To add a panel, click Create.
 - To modify a panel, click a panel link and then click Modify.
 - To delete a panel, click a panel link and then click Delete.
 If you selected Create, the Create Panel page appears.

6. Edit the fields.

The Modify Panel page is similar to the Create Panel page. In both pages, the following fields are available:

Label	Description	Label
Panel Label	A name for this panel in the user application. This name can be localized.	Panel Label
Description	Text displayed in the View Panel page. This text can be localized.	Description

Label	Description	Label
Attributes	Attributes selected from the lists. If you need additional attribute fields, click Add at the right side of the page. Note that if you select template attributes, the attribute label will not appear on this panel. Template attributes are only displayed in the context of a workflow. These Attributes can be localized.	Attributes
Title Image	You can view a user profile as a tab-separated page or as a single page. The Title Image is a GIF image that is used for the panel title when viewing a profile as a single page. The GIF must be stored in <i>WebPass_install_dir/identity/oblix/lang/langTag/style0</i> where <i>WebPass_install_dir</i> is the directory where you installed WebPass and langTag is the folder that contains the specific language that you are using. Enter the name of the GIF file, not the path. A Title Image can be modified as described in "Configuring Styles for Identity System Applications" on page 7-2.	Title Image
Tab Image and Tab Image (Bottom)	You can view a user profile as a tab-separated page or as a single page. The Tab Image is a GIF image that is used when viewing a profile as a tab-separated page. The Tab Image usually matches the Panel Label. Until you define a Tab Image, the Panel Label appears as a link on user profile pages. Clicking the link or the Tab Image opens a panel. The (Bottom) version is displayed at the bottom of user profile pages.	Tab Image and Tab Image (Bottom)
Depressed Tab Image	The image used when a user clicks a panel tab in a user profile.	Depressed Tab Image

- When this panel is ready for use, select Panel information is complete at the bottom of the page.
- Click Save.

Note: Checking the box beside "Panel information is complete" saves the panel definition. However, a user's ability to see the contents of a panel is governed by read permissions. The options are described in ["Allowing Users to View and Change LDAP Data"](#) on page 4-22.

Viewing, Modifying, and Localizing a Panel

The following procedures describe how to view, edit, and add localized text to a panel.

To view or modify a panel's configuration

- From the Identity System Console, click User, Group, or Org. Manager Configuration.
- Click the Tabs link in the left navigation pane.

3. Click the link for the tab.
The View Tab page appears.
4. Click the View Object Profile button.
The Profile page appears.
5. Click Configure Panels at the top of the page.
The appropriate Panels page appears. Links for each of the configured panels are displayed on the page.
6. Click a panel link to view its details.
7. Click Modify to display the Modify Panel page.
8. Modify the information as needed.
9. Click Save to save your changes.

To localize a panel

1. From the Identity System Console, click User, Group, or Org. Manager Configuration, then click the Tabs link in the left navigation pane.
The existing tabs appear on the page.
2. Click the link for the tab.
3. Click the View Object Profile button to display the Profile page.
4. Click Configure Panels to display links for each of the configured panels.
5. Click a link to display the View Panel page.
6. Click Translate.
If the Translate button does not appear, you have only one language installed and you cannot localize the panel. Click Modify to edit display names for panel elements in the one language that you have installed.
If you click Translate, the Summary of Panel Display Names page appears. This page displays all configured language-specific display names for the following fields:
 - Panel Label
 - Description
 - Attributes
 - Mouse Over Message
 Display names that have not been configured are marked Not Configured.
7. Click Modify to create or modify a display name.
The Panel Display Names page appears. This page contains fields for the panel display names and links for all the installed languages.
8. Click the language of your choice.
9. Enter the display name in the appropriate field.
10. Click Save to save your changes.

Ordering the Panels

Panels appear in a particular order on a profile page. You can change the order in which they appear in the Group Manager.

To change the order in which panels are displayed

1. From the Identity System Console, click Group Manager Configuration.
2. Click Group Types, then click the Order Group Type Panels at the top of the page.

Note: You can also select User Manager Configuration, Group Manager Configuration, or Organization Manager Configuration then select Tabs, point to link, View Object Profile, and Order Panels. In the Group Manager Configuration, the option at the top of the page is Order Group Profile Panels.

The Order Panels page appears.

3. Use the lists beside each panel number to identify the name of the panel to display.
4. Click Save.

Viewing Group Type Panels

Group Type panels allow you to organize attributes on the My Groups tab. For example, if you have configured `groupOfUniqueNames` as a structural object class and `oblixAdvancedGroup` as an auxiliary class, you can organize attributes from these classes on the My Groups tab by creating Group Type panels.

Note that Group Type panels are reserved for LDAP attributes. You should not configure template attributes on a Group Type panel.

Each object class identified as a Group Type (as described in ["Object Class Types"](#) on page 3-5 in the Identity System) can be associated with a Group Type panel.

To view Group Type panels

1. In the Identity System Console, click Group Manager Configuration, Group Types, then the Configure Group Type Panels link.

The Panels page displays a list of configured Group Type panels.

2. Click its link to view a Group Type's settings.

The View Panel page appears showing the settings for the selected panel.

The screenshot shows the Oracle Identity Administration console. The top navigation bar includes links for Provisioning, Help, and About. Below this, a breadcrumb trail shows the path: System Configuration | User Manager Configuration | **Group Manager Configuration** | Org Manager Configuration | Common Configuration. The left sidebar contains a tree view with categories like Tabs, Reports, Group Types (selected), Group Manager Options, Audit Policies, and Group Cache. The main content area is titled 'Configure Group Type Panels' and 'Order Group Type Panels'. Under 'View Panel', the following details are shown:

View Panel	
Panel Label	Adv Group
Associated ObjectClass	oblixadvancedgroup
Description	
Title Image	(no image)
Tab Image	(no image)
Depressed Tab Image	(no image)
Tab Image (Bottom)	(no image)
Mouse Over Message	

At the bottom of the configuration area, there are three buttons: Modify, Delete, and Back.

Adding, Modifying, Localizing, and Deleting a Group Type Panel

You must configure a Group Type panel to organize the attributes for a group object class. At least one panel should be created for the group structural object class. This enables you to view groups that contain only the group structural object class attributes on the My Groups profile page.

If you have installed and configured multiple languages, you can localize display names for the following panel fields:

- Panel Label
- Description
- Mouse Over Message

To add, modify, or delete a Group Type panel

1. From the Identity System Console, click Group Manager Configuration, Group Types.

The Group Types page displays a list of Group Types.

2. Click Configure Group Type Panels to display the Panels page.

3. Choose an operation:

- To add a Group Type panel, click Create.
- To modify an existing panel, click a panel link and from the View Panel page click Modify.
- To delete an existing panel, click a panel link and from the View Panel page click Delete.

4. In the field labeled Select the Group Type, select the object class to associate with the Group Type.

Note: Select only auxiliary object classes that extend the group structural object class or are attached to the group structural object class in the schema. Only configured auxiliary classes can be selected from this page. For more information, see ["Adding Object Classes"](#) on page 3-8.

5. In the remaining fields, enter values as described in ["To create or add a panel"](#) on page 4-15.
6. Select the box beside Panel information is complete.
7. Click Save.

Note: Selecting Tab information is complete saves the panel definition, but a user's ability to see the contents of a panel is governed by read permissions, as described in ["Allowing Users to View and Change LDAP Data"](#) on page 4-22.

To localize panel display names

1. In the Identity System Console, click Group Manager Configuration.
2. Click Configure Group Types in the left navigation pane.
3. Click Configure Group Type Panels.S
4. The Panels page displays a list of configured Group Type panels.
5. Click the panel for which you want to configure display names.

The View Panel page appears.

6. Click Translate.

This button only appears if you have more than one language installed. If you only have one language installed, click Modify to configure display names for panels elements.

The Summary of Panel Display Names page appears. This page lists all the configured display names for the following fields:

- Panel Label
- Description
- Mouse Over Message

Display names that have not been configured for a particular language are marked as Not Configured.

7. Click Modify.

The Panel Display Names page appears.

8. Click the language for which you want to configure display names.
9. Enter the display names for the panel fields.
10. Click Save to save your changes.

Modifying and Localizing Attributes Displayed on a Panel

The attributes you configure through Common Configuration pages are used in each application using that object class. For instance, through common configuration you can set the display name for the cn attribute to be Full Name. This is what appears on a user Profile page. If you then configure the cn attribute to display as Legal Name from the User Manager configuration screen, it is displayed by default as Legal Name on the user Profile page. See ["Making Schema Data Available to the Identity System"](#) on page 3-1 for details.

You can also localize display names of attributes that are displayed on a panel. This enables you to present attributes in the user's native language. See ["Configuring Multiple Languages for Oracle Access Manager"](#) on page 7-7 for information on managing multiple languages.

Note: The only way to change the display type or semantic type of an attribute once it has been assigned to a panel is to delete and then re-create the panel.

However, as described in the following paragraphs, you can override the information configured for the attribute at the object class level.

Each Identity System application (User, Group, and Org. Manager), provides an application-specific Configuration tab with a Tabs function. Using the application-specific Tabs function, you can provide a different display name or display type for an attribute on that application-specific Configuration tab only (different than what is configured at the object class level). For example, you may have a different display name for the "description" attribute on the User Manager Configuration tab.

For proper localization when you have more than one language installed, when you reconfigure an attribute at the tab level you must provide display names for that attribute in all installed languages. For example, suppose you have two installed languages. To provide translations for the "description" attribute on the User Manager Configuration tab (for example) in both languages, you must specify the display name for the attribute in the installed languages at the same tab level. The Translate button appears only when more than one language is installed.

The following procedures illustrate how to reconfigure an attribute at the tab level to override the information configured for the attribute at the object class level.

To modify attributes specific to the User, Group, or Organization Manager

1. From the Identity System Console, click the User, Group, or Organization Manager Configuration tab.
2. Click the Tabs link in the left navigation pane.
The Tab page appears. There may be multiple tabs for the Organization Manager.
3. Click the link for the tab.
The View Tab page appears.
4. Click Modify Attributes.
The Modify Attributes page appears.

Details on modifying an attribute are provided in ["Configuring Attributes"](#) on page 3-17. You can localize attribute display names as described next.

To localize attribute display names

1. From the Identity System Console, click User, Group, or Organization Manager Configuration.
2. Click the Tabs link in the left navigation pane.
The Configure Tab page appears. There may be multiple tabs for the Organization Manager.
3. Click the link for the tab.
The View Tab page appears.
4. Click Translate.

Note: The Translate button appears only if more than one language has been installed.

The Summary of Attribute Display Names page appears. This page lists all configured attribute display names for all languages. Display names that have not been configured are marked Not Configured.

5. Click Modify.
The Attribute Display Names page appears. This page lists display name fields for attributes and links for the installed languages.
6. Click the language for which you want to configure display names.
7. Enter the name in the Display Name field.
8. Click Save to save your changes.

Allowing Users to View and Change LDAP Data

You can think of configuring objects and attributes and assembling attributes into panels on application tabs as being like playing with building blocks. Once you have arranged your building blocks, you can determine who is allowed to play with them.

You must configure the Identity System to allow people to search for and view the LDAP attributes you have configured on the application panels. To do this, you:

- Determine the level of the directory tree that users are permitted to search.
- Set View and Modify permissions for specific attributes in the directory tree.

The following section discusses setting the searchbase as a method of configuring view and modify permissions. The searchbase refers to searching the LDAP directory tree. Template attributes are not relevant to setting a searchbase. To give users the ability to enter values for template attributes, the users must be participants in a workflow where these attributes are used. See ["Chaining Identity Functions Into Workflows"](#) on page 5-1 for details.

This section contains the following topics:

- [About the Searchbase](#)
- [Guidelines for Setting the Searchbase](#)
- [Indexing and the Searchbase](#)
- [Setting the Searchbase](#)

- [Configuring and Deleting Disjoint Searchbases](#)
- [Writing LDAP Filters Using Query Builder](#)
- [Building Advanced LDAP Filters Using QueryBuilder](#)
- [About View and Modify Permissions](#)
- [Setting and Modifying LDAP Attribute Permissions](#)
- [Keys for Selecting Multiple Attributes](#)
- [Evaluation of LDAP Attribute Permissions](#)

Note: You can modify the default processing of large static groups if these groups adversely affect performance. See the chapter on performance tuning in the *Oracle Access Manager Deployment Guide* for details. Note that if you do this, subgroups in this group are not considered when the searchbase or a filter is set for the parent group. Set the searchbase or filter for the subgroup directly if you modify the evaluation of the static group that it belongs to.

About the Searchbase

A searchbase is a branch in the directory tree, or it can be the top node of the tree. At installation time, you select the default searchbase. The default searchbase is the node in the directory tree under which all user data is stored and the highest possible base for all user data searches. The searchbase determines the part of the directory tree that is available to a user during a search. You must set a searchbase for each structural object class configured for the Identity System before a user can view its entries. You can set multiple searchbases for each structural object class.

When you set a searchbase, you determine who can search what (an object class, at a particular level of the directory tree), optionally using a search filter.

Before setting a searchbase you need to determine the following:

- What object class (users or groups) do I want to set the searchbase for?
- Where will the search begin?
- Who can search there?

For example, you can configure one searchbase for employees and another for customers to ensure that customers cannot see employee information.

As another example, if two competing suppliers provide you with parts, you can set the searchbase so that users from each supplier can view only their own portion of the DIT.

Note: You set the searchbase from the User Manager application. This is the end user application rather than the User Manager Configuration function. You also need to configure read permissions for your group profile pages for the group class.

Guidelines for Setting the Searchbase

When you set a searchbase, you have the option to define a filter to identify what branch of a searchbase a logged-in user can view. If your directory tree is particularly flat, so that selecting a node does little to filter the searchbase, the filter feature helps

narrow searches. The filter is also useful if your directory tree has a large number of branches, for instance, if you have 10,000 dealerships, you probably want to narrow down searches within the dealerships.

However, a filter can affect performance if it yields a large number of entries. Instead of using a searchbase filter, you can set read permissions for the class attribute, as described in ["Selecting a Class Attribute"](#) on page 3-7. The class attribute is used for attribute access and to link search results to a Profile page.

For example, suppose you remove the resource filter from the searchbase, allowing the role of Anyone to access the person object class. Instead of setting the searchbase, you define read permissions for the class attribute, using a rule to specify who can access this attribute. This can reduce the number of directory searches that the Identity System conducts. See ["Setting and Modifying LDAP Attribute Permissions"](#) on page 4-33 for details.

Note: You can set several searchbases for the same user or group if specific users need to access different parts of the directory tree. For example, if employees need to search both the employee and the customer branches of the tree, you can define searchbases for employees and for customers, and give employees permission to view both. However, be sparing when configuring multiple searchbases for a particular object class. Where possible, define read and write permissions for attributes instead. Multiple searchbases for the same object class can degrade performance.

If You Need to Modify a Searchbase

If you change the levels being searched in the directory tree or if you change the search attribute, you cannot directly modify a searchbase. If you attempt to do so, the Identity System treats the modified searchbase as a newly defined searchbase. The only way to modify a searchbase is to delete it and create a new one.

Note: You can modify a searchbase if the changes are other than those described in this section.

Indexing and the Searchbase

Searches of the directory are a significant factor in system performance. See the *Oracle Access Manager Deployment Guide* for guidelines on indexing.

Indexing Requirements for Oracle Internet Directory

Oracle Internet Directory returns an error when un-indexed attributes are used in a search. For example, suppose that you define a derived attribute using an un-indexed "Match Attribute" and add the attribute to a profile page in Oracle Access Manager. When the page is displayed, Oracle Internet Directory returns an error and no values are displayed for the derived attribute in the profile page. In the Oracle Access Manager log file, an error message "Operation not supported" is logged.

To use additional attributes in search filters, you must add them to the catalog entry. You can index only those attributes that have:

- An equality matching rule
- Matching rules supported by Oracle Internet Directory as listed in "About LDAP Attribute Matching Rules" in *Oracle Identity Management User Reference*.

- No more than 128 characters in their names.

You can index a new attribute—that is, one for which no data exists in the directory—using the `ldapmodify` tool. You can index an attribute for which data already exists in the directory by using the Catalog Management tool. You can drop an index from an attribute by using `ldapmodify`, but Oracle recommends that you use the Catalog Management tool.

Once you have defined a new attribute in the schema, you can add it to the catalog entry by using `ldapmodify`.

To add an attribute for which no directory data exists, import an LDIF file by using `ldapmodify`. For example, to add a new attribute `foo` that has already been defined in the schema, import the following LDIF file to Oracle Internet Directory using `ldapmodify`:

```
dn: cn=catalogs
changetype: modify
add: orclindexedattribute
orclindexedattribute: foo
```

Do not use this method to index an attribute for which data exists in the directory. To index such an attribute, use the Catalog Management tool.

Setting the Searchbase

The following procedure describes the steps for setting the searchbase.

To set the searchbase

1. In the User Manager application, click the Configuration sub-tab.

The Configuration page appears.

2. Click Set Searchbase.

On some browsers you may receive a prompt asking if you trust the certificate of the application. If this happens, select the Trust Always option.

The Set Searchbase page appears.

ORACLE Identity Administration

User Manager Group Manager Org. Manager Id

My Profile | Reports | Create User Identity | Deactivate User Identity | Substitute Rights | Requests | **Configuration**

Search Full Name That Contains All 8 Results Go Advance

Attribute Access Control Delegated Administration Workflow Definition **Set Searchbase**

Set Searchbase

Set the searchbase for a particular organization or person. This will localize access to ensure security

1) Objectclass gensiteorgperson

2) Searchbase Domain

o=company,c=us

Filters (ou=\$ou\$)

Add Filter (ou=\$ou\$)

3) Target Domain

o=company,c=us

Role ☐ Anonymous ☒ Anyone

Build Filter

Rule

3. In the Object Class list, select an object class.

The object class you select defines what is being searched. For instance, to set a searchbase for widgets, you would select the widget object class.

The Searchbase Domain box indicates the top node for the search. The field beneath the Searchbase Domain box is where you enter or edit information.

4. In the field under the Searchbase Domain box, specify the part of the directory tree where the search for the object may be conducted.

For instance, if you want to define a searchbase for widgets in the Manufacturing Department, you might select the Manufacturing branch of the searchbase.

Selecting the top level of the directory tree indicates that the entire domain is available for searches. You can refine the searchbase by selecting a node further down the tree or by entering a filter. For example, to restrict searches to North America, you could select the top node and enter region=North America as the filter. This example assumes there is a branch called North America in your directory tree. See ["Usage of Rules and Filters"](#) on page 3-23 for details on writing a filter.

The Filters box indicates the current filters for the search. You use the Add Filter field, beneath the Filters box, to enter another filter.

5. **Optional:** In the Add Filters field, enter another filter.
6. Click Save.

The new filter appears in a field under the previous filter.

Users and groups permitted to search this portion of the directory tree are defined in the next panel.

7. Specify the user or group that is permitted to search this portion of the directory tree.

For Example:

- **Target Domain:** Any user object in the tree under the node you select.

Do not use full LDAP URL while specifying the filter for target domain (or workflow domain) while creating the workflow. Only the LDAP filter is expected. For example, cn=Shutterbug Canavan is expected rather than ldap:///ou=Partners,o=Company,c=US??sub?(cn=Shutterbug Canavan).

- **Role:** The role of the users.

If you want to give this right to everyone whether they have logged in or not, select Anonymous.

If you want to give this right to anyone who has logged in to the User Manager, Group Manager, or Organization Manager, select Anyone.

Note: Anonymous access is used only in the Self Registration function in User Manager and Organization Manager. Also, anonymous access applies only to display type attributes (a check box, radio button, or list) that are configured as a Rule. For example, suppose you configure the ou attribute as a list display type with a rule that uses the LDAP filter (objectclass=organizationalunit). To configure this attribute for self registration, you would access the Organization Manager tab for organizationalUnit, configure attribute access for the class attribute (as described in ["Setting and Modifying LDAP Attribute Permissions"](#) on page 4-33, and grant Anonymous access.

- **Rule:** Any person you specify with an LDAP filter. Click Build Filter and use the Query Builder to create a rule. See ["Writing LDAP Filters Using Query Builder"](#) on page 4-29 for details.
- **Person(s):** Any person you select. Click Select User and use the Selector to choose individuals.
- **Group(s):** Any group you select. Click Select Group and use the Selector to choose one or more groups.

To copy users and groups from one searchbase to another, click Copy, click Reset, select another Searchbase Domain and Target Domain, and click Paste. The users and groups appear in their respective boxes.

Note: If you specify users by more than one means (for instance, by a rule and by selecting individual users), both methods apply. The only exception is when Anyone is selected. Anyone supersedes all other methods.

8. Click one of these buttons to take the appropriate action:

- **Save:** Save and implement changes.
- **Reset:** Clear all selections.

- **Delete:** Clear all rule, group, and user specifications.
- **Report:** Generate a report summarizing the configured searchbases.

If You Set a Searchbase for a Group

You can set the searchbase for the groupOfUniqueNames object class and select the groups for which you are defining the searchbase. Before people in the group can view entries in a searchbase for a group, you need to configure read permissions for your group profile pages for the group class, as described in ["Setting and Modifying LDAP Attribute Permissions"](#) on page 4-33.

Configuring and Deleting Disjoint Searchbases

A disjoint searchbase is a searchbase that supplements the one you selected when you set up the Identity System. You create a disjoint searchbase to identify an additional LDAP directory tree under which user data can exist.

You can add multiple disjoint searchbases to a domain. The following procedures describe how to add and delete a disjoint searchbase.

For more information on managing disjoint searchbases, see ["Working With Multiple Directory Searchbases"](#) on page 7-34.

To add a disjoint searchbase for a disjoint domain

1. From the Identity System Console, click System Configuration, Directory Profiles.
2. Click the Directory Server link.
3. Add a disjoint searchbase in the Disjoint Search Base field and click Save.
4. From the Identity System Console, click User Manager Configuration.
5. From the left navigation pane, select Tabs.
The Configure Tab page appears.
6. Select the tab link.
7. Click Modify.
8. Make sure there is no value in the Tab Search base field.
9. Save your changes, if necessary.

To delete a disjoint searchbase

1. Disable all directory profiles that use this searchbase.

You can view the searchbases that are configured for a directory profile in the Name Space field for the profile. There will be one directory profile for each searchbase in the disjoint domain. See ["Creating an LDAP Directory Server Profile"](#) and ["Creating an LDAP Directory Server Profile"](#) on page 7-34 for details.

2. Remove all access control policies for the disjoint searchbase.

If there are policies defined for the deleted searchbase, a user who has this searchbase on this node will be able to create a filter using Query Builder whose base is this searchbase.

3. From the Identity System Console, select System Configuration, then Directory Profiles.
4. Click the Directory Server link.

5. Remove the information in the Disjoint_domain field, then click Save.

Writing LDAP Filters Using Query Builder

The Query Builder enables you to write LDAP filters when you perform activities such as setting the searchbase.

The Identity System enforces a limit of 20 hits for a query. This applies to both Selector and Query Builder. If you perform a search or query that results in more than 20 hits, you receive truncated results. For instructions on changing the search limit, refer to the cookieBustLimit parameter in the *Oracle Access Manager Customization Guide*.

You access the Query Builder function from the Build Filter button. For example, this function is available when setting a searchbase. See ["Setting the Searchbase"](#) on page 4-25 for details.

Note: If you choose the Is Present or Is Not Present operator when building a query, the value specified for the display type is not taken into consideration, since the filter that is used is a presence filter.

Note: You can modify the default processing of large static groups if these groups adversely affect performance. See the chapter on performance tuning in the *Oracle Access Manager Deployment Guide* for details. Note that if you do this, members of any subgroup that is nested within the large static group are not considered when the filter is set for the parent group. Set the filter for the subgroup directly if you modify the default evaluation of the parent group.

To use the Query Builder

1. Click the User Manager application tab.
These are the Identity System applications.
2. Click the Configuration sub-tab.
3. Click Set Searchbase.
4. From the Set Searchbase page, locate and click the Build Filter button.
The Query Builder page appears. By default, the Basic query page is displayed.
5. In the Attribute list, select an attribute you want to use as search criteria.
For example:
Admin
6. Click Add.
The attribute is added to the filter.
7. From the list beside the new attribute, select a matching method.
For example:
greater than or equals
The available methods depend on the attribute. See ["Methods for Retrieving Matches"](#) on page 4-30 for details.
8. In the field beside the method, select or type the query string.

For example:

January 22 2003

9. Click Add to add other attributes.
10. From the list to the left of the attribute, select the relationship between attributes:
 - And: Results must match criteria in all rows.
 - Or: Results can match criteria in one row.

For example, you can search for everyone with the Administrator attribute and a start date after (greater than) January 22, 2003.
11. Click Test to test your filter.

If too many or too few results are received, make your criteria more or less restrictive.
12. Click Delete next to an attribute to remove it from the filter (or click Delete All to delete all attributes).
13. Click Save.

When you Save, the filter appears in the previous page under the Build Filter button.

Note: If you receive a Bad Request message when you save, your query string is too long for your browser. Browsers handle the filters as URLs, and they generate an error if the query string exceeds their maximum URL length.

Methods for Retrieving Matches

The matching methods that you can select in the Query Builder depend on the display type of the attribute. For instance, the display type of an attribute may be a list or a set of radio buttons. See ["Attribute Display Types"](#) on page 3-14 for details. When you use the Query Builder to create a filter for an attribute with a display type that contains multiple values, for example, a list, the query returns a match even if only one value satisfies the filter.

When building a filter, you can select multiple values for an attribute in one row only if the attribute display type is a check box or a radio button.

[Table 4-4](#) lists the matching methods that the Query Builder uses:

Table 4-4 Matching Methods in the Query Builder

Method	Description
equals	Results are an exact match of the value.
does not equal	Results do not include the specified value.
less than or equals	Results are less than or equal to the specified value. For example, specifying k for a full name query returns people whose name begins with a letter from A to K.
greater than or equals	Results are greater than or equal to the specified value. For example, specifying k for a full name query returns people whose name begins with a letter from K to Z.

Table 4–4 (Cont.) Matching Methods in the Query Builder

Method	Description
less than	Returns any directory entry with a value that is less than the specified value. When filtering a text string, a value of less than returns entries that precede the specified value alphabetically. For example, specifying k for a full name query returns people whose name begins with the letters from A to J.
greater than	Returns any directory entry with a value that is greater than the specified value. When filtering a text string, a value of less than returns entries that follow the specified value alphabetically. For example, specifying k for a full name query returns people whose name begins with the letters from L to Z.
contains	Returns any directory entry that contains the specified string anywhere in the value of the entry. For example, an entry of st might return values of street or best.
does not contain	Returns any directory entry that does not contain the specified string anywhere in the value of the entry.
is present	Returns any directory entry that contains this attribute. For instance, if you select the Administrator attribute and the is present method, all administrators are returned.
is not present	Returns any directory entry that does not contain this attribute.
begins with	Returns any directory entry that begins with the specified value.
ends with	Returns any directory entry that ends with the specified value.
does not begin with	Returns any directory entry that does not begin with the specified value.
does not end with	Returns any directory entry that does not end with the specified value.
sounds like	Results approximate the sound of the specified value. Use this option if you are unsure of the spelling of your desired search object. Use phonetic spelling. For example, specifying kiero might return values for cairo. This option is not supported by Novell Directory Services.
does not sound like	Results display entries that do not approximate the sound of the specified value. Use your best phonetic spelling. This option is not supported by Novell Directory Services.

Building Advanced LDAP Filters Using QueryBuilder

Filters can work on multiple attributes and use logical operators such as And, Or, and Not.

To build a complex filter

1. From the Identity System Console, click the tab for the User Manager application.
2. Click the Configuration sub-tab.
3. Click Set Searchbase.
4. Click the Build Filter button.
5. In the Query Builder page, click the Advanced tab.
6. If you switch from Basic to Advanced, and you choose OK, you lose the current filter (click Cancel to keep the displayed filter).

The Advanced page appears.

If the Advanced page does not appear after you click the Advanced tab, the URL could be too long. The length of the URL is determined by the browser.

7. In the Select Attribute list, select the attribute you want to use as the search criteria.
8. In the associated list select a matching method, and in the associated text entry field add a query string.
See ["Writing LDAP Filters Using Query Builder"](#) on page 4-29 for details.
9. Click Add.

The attribute is added to the Constructed Visual Filter box.

10. You can perform the following optional steps:
 - To add to your LDAP commands, use the And, Or, or () buttons.
 - To remove an attribute from the Constructed Visual Filter box, select it, and click Delete (or Delete All to remove all attributes).
 - To modify an entry in the Constructed Visual Filter box:
 - Select the entry.
 - Make your changes to the query characteristics at the top of the page.
 - Click Modify.

11. Click Show LDAP Filter to view the filter you are building.

The LDAP string displays in the LDAP Filter box. You can edit the text in this box and click Update Visual Filter. For examples of LDAP filters, see ["Static LDAP Search Filters"](#) on page 3-24 and ["Examples of Dynamic LDAP Search Filters"](#) on page 3-25.

If you manually enter a very complex filter, the Constructed Visual Filter box may not be able to interpret it correctly. However, the filter will work correctly.

12. Click Test to view the results of your query.

the Identity System displays output that conforms to your filter.

13. Click Save to save and apply your filter.

If you receive a "Bad request" message when you click Save, your query string is too long for your browser. Browsers handle the filters as URLs, and they generate an error if the query string exceeds their maximum URL length.

About View and Modify Permissions

Until you configure permissions for an attribute, no users can see attributes displayed in the User Manager, Group Manager, and Organization Manager. For example you can allow all users to view employee work phone numbers in the User Manager, but allow only managers to view home phone numbers.

If you are a Master Identity Administrator or a delegated administrator with appropriate permissions, you can configure user permissions. By default, Master Administrators specified during Identity Server installation have full access to all attributes. You can change the default by setting the `BypassAccessControlForDirAdmin` parameter to false in:

`IdentityServer_install_dir/identity/oblix/apps/common/bin`

Setting and Modifying LDAP Attribute Permissions

The Attribute Access Control function lets you specify permissions that determine who can read and modify the values for each LDAP attribute. It also lets you create a list of users or groups to be notified when an attribute is changed. As with setting the searchbase, this functionality only applies to LDAP attributes. You configure permissions for template objects when you add participants to workflow steps. See ["Chaining Identity Functions Into Workflows"](#) on page 5-1 for details.

Users must have a searchbase defined as well as read permissions to be able to view an attribute. For instance, to be able to view the class attribute on the User Manager, Group Manager, or Organization Manager tab, a user must be a trustee of the appropriate searchbase domain for the class attribute, and they must have read permissions for this attribute.

To set or modify attribute permissions

1. In User, Group, or Organization Manager, click Configuration at the top of the page.

The Configuration page appears.

2. Click Attribute Access Control.

If you receive a prompt asking if you trust the certificate of the application, select the Trust Always option.

The Attribute Access Control page appears.

3. In the Management Domain box, specify the scope of the Directory Information Tree (DIT) that this permission applies to.

Initially, this field displays the searchbase set during product setup. This searchbase can only be changed by performing setup again. Selecting a lower level in the tree applies access control for that branch. For example, if you select the Full Name attribute, and then select a lower level department such as Sales, you are applying access control to all of the people in Sales with Full Name in their profile.

4. Optional—Use the Filters field to enter an LDAP rule to specify the objects and attributes more precisely.

A filter refines the attributes you are allowed to read or modify. If you do not use a filter, the Identity System uses objectclass=.

Note: A filter is useful if your database design is particularly flat or has a particularly large number of branches.

Add the Filter in the Add Filter field. Once the configuration is saved, the filter is added to the Filters list. If you later want a different filter, you must delete the original searchbase and create a new configuration.

For more information on filters, see ["Usage of Rules and Filters"](#) on page 3-23.

5. Specify the Right:
 - Read—Selected users can view the attribute and its value on a profile page.
 - Modify—Selected users can change the attribute value. Note that you must confer read permissions for these users to be able to see the attribute value.
 - Notify—Sends an email to the specified users when an attribute value is changed.

For example, you can give read and modify permissions to the Title attribute for a manager. Then you can set notification to be sent to the HR department when the value for this attribute is modified in a user profile. For details about email post-notification for a self registration step, see ["Descriptions of Step Actions"](#) on page 5-13.

6. In the Attribute box, select the attribute to associate with this right.

If you want to make multiple selections, see ["Keys for Selecting Multiple Attributes"](#) on page 4-35.

Note: If an attribute in your multi-select range has a different set of trustees, an error appears. This prevents you from inadvertently allowing access to incorrect trustees (participants).

7. Confer this right to one or more of the following:

Role: Assigns the right based on the user's role. Any attribute with a data type of DN and a display type of Object Selector appears in the Role area. Self and Anonymous are shipped with the Identity System. Each application contains different roles, largely dependent on your configured attributes. For example, the User Manager may have the Manager role, but not any role based on the secretary attribute, depending on your configuration. Common roles include the following:

Role	Description
Anyone	All users who log in to the User, Group, or Organization Manager can either view or modify the attribute at the selected level. For example, all logged-in users can view the phone number attribute at the specified level in the directory.
Anonymous	All users can view entries, whether they are logged in or not. Anonymous access is only used for self-registration.
Self	The user logged into the User Manager application can view or modify the attribute for his or her own identity, assuming the read and write permissions for attributes is high enough on the directory tree to include the user's profile. For example, if you select Self to be able to view the Name attribute at the top level, then you, as a person logged into the User, Group, or Organization Manager, are able to view your name. But if you specify ou=Marketing as the level on the directory tree, and the user is not in Marketing, then you cannot view your name.
Manager	The user logged in to the User Manager application can either view or modify the attribute for their direct reports.
Secretary	If the user logged in to the User Manager is an administrative assistant, he or she can view or modify the attribute for the people they support.
Group Owner	The user logged in to the Group Manager can view or modify the attribute for the group that he or she is an owner of.
Group Administrator	The user logged in to the Group Manager can view or modify the attribute for the group that he or she administers.
Group Member	The user logged in to the Group Manager can view or modify the attribute for the group that he or she is a member of.

Rule: Click Build Filter and use the Query Builder to create a rule. See ["Writing LDAP Filters Using Query Builder"](#) on page 4-29 for details.

Person(s): Click Select User and use the Selector to specify one or more users.

Group(s): Click Select Group and use the Selector to specify one or more groups.

See "[Evaluation of LDAP Attribute Permissions](#)" on page 4-36 for information on the order for evaluating permissions.

8. Click Copy, click Reset, select a new attribute, and click Paste to copy users and groups from one attribute to another.
9. Click one of these buttons:
 - **Save:** Save and implement your changes.
 - **Reset:** Clear all selections.
 - **Delete:** Clear all rule, role, group, and user specifications.
 - **Report:** Generate a report of attributes and their access permissions in the domain.

Keys for Selecting Multiple Attributes

You can configure access control for multiple attributes at one time using the following keyboard combinations:

- Ctrl + Home: Selects all attributes above and including the highlighted attribute.
- Ctrl + End: Selects all attributes following and including the highlighted attribute.
- Ctrl + Page Up: Selects only attributes above the highlighted attribute.
- Ctrl + Page Down: Selects only attributes following the highlighted attribute.

Note: If an attribute in your multi-select range has a different set of trustees (participants), you receive an error. This prevents you from granting access to incorrect trustees.

Platform-specific key combinations are as follows:

Browser Type	Function
Windows Browsers	■ To select multiple attributes, hold down the Ctrl key and select the attributes. ■ To select an attribute and all attributes before it, hold down the Ctrl+Shift+Home keys and select the attribute. ■ To select an attribute and all attributes following it, hold down the Ctrl+Shift+End keys and select the attribute. ■ To select an attribute and an arbitrary number of attributes following it, select it and press Shift+Down Arrow. ■ To select an attribute and an arbitrary number of attributes before it, select it and press Shift+Up Arrow.
Unix Browsers	■ To select multiple attributes, hold down the ESC key and select the attributes. ■ To select an attribute and all attributes before it, hold down the ESC+Shift+Home keys and select the attribute. ■ To select an attribute and all attributes following it, hold down the ESC+Shift+End keys and select the attribute.

Evaluation of LDAP Attribute Permissions

When you assign multiple methods for view and modify permissions, the Identity System evaluates the methods in this order:

1. Users
2. Roles
3. Groups
4. Rules (LDAP filters)

When the Identity System finds a match, it stops checking. For example, suppose you grant read permission for the Name attribute for User=Lou Reed, but you also have a rule that says (&(!(cn=Lou Reed)) objectclass=person object class), which allows everyone except Lou Reed. Lou Reed has access because he is a User, which precedes Rule in the evaluation order. As another example, if you specified a rule denying access to the Human Resources department but used the people selector to specify an individual employee in Human Resources, the union of the rule and people categories would allow access to the specified employee.

Note: If you select the Anyone role, all users, roles, groups, and filters are superseded.

Examples of Configuring an Application

The following sections describe different scenarios for configuring an application. Separate examples are provided for the User Manager, Group Manager, and Organization Manager.

Displaying Photos in User Profiles

Photos appear in the header panel of a user profile. Users with self-service permissions on relevant attributes can manage their own photos.

There are two ways you can store photos in the Identity System:

- In an LDAP directory
- Referencing photos in a file system.

You cannot use both methods. All of your photos must be stored in either a directory or a file system.

Importing and Storing Photos in a Directory

When you want to store your photos or other images in a directory, place the photos on the Identity Server and use the Identity System to import the photos into the directory and to configure an attribute to be the photo attribute. You can create your own attribute, or you can use an attribute that already exists. This attribute must be defined as a binary type in your directory, and in the Identity System the attribute must be defined as a Photo semantic type with a GIF display type. The GIF display type supports GIF and JPEG formats, and other image file formats that are supported by your Web server.

Before associating a photo with a user's identity, be sure the photo's file name is based on the value of the attribute with the Login semantic type. For example, if your Login semantic type is assigned to the uid attribute, you would use the following file name conventions:

```
attribute_value_of_uid.gif  
or  
attribute_value_of_uid.jpg  
or  
attribute_value_of_uid.jpeg
```

If your login semantic type is something other than uid, use that instead for your file name. For example, if the Login semantic type is assigned to the email attribute, your photo file names must be the following:

```
attribute_value_of_mail.gif  
or  
attribute_value_of_mail.jpg  
or  
attribute_value_of_mail.jpeg
```

The file extension must be compatible with a graphic file format that your Web servers can support.

When the Identity System imports photos and images, it converts the files into Base64 format. This data becomes the value of the Photo attribute. the Identity System uses the Login attribute and the photo or image file name to determine which photo belongs to which user entry.

Steps for configuring the Identity System to use the photos are described in the following discussions.

To configure photos for importing to a directory

1. From the Identity System Console, click Common Configuration, then click Object Classes.
2. Select your person object class from the list.
3. Click Modify Attributes.
4. Modify the Photo attribute as follows:
 - **Attribute:** Photo
 - **DisplayName:** Photo
 - **Semantic Type:** Photo
 - **Data Type:** Binary
 - **Attribute Value:** This is always a single value attribute
 - **Display Type:** GIF Image
5. Save your changes.
6. In the User Manager, under Attribute Access Controls, assign Read and Write permissions to this attribute.

To import photos to the directory

1. From the Identity System Console, click System Configuration, and click Photos.
2. Specify the path to the photos stored on the Identity Server.
3. Click Save.

This imports all of the GIF and JPEG images into your directory.

Referencing Photos in a File System

Another method for storing images and photos for user identities is to store the photos in a location other than the directory. This method is appropriate for GIF and JPEG images, and other image file formats that are supported by your Web server.

The Identity Server's WebPass must be able to access this location. You can name the photo or image file using any valid file name that the Web server recognizes and supports. Avoid using special characters such as spaces in the file name. The Web server may not recognize file names that use special characters.

To reference photos that reside in a file system

1. In the Identity System Console, click Common Configuration, Object Classes.
2. Click the person object class in the Object Class list.
3. Click Modify Attributes.
4. Modify the Photo Path attribute as follows:

- Attribute: Photo Path
- DisplayName: Photo
- Semantic Type: Photo
- Data Type: String (case-sensitive)
- Attribute Value(s): Single or multi-valued
- Display Type: GIF image URL

5. Assign read and write permissions for this attribute.
6. Store the images in GIF or JPEG format in the following directory:

WebPass_install_dir/identity/oblix/lan/langTag/style0

where *WebPass_install_dir* is the directory where WebPass is installed and *langTag* is the folder that contains the specific language you are using.

7. Enter the photo location URL in the User Profile Page for each user.

For example, if the image location is:

```
c:\COREId\WebComponent\identity\oblix\apps\lang\en-us\style0\user1.gif
```

you set the photo location to:

```
user1.gif
```

More than one GIF image can be displayed by setting the photo URL attribute to be multi-valued.

The Default Photo Image

the Identity System supplies a default photo image. This image is presented in case there is no photo image supplied for a user. The image is stored in CIMAGEdefaultphoto.gif in style0 on the Identity Server.

Enabling the Location Tab in Organization Manager

the Identity System provides a Location tab by default in the Organization Manager. This tab enables you to create maps and associate users or objects with locations on those maps.

Task overview: Enabling Location functionality

1. The Master Identity Administrator modifies the Location tab and adds location attributes to Profile pages for the User and Organization Manager applications.
2. The Master Identity Administrator configures access controls for location attributes.
3. The Master Identity Administrator or Delegated Identity Administrator configures workflows for creating a location. See ["Chaining Identity Functions Into Workflows"](#) on page 5-1 for details.
4. The Delegated Identity Administrator creates a new location and establishes the location's hierarchy in relation to other locations, if applicable.
5. The Delegated Identity Administrator or user assigns a value for the location attribute for a user or object profile.

Any user with appropriate permissions can now view the user or object location.

About Assigning the Right to Create Groups in Group Manager

You assign users the right to create a group when you define a Create Group workflow. Only users designated as participants in the workflow can create the group. See ["Chaining Identity Functions Into Workflows"](#) on page 5-1 for information about creating workflows.

A user can be assigned the right to modify a group type if the user is a participant in a Create Group workflow for that group type. The user must also have write access for the group type attribute. See ["Adding Auxiliary and Template Object Classes to a Group Tab"](#) on page 4-9 for information about group types. Also see ["Setting and Modifying LDAP Attribute Permissions"](#) on page 4-33 for information about assigning the modify right to the Group Type attribute.

If you run the Identity System with multiple Active Directory instances and use a dynamic filter to create a group, the filter attribute must be a multi-value attribute.

If you run the Identity System with the NDS directory, the users you select as members of the group are cleared from the page when you click Save. To prevent this from happening, go into the NDS directory and switch the order of the attributes so uniquemember is read first. Also make sure the userCertificate attribute comes before the NDS userCertificate;binary attribute.

Configuring a Group to Accept Subscriptions

If your directory contained groups prior to the installation of Oracle Access Manager, the pre-existing groups do not automatically support group subscriptions. The following task overview describes the general steps for configuring legacy groups in the directory.

Task overview: Configuring a legacy group to support subscriptions

1. Associate the oblixadvancedgroup auxiliary class with the group structural class in the Group Manager tab.

See ["Adding Auxiliary and Template Object Classes to a Group Tab"](#) on page 4-8 for details.

This enables new groups to use subscription policies, provided that you configure corresponding Create Group workflows.

2. Perform a directory search to retrieve the DNs of all existing groups that do not yet contain the `oblixadvancedgroup` auxiliary object class.
3. Create an LDIF file that contains these DNs.
4. Add the `oblixadvancedgroup` auxiliary class to the entries in the LDIF file.
5. In the LDIF file, assign one of the following values to the attribute `obGroupSubscriptionType`:
 - `SubscriptionPolicyOpen`: Matches the Open option in the user interface.
 - `SubscriptionPolicyOpenFilter`: Matches Open With Filter in the user interface.
 - `SubscriptionPolicyControlledWorkflow` : Matches Controlled Through Workflow in the user interface.
 - `SubscriptionPolicyClosed`: Matches Closed in the user interface.

See [Table 5–8](#) in "Starting a New Workflow Definition" on page 5-20 for more information on the subscription policies.

6. Upload the LDIF file.

All groups in the directory are now enabled to use the subscription policy that you configured.

End-User Scenarios for the Group Manager

The following sections describe how an end user interacts with the Group Manager application once it has been configured:

- [Managing Group Members in Group Manager](#)
- [Searching for Group Members](#)
- [Deleting Group Members](#)
- [Adding Group Members](#)
- [Managing Group Subscriptions](#)
- [Subscribing to Groups](#)

Note: When searching for a large static group, the search may omit subgroups of this group. An administrator can tune the default processing for the group for performance reasons, which can produce this behavior. See the chapter on performance tuning in the *Oracle Access Manager Deployment Guide* for details.

Managing Group Members in Group Manager

You can view and manage group members from the Group Profile page if the Master Identity Administrator selected a group-member attribute to display on the group profile page. See ["Configuring Tab Profile Pages and Panels"](#) on page 4-12 for more information.

If your group contains a large list of members, this can negatively impact system performance. The Master Identity Administrator can choose not to display group members on the Group Profile page. See ["Configuring Group Manager Tab Options"](#) on page 4-10.

You can tune the processing of large static groups if these groups adversely affect performance. See the chapter on performance tuning in the *Oracle Access Manager Deployment Guide* for details.

You can view and manage group members from the Manage Group Members page. When managing large static groups, Oracle recommends using the Manage Group Members page because it is optimized to manage groups with 1000 or more members. This will significantly improve performance when managing large groups, in contrast to defining the member semantic attribute as part of the Group Profile page.

Searching for Group Members

The Manage Group Members page enables you to view the members of a group based on criteria that you provide. This page shows tables for:

- Static members
- Dynamic members
- Nested members

Search results are subject to searchbase and attribute access controls configured for the Group and User Manager applications. See ["Setting and Modifying LDAP Attribute Permissions"](#) on page 4-33 for details.

If a user does not have read access to the dynamic member attribute for a group, nothing appears in the dynamic member table and the following error message is shown, "You don't have read access for a dynamic member."

In the nested members table, if the group contains dynamic nested groups and the user does not have read access to the dynamic member attribute for some of the nested groups, the dynamic members are not shown. In this case, no error message is displayed.

Note: When searching for members of a large static group, it is possible that subgroups of this group are omitted. An administrator can tune the processing of the group for performance reasons, which causes this behavior. If a user to a subgroup, go to the profile page for the subgroup and search for the user there.

To view a group

1. In the Identity System Console, click the Group Manager tab.
2. In the Search field at the top of the page, enter search criteria.
3. Click Go.
A list of groups appears.
4. Click the link for the name of the group that you want to view.

To view group members

1. In the Group Manager, click My Group.
2. Conduct a search on groups and click the desired link.
The group profile appears.
3. Click Manage Group Members.
4. Select the Member Type you are searching for in this group:

- Select People to search for users. Search results can include static, nested, and dynamic users.
 - Select Groups to search for groups. Search results can include static groups and any other entity that is configured to be managed in the Group Manager.
5. From the Search Members By list, select an attribute as the basis for the search.
 6. Select a search operator.
 7. Enter search criteria.
 8. Click Go.

By default, the Manage Group Members page displays nested groups and their members in the search results. This includes a child nested group, its members, and its children. An administrator can modify what is shown on this page if the group is too large and requires performance tuning.

Deleting Group Members

You can delete group members displayed in the search results from the Manage Group Members page. You can only delete static members. You cannot delete dynamic or nested members.

To delete group members

1. Search for group members from the Manage Group Members page.
See ["Searching for Group Members"](#) on page 4-41.
2. From the results that are returned on the search, click the checkbox next to the entry that you want to delete.
3. Click Save on the Manage Group Members page.

Adding Group Members

You can add members to a group.

To add group members

1. Go to the Manage Group Members page, as described in ["Searching for Group Members"](#) on page 4-41.
2. From the Manage Group Members page, click the Select Member button beside the Members To Add field.
The Selector page appears. See ["The Selector"](#) on page 1-10 for details.
3. From the Selector page:
 - If you want to add users to this group, select the person member type.
 - If you want to add nested groups to this group, select the group member type.
4. Click Add for each member you want to add.
5. Click Done.
6. Click Save on the Manage Group Members page.

Managing Group Subscriptions

The Group Manager provides the ability for users to subscribe and unsubscribe to groups.

Subscription policies are not required for adding users to groups. For example, if you have pre-existing groups and want to add users to these groups, you can set the relevant access control policies on the `uniqueMember` attribute and use the Selector to add members. However, if you want users to subscribe themselves to groups, you implement this using a subscription functionality.

Only groups configured as Advanced Groups can include a subscription policy. The `oblixAdvancedGroup` is provided by the Identity System to give you attributes that you might need when working with groups. [Table 4–5](#) shows the contents of `oblixAdvancedGroup`:

Table 4–5 Contents of `oblixAdvancedGroup`

Attribute	Characteristics
<code>obGroupAdministrator</code>	Display Name: Group Administrator Semantic Type: Group Administrator Display Type: Object Selector
<code>obGroupDynamicFilter</code>	Display Name: Dynamic Filter Semantic Type: Group Dynamic Member Display Type: Filter Builder
<code>obGroupExpandedDynamic</code>	Display Name: Group Expansion Semantic Type: None Display Type: Radio Buttons Comment: This attribute is used for expanded dynamic groups.
<code>obGroupPureDynamic</code>	Display Name: Dynamic Members Only Semantic Type: None Display Type: Radio Buttons Comment: This attribute indicates whether the group is purely a dynamic group. It affects subscriptions.
<code>obGroupSimplifiedAccessControl</code>	Display Name: Group Access Semantic Type: None Display Type: Radio Buttons Comment: This attribute is used for creating a group workflow. It controls the simplified access control feature.
<code>obGroupSubscribeMessage</code>	Display Name: Subscription Message Semantic Type: None Display Type: Multi-Line Text Comment: This attribute is used for subscription notification.
<code>obGroupSubscribeNotification</code>	Display Name: Notification Semantic Type: None Display Type: Check Box Comment: This attribute is used for subscription notification.

Table 4–5 (Cont.) Contents of *oblixAdvancedGroup*

Attribute	Characteristics
obGroupSubscriptionFilter	Display Name: Subscription Filter Semantic Type: None Display Type: Filter Builder Comment: This attribute is used for group subscriptions using a filter.
obGroupSubscriptionType	Display Name: Subscription Policy Semantic Type: None Display Type: Selection Menu Comment: This attribute is used for group subscriptions.
obGroupUnsubscribe Message	Display Name: Unsubscription Message Semantic Type: None Display Type: Multi-Line Text

Note: If you create a static group with one or more members and then modify the group so that the Dynamic Members Only flag is set to true, the Identity System enables you to do so without issuing a warning.

Subscribing to Groups

There are three ways a user can subscribe to a group, assuming the Master Identity Administrator configured a group subscription policy for that group:

- From the Group Profile page in Group Manager
This enables users to subscribe to the selected group displayed in the profile.
- As the last step of a Create User workflow
Users can subscribe to multiple groups during the last step of a create user workflow. See "[Chaining Identity Functions Into Workflows](#)" on page 5-1 for more information.
- From the Manage Subscriptions page in Group Manager
This enables users to subscribe to multiple groups from the Manage Subscriptions page.

To subscribe to a group

1. From the Identity System Console, click the Group Manager tab.
2. Conduct a search for groups using the search bar.
3. Click the link for the group to which you want to subscribe.
4. Click Subscribe.

To subscribe to multiple groups

1. From the Group Manager application, click Manage Subscriptions.
2. Conduct a search for groups using the search bar.

3. In the Groups for Subscription page, check the box next to each group to which you want to subscribe.
4. Click Save Subscriptions at the bottom of the Manage Subscriptions page.
A list of groups to which you are subscribed appears. This includes:
 - All groups with an open subscription policy.
 - All groups with filter subscription policy, and you satisfy the filter criteria.
 - All groups controlled through a workflow subscription policy where you are a participant in the initiating step of the change-attribute workflow that applies to these groups.

Configuring Auditing Policies

You can capture information about user actions performed in each Identity application. Captured information is stored as audits of Identity System events.

You configure auditing policies for each application. These settings determine the data that is captured in an audit file. You configure where audit files are written for each Identity Server. For information about changing the audit file path, see ["Managing Identity Servers"](#) on page 7-14.

Viewing Auditing Policies

You can view auditing policies from each Identity System application.

To view auditing policies

1. In the Identity System Console, click User, Group, or Org. Manager Configuration, then click Audit Policies.

The Application Auditing Policy page appears, displaying the following information:

Item	Description
Profile Attributes	Appears only if they have been configured for this application.
Event Name	the Identity System operation being audited
Application Auditing Enabled	Indicates whether or not auditing is enabled for this event
Audit Success	Indicates whether or not event successes are audited
Audit Failure	Indicates whether or not event failures are audited

Modifying Auditing Policies

If you have appropriate permissions, you can change any auditing policy that you can view. These settings do not overlap with the Global Auditing Policies feature found in the Identity System Console, Common Configuration, Global Auditing Policies.

To set or modify auditing policies

1. In the Identity System Console, click User, Group, or Organization Manager Configuration, Audit Policies.
2. Click Modify.

The Modify Auditing Policy page appears.

3. In the Profile Attributes lists, select the attributes that can trigger events you want to audit.
4. In the Application Auditing Enabled column, select each event you want to enable for auditing.
5. In the Audit Success and Audit Failure columns, select each event you want to audit.

For example, you can audit every Modify Location event, but audit only View Profile failures.

6. Click Save.

You return to the previous page.

Generating Reports

Reports enable you to view information about an object class. Reports provide an alternative to searches and enable you to report on attributes that are not available from a search.

Configuring Reports

Master Identity Administrators must define a report from the Identity System Console before users can view the report in the User Manager application.

For example, after configuring an Employees tab for User Manager, as described in ["Viewing and Modifying Tab Configuration Information"](#) on page 4-3, you can create reports listing employees in a specific building, employees with specific job titles, or employees in a particular department.

The report functionality uses QueryBuilder to enable you to define complex search criteria that are not possible using a basic search. This provides richer support for searching on various types of attributes that a basic search does not allow. There are two types of reports:

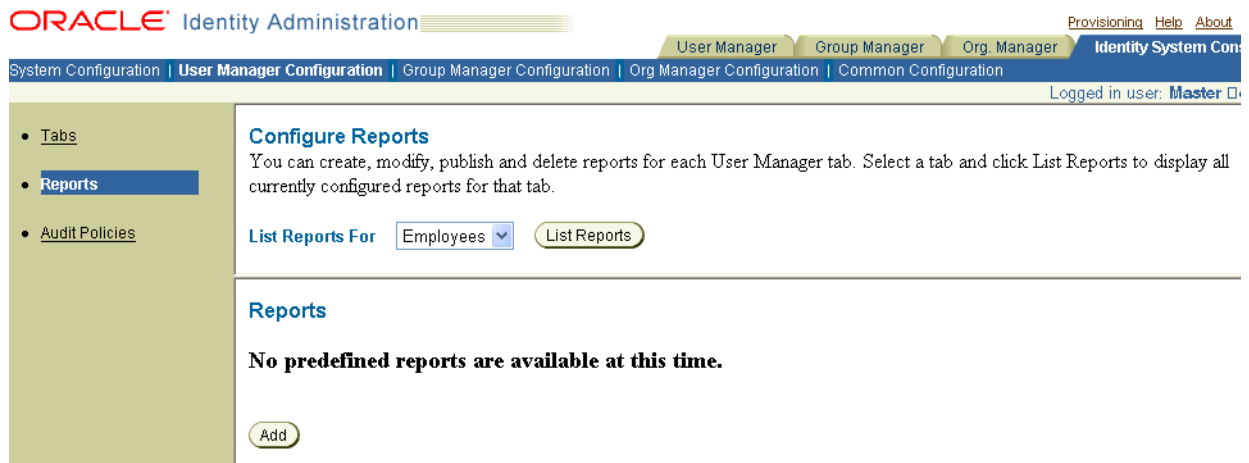
- **Ad-hoc Reports:** Created by end users in the User Manager, Group Manager, and Org. Manager applications. In this case, the Query Builder includes searchable attributes configured in the tab along with other supported display types (see the note).
- **Pre-defined Reports:** Created by an administrator from the System Console. In this case, the Query Builder includes all attributes (whether they are marked as searchable in the tab or not) for all supported display types.

Note: The Query Builder supports building filters on attributes with the following display types: Single line text, Multi-line text, Radio, Select list, Checkbox, Boolean, Date, Mail address, Telephone number, Selector, Postal address and Numeric String.

To configure a report

1. From the Identity System Console, click User Manager Configuration, Reports, then click List Reports.

The following page appears the first time you create a report.



2. Click Add to display the Query Builder on the Configure Reports page.
3. Select the first Attribute for the basis of your report criteria, then click Add.
4. In the field beside the attribute, select the appropriate method.
5. Enter the report criteria.
The format of this criteria depends on the attribute display type.
6. Repeat steps 3-6 for any additional attributes you want added to this report.

Note: When you select more than one attribute for a report, you must select whether this is an And or an Or operation. See the sample page in this procedure.

7. Click Test to verify that the report generates data correctly.
A verification page appears.
8. Click Save.
A page like the following one appears. Several buttons become available and are highlighted in the following screen shot. These will be used in the next procedure.



ORACLE Identity Administration User Manager Group Manager Org. Manager
System Configuration | **User Manager Configuration** | Group Manager Configuration | Org Manager Configuration | Common Con
Logged in user: Master Fdm

- [Tabs](#)
- **[Reports](#)**
- [Audit Policies](#)

Configure Reports
You can create, modify, publish and delete reports for each User Manager tab. Select a tab and click List Reports to display all currently configured reports for that tab.

List Reports For:

Report
Displaying 1 to 8 of 12 results

Full Name	OU	Phone Number	Title	Login
Borneto Avellaneda	Human Resource Los Angeles Corporate HQ	714 372-5085	Director	bavellan
Dee Aimon	Sales San Jose Dealer1k4 Mercury	415 717-5707	Manager	daimon

9. Click Next to see additional report results, or click Publish to save the report.

To change the formatting of a report

1. From the Reports page, click Customize to customize the report column headings.
2. Customize the column names in the form that appears, then click Save.
3. Click the Publish button.
4. Enter a Name and an optional description for this report.
5. Click Save to make this report available in the User Manager application, under the Reports tab.

Viewing, Modifying, Localizing, and Deleting Reports

Viewing reports is subject to access control and searchbase settings.

You can display a report's name and description in more than one language if you install the appropriate language packs and configure them for those languages. See "[Configuring Multiple Languages for Oracle Access Manager](#)" on page 7-7 for more information.

When you export a generated report in an Identity System application and the values contain non-ASCII characters, you must rename the file with a .txt extension. It will pass through Excel's Import Wizard and non-ASCII characters will display properly.

Note that .csv files opened in OpenOffice will pass through the Import Wizard. In this application, encoding can be chosen without renaming the file to *.txt.

To view or modify reports

1. From the Identity System Console, click User Manager Configuration, then click Reports.
2. Select the type of report you want to view or modify from the list.
3. Click List Reports.
4. Select the link to the report you wish to view.

5. Click the Customize button to change the report criteria.
6. Click Save to save the new report format.

See ["Configuring Reports"](#) on page 4-46 for more information on publishing reports for others to view.

To localize reports

1. From Identity System Console, click User Manager Configuration, then click Reports.
2. Click List Reports.
All existing reports are listed on the page.
3. Click the report that you want to localize.
The report details appear on the page.
4. Click Publish.
The Publish Report page appears. This page contains the links for all the installed languages.
5. Click the language in which you want to publish the report.
6. In the Report Name field, enter a display name in the selected language.
7. In the Report Description field, enter a brief description of the report.
This information is optional.
8. Click Save to save your changes.
The reports are displayed in the User Manager.

To delete reports

1. From Identity System Console, click User Manager Configuration, then click Reports.
2. Select the tab that contains the report you want to delete.
3. Click List Reports.
4. Select the (-) icon next to the report name to delete it.

Advanced Configuration

The following sections describe expanding dynamic groups, limiting the scope of a directory search, and editing an XML file to configure attribute permissions.

Expanding Dynamic Groups

If a group's membership is determined by an LDAP filter, you can generate a static membership list by expanding the group. Generating a static list saves the Identity System from having to run the LDAP filter with every group access.

Group expansion updates the static list by running the LDAP rule that specifies dynamic membership, then storing the results in the static member attribute. Many Identity System functions test a group for membership. Since testing static membership is faster than testing dynamic membership, it is preferable to find a member in a static list. Also, third-party applications may only be able to check static

membership. Frequent expansion keeps static membership accurate for third-party applications.

The group expansion operation itself is an expensive process. However, you can expand a group as a background process so the impact is hidden from users.

Note: If you have static members in a dynamic group and you expand the group, the original list of static members is overwritten with the members who currently satisfy the filter criteria. This is true even if you have set the flag for dynamic members only to false. The filter overrides other group settings.

Before a user can expand a group, two conditions must be met:

- The `obgroupexpandeddynamic` attribute must be set to true.
- The person expanding the group must have Read permission for two attributes, `obgroupexpandeddynamic` and `obgroupdynamicfilter`. The user also must have Write permission for the attribute assigned the Group Static Member semantic type.

See the table in "[Managing Group Subscriptions](#)" on page 4-43 for a breakdown of the Identity System-supplied group attributes.

To expand a dynamic group

1. In Group Manager, click the Configuration option at the top of the page.
The Configuration page appears.
2. Click Expand Dynamic Groups.
The Expand Dynamic Group page appears.
3. Select one of these options:
 - Select By Group and click Select Group to choose one or more groups
 - Select All to expand all groups
4. Click Expand.
The Expanded Groups page displays a list of all groups that have been expanded.
5. Click the group link to display the Group Profile page for that group.
6. Click Done.

Modifying the Default Searchbase Scope

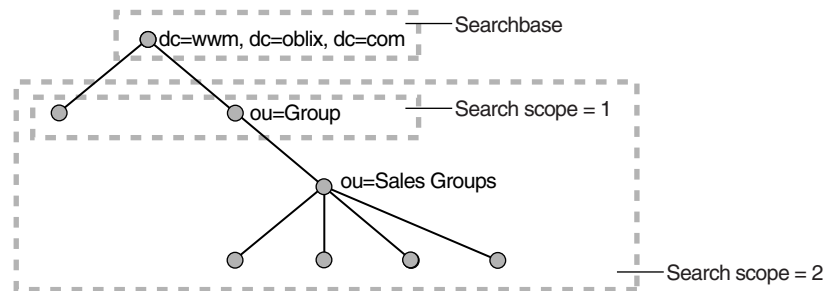
Some portions of the Identity System call out to external XML files to get configuration information. The `globalparams.xml` file is one such file. This file controls search scope among other things.

By default, the search scope is set to subtree for the Identity System, meaning that the search begins at the starting point of the searchbase and includes its children. Depending on the size of your directory, you may want to change the default search scope using the `ResourceFilterSearchScope` parameter. The possible values for this parameter are:

- 1: Search the top node of the searchbase and the first level under it.
- 2: Start at the top node of the searchbase and proceed to the bottom node.

Figure 4-1 shows that setting ResourceFilterSearchScope to 1 could limit results to just a few returned entries, while setting it to 2 could return thousands of entries.

Figure 4-1 Search Scope Options



To set the globalparams.xml file

1. Locate the globalparams.xml file in the following directory:
IdentityServer_install_dir/identity/oblix/apps/common/bin
2. Back up the file.
3. Open the file in an ASCII editor (for instance, Notepad) or an XML editor.
4. Find the ResourceFilterSearchScope parameter and change the value.
5. Restart WebPass and the Identity Server.

Simplified Attribute Permissions for a Group

Simplified attribute permissions lets a group creator select Read, Write, and Notify permissions without having to set permissions for each attribute as described under ["Setting and Modifying LDAP Attribute Permissions"](#) on page 4-33.

Simplified permissions are applied to newly created groups where the management domain of the policies is the DN of the new group. Later, these policies can be modified through the access control feature.

Implementing Simplified Permissions

An administrator can configure as many sets of simplified permissions as needed. The administrator creates permissions in the *IdentityServer_install_dir/oblix/apps/groupservcenter/bin/gscacparams.xml* file.

This file contains embedded compound lists to define the roles, users, and groups the model applies to, the rights assigned, and the attributes to which the rights apply. When this file is applied to a new group, an access control entry is created for each right in the file.

Sample gscacparams.xml File

The following is a sample set of permissions within a gscacparams.xml file. The model name is Public:

- In entry 1, the role is ob_any, the right is read, and the attributes are description, uniquemember, and owner.
- In entry 2, the role is owner, the right is write, and the attributes are description, uniquemember, and owner.

Example 4-1

```

<?xml version="1.0"?>
<ParamsCtlg xmlns="http(s)://www.oblix.com" CtlgName="gscacparams">
<!--#----->
<!-- #Access Control Functions -->
<!--#----->
<!--#----->
<!-- # Public access -->
<!--#----->
<CompoundList ListName="">
<CompoundList ListName="Public">
<CompoundList ListName="entry1">
<ValList ListName="roles" >
<ValListMember Value="ob_any">
</ValList>
<ValList ListName="rights" >
<ValListMember Value="READ" Operation="Add"/>
</ValList>
<ValList ListName="attributes" >
<ValListMember Value="description"/>
<ValListMember Value="cn"/>
<ValListMember Value="uniquemember"/>
<ValListMember Value="owner"/>
</ValList>
</CompoundList>
<CompoundList ListName="entry2">
<ValList ListName="roles" >
<ValListMember Value="owner" Operation="Add"/>
</ValList>
<ValList ListName="rights" >
<ValListMember Value="WRITE" Operation="Add"/>
</ValList>
<ValList ListName="attributes" >
<ValListMember Value="description" Operation="Add"/>
<ValListMember Value="cn" Operation="Add"/>
<ValListMember Value="uniquemember" Operation="Add"/>
<ValListMember Value="owner" Operation="Add"/>
</ValList>
</CompoundList>

```

Simplified Permissions Reserved Words

The following table summarizes the reserved words for simplified permissions.

Reserved Word	When Used	Description
rights	Once for an entry	Specifies the right: read, modify, or notify.
attributes	Once for an entry	List that specifies the attributes. Any group object attribute can be added to the list.
roles	Once for an entry	Roles to which entry applies. Roles can be any pre-defined role, such as uniquemember, owner, ob_any, or ob_anonymous.
people	Once for an entry	Specifies the distinguished names to which this entry applies.
source	Once for a model	Specifies the base uid of the users who will see this model. If a base uid is not specified, everyone can see this entry.

Reserved Word	When Used	Description
target	Once for a model	Specifies the base uid of the target where this model applies. If the group is not part of this base, the rights cannot be set.

Setting Container Limits in Organization Manager

Use the Container Limits function to control the number of objects and child objects for an organizational unit and its object classes. You can define who receives notifications when the limit is about to be exceeded. For example, you can have organizational units in your directory tree that you use for storing extranet customers. You can limit to 10,000 the number of customers with access to your extranet portal.

Note: The Container Limits feature counts the number of objects from the directory. If the number of objects is very large, performance can be affected.

To view and add container limits

1. From the Identity System Console, click the Org. Manager tab, then click Configuration, then click Container Limits.

The Container Limits page appears.

ORACLE Identity Administration

User Manager Group Manager **Org. Manager** Identity System Co

Location | Inventory | OrgTab

Reports | Create New Org. | Requests | **Configuration**

Search Location Name That Contains All 8 Results Go Advanced Logged in User: Master Edmun

Attribute Access Control Delegated Administration Workflow Definition **Container Limits**

Container Limits

1) Management Domain

o=company,c=us

Current Count

Objectclass	One	Total
geninventoryobject	0	626
gensiteorgperson	9	718
groupOfUniqueNames	5	243
oblixlocation	0	0
organizationalunit	3	239

Show All

Add Container Limit to Objectclass

oblixlocation Add

Objectclass	Container Limit	Enforce	Notify
gensiteorgperson	1		

Modify Delete Delete All Copy Paste

Report

In this example, the Current Count table on this page indicates that the gensiteOrgPerson object class has 9 children stored at the current level of the DIT and 718 total children at or under this level.

2. In the Management Domain box, select a DIT entry you want to view.

The Current Count box displays all configured structural classes associated with the entry and the number of their children.

The Objectclass table displays the container limit, enforcement, and notification policies for the selected DIT entry, listed according to object class.

3. Select an object class and click Add to add a container limit, in the Objectclass list.

A second Container Limits page appears showing the Management Domain and Object class you selected in the previous screen.

The screenshot shows the Oracle Identity Administration web interface. The top navigation bar includes 'User Manager', 'Group Manager', and 'Org. Manager'. The breadcrumb trail is 'Location | Inventory | OrgTab'. The main menu has 'Reports | Create New Org. | Requests | Configuration'. The 'Configuration' section is active, with sub-tabs for 'Attribute Access Control', 'Delegated Administration', 'Workflow Definition', and 'Container Limits'. The 'Container Limits' page displays the following fields:

- Management Domain:** A text box containing 'o=company,c=us'.
- Objectclass:** A text box containing 'groupOfUniqueNames'.
- Container Limit:** A text box for the limit value, followed by a checkbox labeled 'Notify if used up' and a percentage input box.
- Override subordinate policies:** A checkbox.
- Rule:** A section with a 'Build Filter' button and a large text area.
- Person(s):** A section with a 'Select User' button and a large text area.
- Group(s):** A section with a 'Select Group' button and a large text area.

At the bottom of the form are three buttons: 'Save', 'Cancel', and 'Reset'.

4. In the Container Limit box, specify the maximum number of children this object class can contain at this DIT level.
5. When you want to notify someone by email that your object class is nearing its container limit, select Notify if used up, and specify the limit percentage when you want the email sent.
6. Select Override subordinate policies to create a container limit that cannot be overridden by a lower policy on the DIT.
7. Use one or more of the following to specify the persons to receive container limit warnings:
 - Select Build Filter, then use the Query Builder to create a rule.
 - Click Select User, then use the Selector to specify one or more users.
 - Click Select Group, then use the Selector to specify one or more groups.

The Users, Roles, and Rules fields have an or relationship. Users specified in any of the fields are notified.

8. Click Save to save your container limit and add it to the Objectclass table.

Copying Container Limits

You can copy container limits from one domain to another.

To copy container limits from one domain to another

1. From the Organization Manager, click Configuration, then click Container Limits.
The Container Limit screen appears (as shown on
2. In the Management Domain box, select the directory information tree (DIT) entry you want to view.
The Current Count box displays the structural classes associated with the entry and the number of their children.
The table Add Container Limit to Objectclass displays the container limit, enforcement, and notification policies for the currently selected DIT entry, listed according to object class.
3. Click Copy.
4. In the Management Domain box, locate the destination entry where you want to add the container limits.
5. Click Paste.
The container limit policies are added to the selected DIT entry.

Modifying Container Limits

You can change container limits. See ["Setting Container Limits in Organization Manager"](#) on page 4-53.

To modify a container limit

1. In the Organization Manager, click Configuration.
The Configuration screen appears.
2. Click Container Limits.
The Container Limits screen appears.
3. In the Management Domain box, select the DIT entry you want to view.
The Current Count box displays all configured structural classes associated with the entry and the number of their children.
4. In the Add Container Limit to Objectclass panel, select an object class from the Objectclass column.
5. Click Modify.
The second Container Limits screen appears.
6. Make your changes.
See ["Setting Container Limits in Organization Manager"](#) on page 4-53 for information about these fields.
7. Click Save.
You can delete a container limit.

To delete a container limit

1. In the Organization Manager, click Configuration, then click Container Limits.
2. In the Management Domain box, select a directory information tree (DIT) entry.
The Current Count box displays all configured structural classes associated with the entry and the number of their children.
3. In the Add Container Limit to Objectclass panel, select an object class.
4. Click Delete.

The object class container limit is deleted.

Note: Click Delete All to delete all container limits for a DIT entry.

Chaining Identity Functions Into Workflows

This chapter includes the following topics:

- [About Workflows](#)
- [Using the QuickStart Tool](#)
- [Using the Workflow Applet](#)
- [Defining a Subflow](#)
- [Advanced Workflow Ticket Routing](#)
- [Performing Asynchronous Operations](#)
- [Using a Workflow](#)
- [Managing Workflows](#)
- [Advanced Workflow Options](#)
- [Creating a Self-Registration Workflow](#)
- [Creating a Location Workflow](#)

About Workflows

An Identity System *workflow* enables Master Identity Administrators and Delegated Identity Administrators to apply business logic to Identity System functions. A workflow organizes and automates complex procedures, for example, creating benefits and email accounts for new employees or changing user profile attributes in the directory.

Each workflow consists of a sequenced chain of actions. For example, an administrator can add a new user identity in the User Manager, and after submitting the new information, the new information can be routed to various departments for additional information, approvals, and so on. Rather than making a single person responsible for completing all the tasks in the workflow, you can assign each step to the specialist most appropriate to perform that step. When a step is completed, the workflow engine can send the workflow ticket to the person responsible for the next step in the sequence.

In sum, workflows enable you to:

- Automate and standardize processes for creating objects, deleting objects, and modifying attributes in the directory.
- Apply data integrity and rule checking when creating objects, deleting objects, and modifying attributes.

- Configure the Identity System as a data entry system for provisioning back-end applications.

As with user, group, and administrator information, Oracle Access Manager stores information about the workflow definition in the directory. This includes information about the people who can approve a step in a workflow. Workflow definitions are located in the configuration branch of the directory, as opposed to the user or group branch.

How Workflows Are Initiated

Workflows can be initiated by a user. For example, a new employee can initiate a self-registration workflow.

Workflows can also be initiated programmatically. For example, you can initiate the `workflowSaveCreateProfile` function in IdentityXML to initiate a create user workflow. See *Oracle Access Manager Developer Guide* for details.

You can also copy a link to a URL for the workflow and embed it in a page from another application, and access it as a portal insert. See the *Oracle Access Manager Customization Guide* for details.

Typical Workflow Examples

Workflows are appropriate for just about any frequently repeated, multi-step task involving any combination of user actions or automated data retrieval. Each workflow is associated with one of the Identity System applications. The following list covers some common workflows:

- **User Manager:** You can define a workflow to permit users to change their department number and phone number pending approval by a manager. You can ensure that when a new user is created, the appropriate people obtain information about this person programmatically from an external system.

A different workflow can add new users to your corporate email application. If you have defined an object template schema, you can use a workflow to send data from an Identity System application to a back-end application for provisioning. See ["Sending Non-LDAP Data to External Applications"](#) on page 6-1 for details on object templates.

- **Group Manager:** You can create a workflow to route group registration requests to a manager for approval.
- **Organization Manager:** You can give a supplier the ability to create entries for parts, pending manager approval of each entry the supplier adds. You can also create a workflow that first enables a user to add a new part entry, then routes the request to add the data to an appropriate person for approval, and finally, permits the person giving approval to commit the new data to the directory.

Advanced Workflow Options

The Identity System workflows support the following advanced features:

- *Subflows* enable certain workflow activities to occur in parallel.

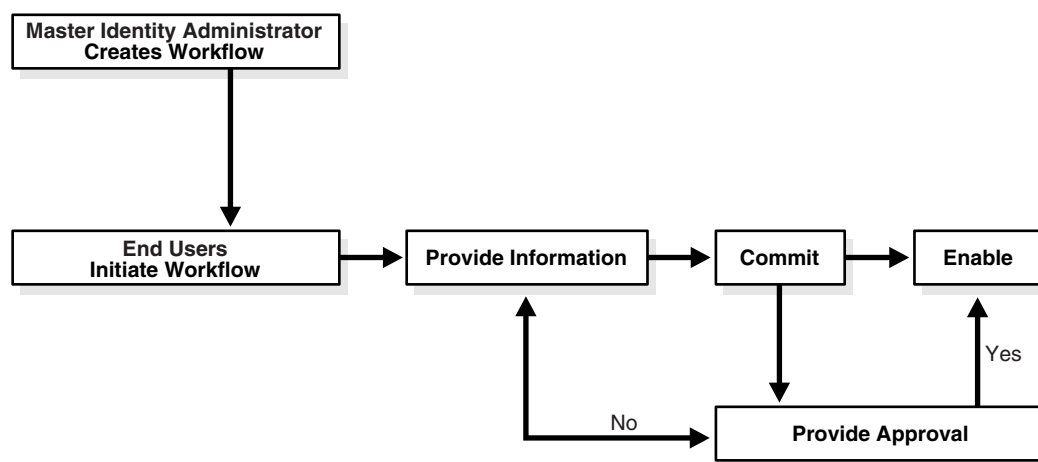
For instance, if a request to create a new user requires approvals from two different departments, both parties can receive approval requests simultaneously. See ["Defining a Subflow"](#) on page 5-32 for details.

- You can route specific workflow steps to different *dynamic participants*, who are selected based on attribute values or business logic evaluated at runtime.
See ["Specifying Dynamic Participants"](#) on page 5-35 for details.
- You can designate *surrogates* to assume responsibility for a step when the primary participant assigned to that task is out of the office or otherwise unavailable to process incoming tickets.
See ["Specifying Surrogates"](#) on page 5-41 for details.
- You can configure *time-based escalation* so that a workflow ticket is routed to a different participant if the original participant does not complete the assigned step within a specific period.
See ["Enabling Time-based Escalation"](#) on page 5-43 for details.
- You can invoke a workflow from any Web page as a portal insert or application using IdentityXML.
See the *Oracle Access Manager Developer Guide* for details.
- Workflow *auditing* enables you to monitor the state of a workflow and to determine exactly who performed particular actions at each step in the process.
See ["Monitoring a Workflow"](#) on page 5-51 for details.
- For actions that do not require human intervention, you can configure workflow steps so that the Identity System automatically obtains the required data from external sources.
See the *Oracle Access Manager Developer Guide* for details.

Workflow Types

Workflows come in various types. For instance, one type of workflow enables you to change one or more attributes for an existing object. Another type of workflow enables you to create a new object. [Figure 5-1](#) illustrates a Create User workflow:

Figure 5-1 Create User Workflow



Creating Workflows

The following overview summarizes the high-level steps for creating a workflow. The actual steps vary slightly for Change Attribute workflows and Create User (or Group, or Object) workflows:

Task overview: Creating a workflow definition

1. Add objects to the tab for the relevant Identity System application.
2. Configure attributes for the objects.
3. Configure read and write permissions for LDAP attributes.

Participants in a workflow must have appropriate read and write permissions for LDAP attributes that are viewed and changed during the processing of the workflow. See ["Allowing Users to View and Change LDAP Data"](#) on page 4-22 for details.

4. Add the attributes to the application panels.

This applies to both LDAP and template attributes. For Change Attribute workflows, users will not see the attributes from template objects on the profile page that contains the panel. However, the template attributes must be added for the workflow to operate properly.

5. Configure the workflow.

As discussed in ["Defining Step Attributes"](#) on page 5-26 for details, when you add attributes for a step in a Change Attribute workflow, the topmost attribute in the list must have been configured on a profile page. The subsequent attributes in the list will be added to the page automatically as long as the topmost attribute has been configured correctly.

A workflow definition changes the appearance of its related profile page in the Identity System application for which the workflow was created. For example, when a Modify Attribute workflow has been configured correctly, a Modify button appears next to the attribute on the Modify Profile page for the target object. When a Create User workflow has been configured correctly, the desired attributes appear when you select Create User Identity in the User Manager.

How Users Access Workflows in an Identity System Application

After a workflow definition has been created, an instance of the workflow is initiated in one of several ways, depending on the type of workflow that is being used:

Table 5–1 Methods for initiating Workflows

Workflow Type	User Initiates This Workflow From. . .
Change Attribute	A Request to Modify button on a Modify Profile page for the user
Create User	The Create User Identity page that is accessed from the Create User Identity link in the User Manager.
Deactivate User	An Initiate User Deactivation button on the View Profile page for the user.

Table 5–1 (Cont.) Methods for initiating Workflows

Workflow Type	User Initiates This Workflow From. . .
Reactivate User	An Initiate User Reactivation button appears on the View Profile page when you have created a Reactive User workflow. You first must find the user from the Deactivated User Identity page in the User Manager. A Reactivate user operation can be done by a Directory Administrator or a user with reactivate privileges.
Self-Registration	When this type of workflow is created, a URL is generated that initiates this workflow. You must save the URL and use it to initiate the Self-Registration workflow.
Create Group	The Create Group page that is accessed from the Create Group link in the Group Manager.
Delete Group	The View Profile page for the group.
Create Object	Create page in the Organization Manager.
Delete Object	View Profile page for the object.

About Workflow Tickets

As program execution reaches a given step in a workflow, the workflow engine creates a *ticket* for that step instance. During a Create User workflow, for example, a ticket is typically sent to specific participants in IT as soon as the user selects the Create User function in the User Manager.

Each workflow ticket is initially displayed in the form of a link. When the participant clicks this link, he or she is prompted to perform the action associated with that step in the workflow. For example, when someone in IT processes a ticket for a Create User workflow, he or she is typically prompted to supply a login id and password for the new user.

A workflow log is created upon completion of each step in the workflow.

See ["Using a Workflow"](#) on page 5-47 for more information.

For example, the contents of a Create User Identity page may be based on attributes configured in a Create User workflow.

Once information about a new user is saved on this page, the initiate step of the workflow is complete. The workflow definition generates a ticket for this workflow instance, as illustrated in the following screen shot:

ORACLE Identity Administration

My Profile | Reports | Create User Identity | Deactivate User Identity | Substitute Rights | **Requests** | Configuration

Search Full Name That Contains All 8 Results Go Advanced Logg in Us

Incoming Request Outgoing Request Monitor Requests

Request

Request Number 669802c61f184d54a10a45d34c0434d5

Request Type Create User

Application Name User Manager

Requested For new2

Status Last step done

Workflow Name Create user - Basic

Parent Request Number

Step Number	Action	Action Taker	Status	Subflow Number	Escalation Count
1	Initiate	Josefa Collins	Completed		
2	Enable	Sri Damodaran	Completed		

Back Delete

A participant in this workflow can view the ticket generated for this workflow step, and can approve the addition of the new user. You display ticket information by clicking on the ticket number next to the approval label.

A Workflow Scenario

Suppose you create a workflow for adding a user in the Identity System. You could define a Create User workflow that performs the following steps. Use caution when you create a user that includes multi-byte characters in the password. User creation might fail if you are using a non-English keyboard. For more information, see ["User Creation Might Fail When You Have Multi-byte Characters in the Password"](#) on page F-9.

Process overview: Creating and using a Create User workflow

- From the User Manager application, you create a new workflow definition.
In this example, the workflow definition has three steps and specifies that anyone in IT who has logged in to the User Manager can create a new user. workflow:
Step 1. Initiate: This step allows anyone who has logged in to the User Manager to input data for a new user.
Step 2. Provide Information and Approval: This step allows the user's manager to approve the data entered for the user.
Step 3. Activate: This step activates the new user.
- A user logs in to the User Manager.
- The user selects a Create User button.
The workflow instance prompts the user to supply a name, user ID, and password for the new user, plus the user ID and email of the new user's manager.
- The workflow instance then routes a request to create the new user, along with information about the new user, to the manager of that user.
- The manager clicks the Requests function in the User Manager application to display the request in the form of a link to a job ticket.
- The manager clicks the link for the ticket to display the request.

7. To approve the request, the manager clicks a Process Request button.
8. In the Process Requests page, the manager clicks an Approve button.
9. The request is processed and the new user is enabled in the Identity System.

The user is now allowed to log in and use the functions they are entitled to as defined by their directory profile and the rights assigned to attributes in that profile by a Master administrator. See ["Allowing Users to View and Change LDAP Data"](#) on page 4-22 for details.

LDAP Versus Template Attributes in a Workflow

When you define a workflow, you have a choice of using two types of objects and attributes in most workflow steps:

- **LDAP Objects and Attributes:** You can use a workflow to modify objects and attributes that you have configured for an application profile page. The people who participate in the workflow must have appropriate privileges for viewing and modifying these objects and attributes.
- **Template Attributes:** If you are using a workflow to provision a back-end application, you configure workflow steps for adding information based on a template schema. When template attribute values are committed during the workflow, an Identity Event API plug-in can intercept this data and send it to a back-end application for provisioning. See ["Sending Non-LDAP Data to External Applications"](#) on page 6-1 and the *Oracle Access Manager Developer Guide* for details.

As of version 7.0, provisioning allows only for a one-way flow of data from the Identity System to the back-end system. As a result, you might want to configure provisioning workflows to write data to both the LDAP directory and to the back-end system. This enables your users to view the data that has been configured for the workflow target. However, to see the current state of the target in the back-end application, you must access the application or its logs.

For provisioning workflows, you should have separate Commit, Activate, Enable, Delete, Disable, and Deactivate steps for each schema to which the workflow is written.

Workflow Types, Steps, and Actions

A workflow *type* determines the purpose of the workflow, for example, creating a user. A workflow *step* is a discreet segment of the workflow. Steps are performed in a series. A workflow *action* is an activity performed during a step, such as issuing a request for information.

For example, the Create User workflow type enables you to create a directory entry for a user. This type of workflow can have actions for requesting information about the user, actions for collecting the information, actions for approving the request, and so on.

The following table correlates the different types of workflows to the Identity System applications:

Table 5–2 Workflow Types

Application	Workflow Type and Description
User Manager	■ Create User: Adds a user to the directory. ■ Self-Registration: Enables users to add themselves to the directory. ■ Deactivate User: Makes a user unable to log in and unavailable for viewing in the Identity System. Deactivation takes effect once a user has logged out. It removes a user's future access to the system. An administrator with sufficient access privileges can view deactivated users and either permanently delete them or reactivate them. ■ Reactivate User: Displays the Initiate User Reactivation button on the User Profile page and changes the status of a deactivated user, allowing the user to log in to and use the Identity System again. ■ Change Attribute: Changes an attribute value on a user profile. Attributes designated on this workflow will have a Request to Modify button on the target profile page.
Group Manager	■ Create Group: Adds a group to the directory. ■ Delete Group: Deletes a group from the directory. ■ Change Attribute: Changes an attribute value on a group profile. Attributes designated on this workflow will have a Request to Modify button on the target profile page.
Organization Manager	■ Create Object: Adds an object to the directory. ■ Delete Object: Deletes an object from the directory. ■ Change Attribute: Changes an attribute value on an object profile. Attributes designated on this workflow will have a Request to Modify button on the target profile page. ■ Self-Registration: Enables users to add organization objects to the directory.

About Workflow Steps

You must define at least two steps for each workflow: one to initiate an instance of the workflow and one to finish it. A step consists of the following:

- **A Number:** A unique identifier for this step.
All workflow steps have a number.
- **Actions:** An action is an activity can occur in the Identity System or in an outside system.
All workflow steps perform an action. Examples include starting the workflow, providing information, and requesting approval. See ["About Step Actions"](#) on page 5-10 for details.
- **Attributes:** An attribute value may be added or modified as part of a step.
For example, you might define a step for changing the value of a user's phone number attribute. Step attributes may be required, optional, or supplied by completion of another workflow step.

For values that are used locally within the Identity System, you configure LDAP attributes as part of the workflow. For provisioning to a back-end application, you configure both LDAP and template attributes in a workflow step.

Note: If Location ID has the Semantic type DN Prefix it is important to note Active Directory and ADAM do not allow multi-valued RDNs (although iPlanet/SunOne do). For Active Directory and ADAM, ensure that the Attribute Value(s) selection is Single in the meta-attribute configuration.

- **Participants:** A participant is a user or users who perform an action.

For example, for a Create User workflow, you may create an Initiate step and configure this step so that anyone who is logged in to the User Manager can start the process for creating a new user. Or you may define a specific participant in a workflow who is responsible for approving a change request. Participants can be assigned based on their role, name, group membership, or another characteristic.

For LDAP attributes, you can also define an LDAP filter that selects participants according to their DN.

Not every step in a workflow requires a participant. For example, in the default Create User workflow, the Enable step does not require participants. The action (enable) is performed automatically by the system when the previous step, for example, Approval, exits. The Commit and External Actions steps also do not use participants.

- **Target:** The person, group, or other LDAP object that is being created, deleted, and so on.

The target in the workflow definition is an LDAP object, not a template object.

- **Entry Conditions:** A step or subflow that must be completed before the present step.

For example, the first step in a workflow may be the Initiate step. The second step in the workflow may have an entry condition of successful completion of the Initiate step. Typically, the entry condition is successful completion of the previous step.

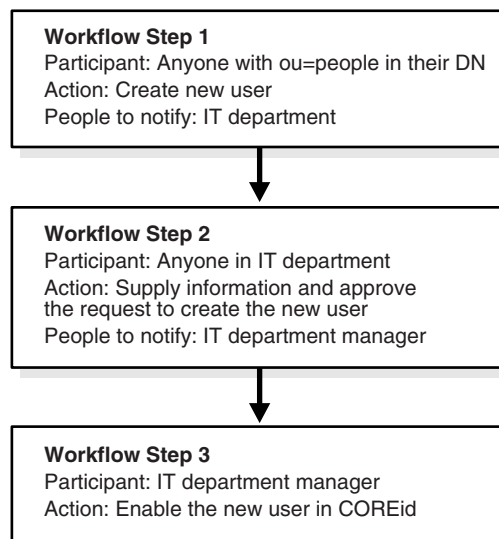
- **Notifications:** Users who receive email notification before or after a step is performed.

Other participants can see pending tickets in their incoming request queues whether or not email notification is configured. See "[Descriptions of Step Actions](#)" on page 5-13 for details.

- **Pre and Post Processing:** External functions that are executed as part of the workflow.

For example, in a create user workflow you might want to have a Java program that is called after an initiating step to assign a unique login ID.

A workflow process illustration is shown in [Figure 5-2](#).

Figure 5–2 Sample Workflow Process

About Step Actions

You assign one action to each step in a workflow. Actions are performed by people or by an automated method.

For example, required actions in a workflow to create a user include:

- Initiating the request.
- Enabling or activating the user.

Available actions depend on the workflow type and the action defined in the previous step. For example, the Initiate action is available for only the first step of a workflow.

[Table 5–3](#) lists the actions that you can associate with steps in User Manager workflows.

Table 5–3 Actions Permitted in User Manager Workflows

Workflow Type	Actions
Change Attribute	Request (required) Provide Information Approval Provide Information and Approval Subflow Approval Commit (required) External Action Error Report

Table 5–3 (Cont.) Actions Permitted in User Manager Workflows

Workflow Type	Actions
Create User	Initiate or Self Registration (one of these two is required) Provide Information Provide Information and Approval Approval Subflow Approval Commit Enable or Activate (one of these two is required) Select Groups Delete Error Report External Action
Delete User	Initiate (required) Change Information Disable or Deactivate (one of these two is required) Approval Subflow Approval Change Approval Commit Delete Error Report External Action
Deactivate	Initiate Change Information Approval Change Information and Approval Commit External Action Error Report Deactivate Disable Delete
Reactivate	Initiate Provide Information Approval Provide Information and Approval Subflow Approval Commit External Action Error Report Activate Enable

[Table 5–4](#) lists the actions available in Group Manager workflows:

Table 5–4 Actions Permitted in Group Manager Workflows

Workflow Type	Actions
Change Attribute	Request (required)
	Provide Information
	Approval
	Provide Information and Approval
	Subflow Approval
	External Action
	Commit (required)
	Error Report
Create Group	Initiate (required)
	Provide Information
	Provide Information and Approval
	Approval
	Commit (required)
	Subflow Approval
	Delete
	External Action
Delete Group	Error Report
	Initiate (required)
	Change Information
	Change Approval
	Subflow Approval
	Approval
	Commit (required)
	Delete
	Error Report
	External Action

Organization Manager workflow actions are described in [Table 5–5](#):

Table 5–5 Actions in Organization Manager Workflows

Workflow Type	Actions
Change Attribute	Request (required)
	Provide Information
	Approval
	Provide Information and Approval
	Subflow Approval
	External Action
	Commit (required)
	Error Report

Table 5–5 (Cont.) Actions in Organization Manager Workflows

Workflow Type	Actions
Create Object	Initiate (required) Self Registration Provide Information Provide Information and Approval Approval Subflow Approval Commit Delete Error Report External Action
Delete Object	Initiate (required) Change Information Approval Change Approval Subflow Approval Commit (required) Delete Error Report External Action

Descriptions of Step Actions

[Table 5–6](#) describes the actions available in workflows.

Table 5–6 Workflow Step Actions

Action	Description
Activate	User Manager only. Activates a new user in the Identity System. An activated user is enabled, and can log in and perform operations granted by administrators. The obuseraccountcontrol attribute in the user's entry controls activated/deactivated status. The Activate action requires a participant, such as a manager, to activate the user.
Approval	This action can be configured with required attributes. At run time, the values for the required attributes are presented to the participant for approval. No information can be changed by this action.
Change Information and Approval	Performs the same function as the Provide Info and Approval actions, but used only when deactivating users.
Change Information	Performs the same role as the Provide Info action, but is used only when deactivating users.
Commit	Writes the information collected in the previous steps to the directory. A commit operation writes information to the location of the e object in the directory. For example, during a Create operation, the Commit action adds a new entry to the directory. If the workflow contains additional Commit action, the information is written to the location in the directory that contains the newly created object. A Commit action can be used more than once in a workflow. No user action is required.

Table 5–6 (Cont.) Workflow Step Actions

Action	Description
Deactivate	User Manager only. Deactivation takes effect once the user's current session has ended. A deactivated user cannot log in. Others cannot find a deactivated user in the Identity System except when searching for deactivated users. Deactivating does not delete the user from the directory. The obuseraccountcontrol attribute in user's entry controls activated/deactivated status. A participant is required for a deactivate step in a workflow. Note: To create an .ldif containing deleted users, use the Deactivate or Disable workflow steps instead of Delete. Go to the Deactivated User Identity page, and use the Archive option. This will delete the users from the directory and create a deactivateduser.ldif in the <i>IdentityServer_install_dir\oblix\data\common</i> directory.
Delete	The delete action in a Create User, Group, or Object workflow permanently removes the target entry from the directory. It is possible for a Create workflow to be rejected after a target entry is created. The Delete step cleans up the directory so that new attempts to create the same user can be made.
Disable	User Manager only. Deactivates a user, which means the user cannot be recognized by the Identity System once the user's current session has ended. Deactivation takes effect the next time the user attempts to log in. Deactivating does not delete the object from the directory. This action does not require a participant.
Enable	User Manager only. An Enable action is a combination of a Commit and an Activate action. Automatically activates the new user, who is then recognized by the Identity System after the previous step is completed. This action does not require a person to activate the user.
Error Report	When a background process encounters a processing error, you can configure an error report to send the error to particular users. You can also configure an error report when a step is rejected, for instance during the approval process.
External Action	An action performed outside of Oracle Access Manager.
Initiate	Starts the Create and Deactivate workflows. This action can be used once in a workflow. It must be the first action. The self-registration action can also be the initiating action of a workflow. All users see the Create Profile button or Initiate Deactivate option on their pages, regardless of whether they have been defined as a participant for this particular workflow. If a user clicks the button or link to the workflow but they have not been defined as a participant in the workflow, an error message will be displayed.
Provide Information and Approval	Combines the Provide Info and Approval actions into one action.
Provide Information	Collects information from the user. This action is similar to Initiate, but it cannot be the workflow's first action.
Request	A user's request to change, add, or delete an attribute. Participants for this action see the Request to Modify or Request to Remove button on the Modify Profile page.
Self-Registration	Lets users complete and submit a registration form. Other participants approve the request and activate the user. This action must be the first step in a workflow. The self-registration action does not necessarily require other participants to approve and activate the new user.

Table 5–6 (Cont.) Workflow Step Actions

Action	Description
Select Groups	Enables the workflow participant to subscribe a target user to a group or groups during a create user workflow. The new user has to meet the subscription policy. Available only after an Enable or Activate step.
Subflow Approval	Reports the current status of a subflow that has been invoked from a main workflow step. It does not apply to subflows invoked from other subflows.

Note: Email post-notification for a self-registration step requires two parameters in the `globalparams.xml`, `sendMailFromName` and `sendMailFromEmail`. The values for these parameters are placed in the "mail From" or "senders name" and "mail" or "senders email" parts of the SMTP message respectively.

For self registration, these values are provided through `globalparams.xml` because the target is not yet created. In this case, you need to locate the parameters in the `globalparamams.xml`, then modify the values to suit your environment. For example:

```
<SimpleList>
  <NameValPair
    ParamName="sendMailFromName"
    Value="SelfRegistration"></NameValPair>
</SimpleList>
<SimpleList>
  <NameValPair
    ParamName="sendMailFromEmail"
    Value="SelfRegistration@Oracle.com"></NameValPair>
</SimpleList>
```

If the target user has been created the values for `sendMailFromName` and `sendMailFromEmail` are obtained from the naming attribute and email attribute of logged in user's profile, respectively.

About Subflows

In a simple workflow, all steps execute sequentially. If one step is in a pending state, the workflow does not progress to the next step. Because workflows often involve different participants, this can delay completing the workflow. To speed up processing of a workflow, you may want to define *subflows* that occur in parallel.

Subflows lets you break down workflows into chunks. A subflow can trigger subflows of its own. You can trigger multiple subflows from a single workflow.

Note: A subflow is always a Change Attribute workflow.

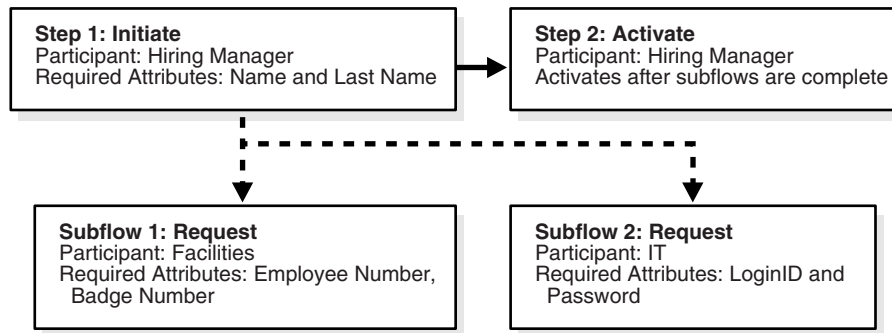
Process overview: A Create User workflow example

1. The hiring manager initiates a Create User workflow.
2. A request is sent to Facilities for an employee number and badge number.
3. A request is sent to IT for the login and password.

4. A request is sent to the hiring manager for final approval and activation of the user.

By using subflows, some of the requests can occur in parallel. The approval waits until the subflows are complete, as illustrated in [Figure 5–3](#).

Figure 5–3 Order of step completion when using subflows



A workflow does not move to the next step until a subflow is complete.

Note: For a subflow to launch, the target object or attribute must meet any filter criteria specified in the Workflow Domain filter.

Using the QuickStart Tool

For Master Administrators, the QuickStart tool enables the rapid creation of simple workflows based on default settings.

After completing a workflow definition using QuickStart, you can use workflow tools to modify the workflow definition, for example, you can specify dynamic participants or surrogates.

The QuickStart tool enables you to define the following workflows:

Table 5–7 Workflows that Can Be Created with the QuickStart Tool

The workflow named. . .	Contains these steps. . .
Create User, Group or Object (Basic)	Self-Registration or Initiate Commit Error Report Note: For a simple Create User workflow, required attributes are Last Name and Name on most directory servers and Login ID if you use Active Directory.
Create User, Group or Object (Advanced: with Approval)	Initiate Approval Commit Error Report

Table 5–7 (Cont.) Workflows that Can Be Created with the QuickStart Tool

The workflow named. . .	Contains these steps. . .
Self Registration for Users or Objects (Advanced: with Approval)	Self-Registration Approval Commit Error Report
Change Attribute (Basic)	Request Commit Error Report
Change Attribute (Advanced: with Approval)	Request Approval Commit Error Report

The QuickStart tool assigns anyone who is logged in to the Identity System as the participant for most steps. For the User Manager, the participant in a Change Attribute workflow is any person who has been assigned the role of Manager. For the Group Manager, any person assigned the role of Group Owner is the participant in the Approval step of a Change Attribute workflow.

Once you create a workflow using the QuickStart tool, you can view and modify the workflow steps, participants, affected attributes, and so on. Information on viewing a workflow definition is provided in "[Viewing and Exporting a Workflow Summary](#)" on page 5-53. Information on modifying a workflow is provided in "[Modifying a Workflow](#)" on page 5-55.

Note: Your ability to define a workflow depends on your administrative privileges.

To define a workflow using the QuickStart tool

1. From the Identity System Console, select the User, Group, or Organization Manager.
2. Click Configuration, then click Workflow Definition.

By default, only Master Administrators, Master Identity Administrators, and Delegated Identity Administrators have permission to view configuration information.

3. Click the link labeled "Click here".

ORACLE Identity Administration

User Manager Group Manager Org. Manager Identity

My Profile | Reports | Create User Identity | Deactivate User Identity | Substitute Rights | Requests | Configuration

Search Full Name That Contains All 8 Results Go Advanced

Attribute Access Control Delegated Administration Workflow Definition Set Searchbase

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, and participants. To create a workflow using QuickStart [Click here](#)

[1 of 3] Workflow Definition

Workflows Create user - Basic [Enabled]

New Modify Copy Delete View Disable Export All

1) Workflow Name

This launches the QuickStart tool.

4. Select the type of workflow you want to create.

Note: You can define a Create workflow and a Change Attribute workflow from the same QuickStart page. Scroll to the bottom of the page to see the Change Attribute fields and options.

You can also provide a name for your workflow. A default name is provided, but it does not change if you use the QuickStart tool to create multiple workflows of this type.

5. If you select a Create workflow type, you can also specify one target location for the object the workflow creates.

The default target location is the searchbase for the Identity System.

6. Optionally, you can select additional attributes.

For a Create User, Create Group, or Create Object workflow, these attributes are entered during initiation or self registration steps.

For a Change Attribute workflow, these attributes are modified when running the workflow. A separate workflow is created for each attribute you select. For example, if you select five attributes, the QuickStart tool generates five change attribute workflows.

7. Click Generate.
8. View the summary report generated by the QuickStart tool.

Workflow Definition - QuickStart (2 of 2)

Summary report

The following workflows have been generated:

Workflow name	Warning (if any)
Create user - Basic	

Done

9. To test the workflow, click a link to one of the workflows on the summary report.

This initiates a workflow instance. For information on the process for using a workflow, see ["How Users Access Workflows in an Identity System Application"](#) on page 5-4.

Note: To use a workflow as a portal insert, copy the resulting URL from your browser. See the *Oracle Access Manager Customization Guide* for more information on creating portal inserts.

10. Click Done.

Creating a Self-Registration Workflow Using the Quickstart Tool

If you want to provide a user registration page for your Web portal, you can define a self-registration workflow and capture the resulting URL of the workflow. This URL can be used as a portal insert.

To define a self-registration workflow using the Quickstart tool

1. Create a self-registration workflow as described in ["To define a workflow using the QuickStart tool"](#) on page 5-17.
2. After clicking the Generate button, click the link for the newly-created workflow.
3. When the new workflow appears, copy the URL.

You can use this URL when you set up the user registration page for your Web portal. This URL is the link to the first page of the workflow. See ["Creating a Self-Registration Workflow"](#) on page 5-62 for other methods of defining self-registration workflows.

Using the Workflow Applet

In addition to using the QuickStart tool, you can define workflows using configuration pages that allow you to specify multiple options and subflows.

You must have permissions to define workflows. See ["About Delegating Administration"](#) on page 2-5 for more information.

Typically, workflows contain at least two steps: one step to initiate the workflow and another step to commit the changes.

Task overview: Defining a workflow using the workflow applet

1. Invoke the Workflow Definition applet.
See ["To access the Workflow Definition applet"](#) on page 5-20 for details.
2. Select New to start creating a new workflow definition.
["Starting a New Workflow Definition"](#) on page 5-20.
3. If you selected a Create workflow type, identify a workflow target.
The target is the location in the directory tree where the object will be created. See ["Defining an LDAP Target for Create Object Workflows"](#) on page 5-22 for details.
4. Define a workflow step and action.
For each step in a workflow, there is an action. Actions are performed by people or by an automated method. You assign one action to each step in a workflow. You

also assign participants to each step. See ["Defining the First Step in a Workflow"](#) on page 5-24 for details.

5. Associate attributes with the step.

Step actions are performed on one or more attribute values. These attributes may be taken from the directory or from an object template. See ["Defining Step Attributes"](#) on page 5-26 for details.

6. Define entry conditions for subsequent steps.

See ["Defining Subsequent Steps"](#) on page 5-29 for details.

7. Define the subsequent steps.

8. Define one or more subflows.

Subflows are conditions that must be satisfied for a particular step or workflow to complete. Like main workflow steps, subflows have associated actions, participants, and attributes. See ["Defining a Subflow"](#) on page 5-32 for details.

9. Define one or more commit steps to end the workflow.

If you are configuring a workflow using more than one schema (for example, an LDAP schema and a template schema), you should configure separate commit steps for each schema type.

Note: Template attribute values can be sent to the back-end system, but these values cannot be read back in to the Identity System for display on profile pages. As a result, to check if a workflow configuration was done correctly and an instance of using the workflow was successful, you may have to examine the data in the back-end system.

To access the Workflow Definition applet

1. From the Identity System Console, select the User, Group, or Organization Manager.

If the Organization Manager has more than one tab, select the appropriate tab.

2. Click Configuration, then click Workflow Definition.

On some browsers, you may receive a prompt asking if you trust the certificate of the application. If you receive this prompt, select the Trust Always option.

For the User Manager and Organization Manager, a Workflow Definition page is displayed.

3. If you are using the Group Manager, indicate the appropriate Group Type, if applicable. The available group types depend on your configuration, as described in ["Adding Auxiliary and Template Object Classes to a Group Tab"](#) on page 4-9.
4. If you are using the Group Manager, from the Workflow Definition page, select an appropriate group type if applicable and click Next.

If you do not select a group type, the Basic group type is used for this workflow.

Starting a New Workflow Definition

You can create workflow definitions for different sets of users. For example, you can define different Create User workflows for Engineering and Sales.

For a simple Create User workflow, required attributes are Last Name and Name on most directory servers and Login ID if you use Active Directory.

Note: By default, the Identity System does not display every object and attribute in the directory. This parameter `excludeOCsForTreeInApplet` in `Identity_install_dir/apps/common/bin/globalparams.xml` enables you to expose object classes in the Identity System applications that would otherwise be hidden.

See the parameter reference chapter in *Oracle Access Manager Customization Guide* for details.

To begin a new workflow definition

1. Invoke the workflow definition tool as described in ["Using the Workflow Applet"](#) on page 5-19.
2. Click New and wait for all buttons except the New button to become deactivated.
3. In the Workflow Name field, enter a name for your workflow.
4. From the Workflow Type list, select the type of workflow you want to create.

For more information on workflow types, see ["Workflow Types, Steps, and Actions"](#) on page 5-7.

If you are creating a subflow, see ["Defining a Subflow"](#) on page 5-32.

5. In the Workflow Description field, you can enter an optional description of this workflow.
6. In the Workflow Domain field, select the starting point in the directory tree from which this workflow is available.

If you want the workflow domain to match a directory entry of the logged in user who initiates the workflow, use substitution syntax. For example, if you want the "ou" of the workflow domain to always match the "ou" of the person who is generating the workflow, you would enter the following:

`(ou=$ou$)`

See ["Substitution Syntax: Returning Targets that Match the DN of the Logged In User"](#) on page 3-25 for examples.

Note: Do not use full LDAP URL while specifying the filter for workflow domain (or target domain) while creating the workflow. Only the LDAP filter is expected.

You can also select a particular domain where the workflow will be available. For instance, if you have different branches in your directory tree for Engineering and Sales, and you want this workflow to only apply to Engineering, you would select the top node for the Engineering branch of the directory tree. If you have a particularly flat directory tree or if the tree has a particularly high number of branches, you can narrow the workflow domain by entering an LDAP filter. See ["Usage of Rules and Filters"](#) on page 3-23. For example, if the starting point in the directory tree is `ou=people`, and you want to create a workflow just for administrators, you may want a filter that contains `(title=admin)`.

Note: Be sure to test performance when using filters. Filters are evaluated at run time, which can affect performance.

7. If you are in User or Organization Manager, click Next.

Depending on your workflow type, you are prompted to select a target as described in ["Defining an LDAP Target for Create Object Workflows"](#) on page 5-22, or you are prompted to define the first step in the workflow as described in ["Defining the First Step in a Workflow"](#) on page 5-24.

8. If you are in the Group Manager and you are working with an Advanced Group, specify the Subscription Type, if applicable.

For example, this might be the case when you define a step for selecting a group or for allowing a user to add themselves to a group. Subscription Type options are available to your participants if the obGroupSubscriptionType attribute was configured for the oblixAdvanced Group object class.

The following subscription types are available:

Table 5–8 Workflow Subscription Types

Option	Description
No type selected	No subscription type is defined. Functionally equivalent to the Open policy.
Open	Enrollment is open to anyone who subscribes.
Open with Filter	Enrollment is open to any user who satisfies the Dynamic Filter (LDAP rule) for the group.
Controlled through Workflow	To subscribe or unsubscribe, the user must be the target of a select group step of a workflow.
Closed	Member list is closed. No changes are allowed. The default setting for the default_subscription policy parameter is SubscriptionPolicyClosed. This is located in <i>IdentityServer_install_dir/identity/oblix/data/common/groupdbparams.xml</i> where <i>IdentityServer_install_dir</i> is the directory where you installed the Identity System.

9. Click Add, then click Next.

Depending on your workflow type, you are either prompted to select a target as described in ["Defining an LDAP Target for Create Object Workflows"](#) on page 5-22, or you are prompted to define the first step as described in ["Defining the First Step in a Workflow"](#) on page 5-24.

Defining an LDAP Target for Create Object Workflows

If you selected Create as the type of workflow you are defining, for example, Create User, you need to define one or more targets. The target is the location in the directory tree where the object will be created. For example, a target of ou=bestmotors,o=company,c=us allows objects to be created under the ou=bestmotors container. When a user is created using a workflow with this target, the directory entry may look like cn=John Smith,ou=bestmotors,o=company,c=us.

You can also use substitution syntax when defining the workflow target. This would ensure, for example, that in a Create User workflow the "ou" entry of the new user would always match the "ou" entry of the logged in user who initiates the workflow. See ["Substitution Syntax: Returning Targets that Match the DN of the Logged In User"](#) on page 3-25 for examples. Note that when using substitution syntax, you may need to modify the ResourceFilterSearchScope parameter value in globalparams.xml. See the *Oracle Access Manager Customization Guide* for details.

If the logged in user has an "ou" entry with multiple values, and you want to provide the ability to create the new user under any of these "ou's," you need to modify the ResourceFilterSearchScope parameter value to 2 in globalparams.xml. See the *Oracle Access Manager Customization Guide* for details. In this case, a list of all the possible targets is shown when the workflow is run. The user can then select the precise "ou" under which the new target user is to be created. The targets in the list are obtained from the multiple values of the "ou" attribute of the logged in user. You can restrict this list can be by having other filter components along with (ou=\$ou\$), such as (objectclass=organizationalUnit).

If you define more than one target, the participant is presented with a selection list when the workflow is run. Workflow targets are always based on the LDAP directory tree. Targets cannot be based on a template schema.

If you are defining another type of workflow, clicking Next on the initial workflow definition page brings you to the step definition page described in ["Defining the First Step in a Workflow"](#) on page 5-24.

Note: The default only displays immediate child nodes of the searchbase. See ["Modifying the Default Searchbase Scope"](#) on page 4-50 for details.

To define a workflow target

1. If you have not already done so, start a new workflow as described in ["To begin a new workflow definition"](#) on page 5-21.
2. From the first Workflow Definition page, click Next.

The targets page appears, displaying fields for selecting characteristics of the target.

3. To define a new target, enter a name in the Target Name field.

For example, if you are creating a target for a dealership, the target name may be Dealer Name.

4. In the Target Domain field, select the location in the directory tree where the object will be created and click Add to add the target domain to the Target(s) field.

When you defined the workflow domain, you selected a branch of the directory tree that the workflow applies to. The target domain is a subset of the main workflow domain. You can use a filter to more closely specify the location for the target (any user object in the tree under the node you select).

Note: Do not use full LDAP URL while specifying the target domain (or workflow domain) while creating the workflow. Only the LDAP filter is expected. For example, cn=Shutterbug Canavan is expected rather than
ldap:///ou=Partners,o=Company,c=US??sub?(cn=Shutterbug Canavan).

See ["Usage of Rules and Filters"](#) on page 3-23 for more information.

Note: If you added a filter for the workflow domain, you cannot specify a filter for the target.

5. Click Add.
6. To apply the workflow to additional targets:
 - Click New.
 - Supply another name and domain.
 - Click Add.
7. When you are done supplying target domains, click Next.

Defining the First Step in a Workflow

After naming the workflow and defining a target, if required, you are prompted to create the first workflow step. You will see a page similar to the following.

Attribute Access Control Delegated Administration Workflow Definition Set Searchbase

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, attributes, and participants. To create a workflow using QuickStart [Click here](#)

Workflow Name : New Employee **Workflow Type :** Create User

Defined Steps: No Steps Specified

Step Properties

Action
Subflows
Attributes
Participants
Out of Office
Escalation
Mail Notification

Select action to be performed' Initiate

Note: On the step definition page, not all tabs may be available. This is because some tabs are not applicable at a particular time. For example, in the default Create User workflow, the Out of Office tab is not available during the Initiate step, when Out of Office notifications are not needed, and during the Enable step, which is performed automatically and does not involve any participants.

To define the first step in a workflow

1. If you have not already done so, start a new workflow as described in ["To begin a new workflow definition"](#) on page 5-21.
2. If you have not already done so, for a Create workflow type, define a target as described in ["To define a workflow target"](#) on page 5-23.
3. From the Select action to be performed list, select an action.

For a Create Object workflow for the User or Organization Manager, the Initiate and Self Registration actions are available.

For a Create Object workflow for the Group Manager, the Initiate action is available.

4. Click Participants.

Most steps require participants to perform an action. The exceptions to this are steps with actions that occur automatically such as Commit and Enable, as well as External Action and the Self Registration action.

Note that for steps that do not require participants, the Out of Office tab is not available.

5. Use any of the following methods to specify participants.

- **Roles:** Note that the role of Anyone refers to any user who is logged in to the Identity System.

Roles are defined in the workflow parameter files gsc_wf_param.xml, usc_wf_param.xml, and osc_wf_param.xml. See ["Customization of Data and Actions in a Workflow"](#) on page 5-60 for details.

Note: If you chose Select Participants to Prenotify in the Select Participants field, do not choose Next Step Participants as the role. Also note that in the commit step for a Group Manager workflow, you should not check owner or member for post notification. There will be no email notifications for owner or member even if they are selected.

Participant roles (roles for people who can process a step) will only work after a commit, enable, or activate step has been completed. The commit, enable, or activate step creates the object's DN from which notification information can be determined.

- **Select Person:** See ["Search Functionality"](#) on page 1-9 for details on using the Selector, and see ["Search Filters for the Object Selector Display Type"](#) on page 3-22 for information on how the Selector can be configured.
- **Select Group:** See ["Search Functionality"](#) on page 1-9 for details on using the Selector, and see ["Search Filters for the Object Selector Display Type"](#) on page 3-22 for information on how the Selector can be configured.
- **Build Filter:** See ["Writing LDAP Filters Using Query Builder"](#) on page 4-29 for information on creating an LDAP filter.

[Attribute Access Control](#)[Delegated Administration](#)[Workflow Definition](#)[Set Searchbase](#)

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, attributes, and participants. To create a workflow using QuickStart [Click here](#)

Workflow Name : New Employee **Workflow Type :** Create User

Defined Steps: [step 1] Initiate

New Modify Delete Step Insert Step

Step Properties

Action Subflows Attributes **Participants** Out of Office Escalation Mail Notification

Select Participants* ☒ Static Participants Available ☐ Static Participants Not Available

Select Role ☒ Anyone

Build Filter Build Filter

Select Person(s) Select User ☒ Channing Haramundanis

Select Group(s) Select Group ☒ Level1Admins

Save Step

- Click Save Step or Save Workflow, or select step attributes as described in the following paragraphs.

Defining Step Attributes

Step actions are performed on one or more attribute values. When you configure a step action, you indicate if certain attribute values are required, and other configuration options. For example, on a Provide Information action, you can specify the mail attribute to ensure that the step participant is prompted to supply an email address.

Defining step attributes consists of the following:

- Selecting the attributes that should be available in this step of the workflow.
- Configuring attribute properties.

For attributes based on an object template (.tpl file), when you configure the attribute in the Identity System Console, it may be helpful to indicate the type of schema that the attribute belongs to. For example, for a workflow that sends information to an application that can set up email accounts for new users, you may want to preface the attribute label with the application name. This will be helpful when users view this attribute. Since the flow of data is one-way for provisioned attributes, the attribute values will not be displayed on the Identity System profile page once the user submits the value. If your users have a question about this, the attribute label will help you determine if this is the expected behavior. See ["Configuring Attributes"](#) on page 3-17 for details.

To select attributes available for a step

1. If you have not already done so, start a new workflow as described in ["To begin a new workflow definition"](#) on page 5-21.
2. If you have not already done so, for a Create workflow type, define a target as described in ["To define a workflow target"](#) on page 5-23.
3. Begin defining a workflow step as described in ["To define the first step in a workflow"](#) on page 5-25.
4. After selecting participants for the workflow step, click Attributes.
5. From the Available Attributes panel, select one or more attributes to associate with the workflow step.

For a Change Attribute workflow, be sure that the topmost selected attribute has already been added to a panel on a profile page. This ensures that a "Request to Modify" button will appear on the appropriate profile page, enabling users to run instances of this workflow.

For information on making multiple selections, see ["Keys for Selecting Multiple Attributes"](#) on page 4-35.

6. Click the right arrow button (➡) to add the selected attributes to the Selected Attributes window.

By default for a Create Object workflow, the attribute that defines the Relative Distinguished Name (RDN) appears in the Selected Attributes window.

By default for a Change Attribute workflow, the attribute you selected as the basis for the workflow appears in the Selected Attributes window.



Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, attributes, and participants. To create a workflow using QuickStart [Click here](#)

Workflow Name : New Employee Workflow Type : Create User

Defined Steps: [step 1] Initiate

New Modify Delete Step Insert Step

Step Properties

Action Subflows **Attributes** Participants Out of Office Escalation Mail Notification

Select Attributes

Available Attributes		Selected Attributes
Post Office Box	➡	Full Name
Postal Address		Employee Number
Postal Code		Employee Type
Preferred Delivery Method		Manager
Preferred Language		Organization
Registered Address		Title
Secretary		Department Number
See Also		Display Name
State		Business Category
Street		Telephone Number
Teletex Terminal Identifier		Room Number
Telex Number		
uid - Login - no s assigned		

Properties

7. Save the step or configure attribute properties, if applicable, as described in the following paragraphs.

Note: You cannot save a workflow until all required attributes (as defined in the object class schema) are configured for the workflow.

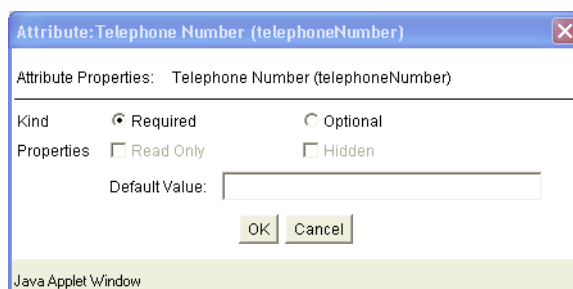
To configure attribute properties

1. From the Selected Attributes window, select one or more attributes that you want to configure.

For information on making multiple selections, see ["Keys for Selecting Multiple Attributes"](#) on page 4-35.

2. Click Properties.

An Attribute Properties dialog appears:



3. Select one or more properties for these attributes:
 - Required: The workflow participant must provide a value for this attribute.

Note: A required attribute cannot be hidden or read-only.

- Optional: The workflow participant may provide a value for this attribute.
 - Read-only: The workflow participant can see but cannot modify the attribute.
 - Hidden: The workflow participant cannot view this attribute value. The attribute is available in the Identity Event Plug-In API and IdentityXML.
 - Default value: Displays a text string. This text string should helpful information for the participant. For example, a string showing the correct format for entering a phone number could be the default value for the phoneNumber attribute. The value is limited to text display types.
4. Click OK.
 5. Click Save Step or Save Workflow.

You can now define Mail Notification participants, or you can define additional steps for this workflow.

Note: When you define Mail Notification participants, if you chose Select Participants to Prenotify in the Select Participants field, do not choose Next Step Participants as the role. Also note that in the Commit step for a Group Manager workflow, you should not check Owner or Member for post notification. There are no email notifications for owner or member even if they are selected.

A commit, enable, or activate step must be completed for a role selected for pre or post notification to work. Before the commit, enable, or activate step is completed, the object exists only in the workflow instance information in the directory tree. The commit, enable, or activate step creates the object's DN from which notification information can be determined.

Note: For information on customizing email notifications, see the *Oracle Access Manager Customization Guide*

Defining Subsequent Steps

A workflow consists of at least an initiating step and a completion step, and may have more steps and subflows. As part of the procedure for creating a second (or third, or more) step in a workflow, you define entry conditions for the step. Entry conditions consist of:

- Identifying what step precedes this one.
- Identifying the required outcome for the previous step.

To define subsequent steps in a workflow

1. After completing the first step in a workflow, as described in "[To define the first step in a workflow](#)" on page 5-25, click New in the Defined Steps area.

New fields appear on this page appropriate for configuring subsequent steps in a workflow.

2. From the Previous Step list, select the step that should precede this action.
3. From the Return Value list, indicate whether the previous step should return a value of True or False in order for this action to execute.

You will want the previous step to return a value of True if it completes successfully. Select False to generate an error report when a previous step returns a value of false. Situations that return a value of false include:

- A participant rejects a workflow ticket
 - The commit step fails
 - The Identity Event Plug-in API or IdentityXML forces a return value of false.
4. Select Wait for Subflows to delay execution of this action until all subflows from preceding steps are complete regardless of their return value.

Selecting this check box appends the return value entry condition with :true. If you do not select the checkbox, the return value entry condition is appended with :false. See "[Defining a Subflow](#)" on page 5-32 for details.

5. Click Add to add the entry conditions to the workflow definition.

The entry conditions appears in the Selected Value field:

- The first entry in this field is a number that corresponds to the number for the step that precedes the current one.
- The second entry in this field is a value of True or False. This entry indicates whether the current step should be run (True) if the previous step completes successfully.

- The last entry in this field is also a value of True or False. This entry indicates whether or not to wait for subflows to complete before running this step. True means wait for subflows to complete.
6. In the Action list, select the action. You may choose enable, activate, and other actions.

The available actions depend on the action in the previous step. Examples:

- Provide Information cannot precede Initiate.
- An error report action usually provides a reason for a step failure. For example, rejection of an attribute value or denying a user activation request.
- Usually a condition of false is the entry condition for an error report step. For example, if a participant in the step prior to the error report step rejects an activation operation, the workflow may proceed to an error report step.

Note: For workflows used for provisioning, you should have at least two commit actions defined for each workflow, one to commit (or enable, or activate, and so on) the data in LDAP, the other to write the provisioning data.

7. As needed, add participants and configure attributes as described in ["Defining the First Step in a Workflow"](#) on page 5-24.
8. Save the step or the workflow.

Note: Create user workflows must end with the Enable step or you will not be able to find users that are added with this workflow.

Committing Workflow Steps

The last step of a workflow commits the data to a particular schema domain. By default, the schema domain is the LDAP directory that the Identity System communicates with. However, if you have configured template attributes in a workflow, you must configure a separate step to commit the data to the schema domain for the template attributes.

Commit steps for attributes in template schema domains can be processed by the Identity Event API and passed to back-end systems for provisioning.

Enabling a Workflow

After you create a workflow, by default it is disabled. When you are ready to allow other participants in the Identity System or external applications, you enable it.

To enable a workflow

1. Access the User, Group, or Organization Manager.
2. Click Configuration, then click Workflow Definition.
3. From the workflows menu, select the workflow.
4. Click Enable.

Note: if you receive a message that attributes are missing, examine the workflow steps. You must configure participants and attributes at each step.

Testing a Workflow

You must enable the workflow before you can test it. See ["Enabling a Workflow"](#) on page 5-30 for details. You must also be a workflow participant to be able to test it.

To test a workflow

1. From the Identity System Console, select the application in which this workflow can be run.

For example, for a Create User workflow, you would open the User Manager.

2. Initiate the workflow.

For example, if you defined a Create User workflow, the workflow that you want to test should appear in a list on the Create User page. To initiate the workflow, you select from the list.

For Change User workflow, you would select the Request to Modify function in the User Manager, and so on.

3. Perform the functions indicated in the workflow steps.

The workflow should behave as expected. For example, after completing the Create User workflow, you should be able to find the user that was added during the Create User operation.

To run a workflow in the Group Manager

1. From the Identity System Console, select Group Manager.
2. Select the group type panel corresponding to the group type in the workflow definition.

For example, you may have different group type panels for the structural group object class and the oblixAdvancedGroup object class. See ["Adding Auxiliary and Template Object Classes to a User or Org. Manager Tab"](#) on page 4-8 for details.

Example of Defining a Workflow

The following is an example of defining a Create User workflow. In this example, you define a workflow that allows anyone who is logged in to the Identity System to create a user. This workflow generates a ticket requesting a name and email address to be provided for this user. When processed, the new user is enabled in the Identity System.

To create this workflow

1. From the User Manager, Click Configuration, then click Workflow Definition to go to the Workflow Definition page.
2. Click New.
3. Name this workflow Test New User Creation Workflow.
4. In the Workflow Type field, select Create User.
5. On the Target DN page, enter a name in the Target Name field and accept the default domain by clicking Add.
6. Click Next to proceed from the Target Domain page to the Workflow Definition Page.
7. Create an Initiate step for the workflow.

Click the Participants tab and define the participants to be the role of Anyone.

Click the Attributes tab and select the attributes that you want workflow participants to provide, for example, First Name and Last Name.

8. Click New to create a new step with the Enable action type.

Click Add to add the Initiate step as an entry condition for the Enable step.

Click Participants and select Anyone as a participant.

Click Attributes and add an attribute that is presented during this step.

9. Save and enable the workflow.

10. Test the workflow by trying to create a new user in the User Manager.

Defining a Subflow

Only the Change Attribute workflow type can be a subflow. These workflows must also be explicitly configured as subflows in the workflow definition page. All subflows must also contain an approval step action.

Note: You must select Use as Subflow on the first page of the workflow definition for a workflow to be usable as a subflow.

To create a subflow

1. From the Identity System Console, click the User Manager, Group Manager, or Org. Manager application tab.
2. Click Configuration.
3. Click Workflow Definition.
4. Click New.
5. In the Workflow Name field, enter a name for your workflow.
6. Click the Use as Subflow checkbox.
7. From the Workflow Type list, select Change Attribute.
8. Click Next.
9. In the first step of the workflow, specify the attribute that the workflow is to change.
10. Complete the rest of the workflow as you would any other workflow.

Note: All subflow definitions must contain an approval step action.

Associating a Subflow with a Workflow

You must associate a subflow with a specific workflow step in the main workflow. During workflow runtime, any subflows configured for a specific step will be invoked after the step action has executed.

To associate a subflow with a workflow

1. Invoke the workflow application, as described in ["To access the Workflow Definition applet"](#) on page 5-20.

2. Select a workflow to which you want to assign a subflow.
3. Click Modify.
The page refreshes and shows the step definitions page.
4. Click the Subflows tab.
5. In the Defined Steps area of the page, select the place in the workflow sequence where you want to insert a subflow.
6. In the Select Subflows area of the Subflows tab, select the subflow that you want to be a part of this workflow.

Select the subflow(s) you want to assign to this step and click the right arrow button (>>).

Note: If you do not see your subflow here, verify that it is marked as a subflow, and it is enabled. The attribute that is the target of the subflow cannot also be used in the workflow to which the subflow is assigned.

7. Save the step.
8. On the subsequent step(s), you may optionally indicate whether or not you want to wait for subflows to complete.
 - a. Select the step that should be delayed until the subflow is complete and click Modify.
 - b. Click the Wait for Subflows checkbox.

Approving Subflow Steps

The Subflow Approval step reports on the current status of subflows triggered from the main flow. By default, the status is set to Approved or Rejected during either an Approval step or a Provide Approval step. This step also allows for the configuration of attributes.

Note: You can set the subflow status programmatically using Identity Event Plug-in API or IdentityXML. See the *Oracle Access Manager Customization Guide* for information on the Identity Event Plug-in API.

Advanced Workflow Ticket Routing

Ordinarily, the static participants you specify when you create a workflow step are the users responsible for completing that step. To avoid processing bottlenecks or to ensure that each ticket reaches the participant most appropriate to process it, the following three advanced ticket routing features enable the replacement of static participants under specific circumstances:

- Instead of specifying static participants when you create a workflow, you can have a workflow plug-in or application choose dynamic participants according to runtime conditions.
- If a static or dynamic participant is going to be out of the office or otherwise unable to process workflow tickets, he or she can set an Out of Office flag in his or

her user profile so that all incoming tickets are redirected to a surrogate participant for as long as the flag remains activated.

- If the participants receiving a given workflow ticket fail to process it within a specified interval, that ticket can be sent to an escalation participant, who assumes full responsibility for the ticket.

Configuring Workflow Actions for Advanced Ticket Routing

Not all workflow steps can be configured for advanced ticket routing. For instance, the first step in a workflow can never be rerouted. Fortunately, it is never necessary to reroute the first step, because the person who initiates the workflow is also the participant for the first step, which is always workflow initiation.

Steps that do not involve user action cannot be rerouted since, by definition, they do not involve user participants. For instance, a step involving the automatic retrieval of provisioning data from an external database involves no human participants, and therefore, participant replacement is moot.

The following table lists the user actions that can be associated with workflow steps:

Table 5–9 User Actions Available for Advanced Ticket Routing

User Action	Availability
Approval	
Provide information with approval	Available by default for dynamic participants, surrogates, and time-based escalation
Initiate	
Self registration	
Provide information	
Subflow approval	Available by default for dynamic participants and surrogates; can be enabled for time-based escalation by modifying the appropriate workflow parameter file. For details, see "To modify the workflow parameter files" on page 5-45.
Activate	
Deactivate	
Error report	
Select groups	
Requests	
Change information	
Change information with approval	

About Notifying Newly Assigned Step Participants

The Mail Notification tab in the workflow applet enables you to configure email notification for participants who are assigned tasks to complete when workflow tickets are rerouted. You can configure mail notification for each step to which ticket rerouting can apply.

To configure mail notification for any step involving advanced ticket rerouting, complete the following procedure:

To configure email notification for advanced workflow ticket rerouting

1. As appropriate for the workflow you are modifying, navigate to the User, Group, or Organization Manager, point to Configuration, then click Workflow Definitions.
2. Select the Workflow you wish to modify, then click Modify.
3. If you are modifying a Change Attribute workflow, click Next once. For any other type of workflow, click Next twice.
4. Select the step for which you wish to set mail notification, then click Modify.
5. Click the Mail Notification tab.
6. To enable notification for static, dynamic, and surrogate participants, click Select Participants to Prenotify or Select Participants to Postnotify.
You cannot select Participants to Prenotify for the first step of a workflow.
7. Specify by Person, Group, Role, or Rule, the users to notify.
You can use the Selector or the Filter Builder as the mechanism for specifying who should be notified.
8. If you also need to notify escalation participants, click Select Escalation Notifees, then repeat step 7.
9. Click Save Step to commit your step-specific changes.
10. Click Save Workflow to save the entire workflow.

Specifying Dynamic Participants

The dynamic participants feature is one of the advanced workflow options that enable the automatic routing of workflow tickets to alternate participants, as determined by circumstances at runtime.

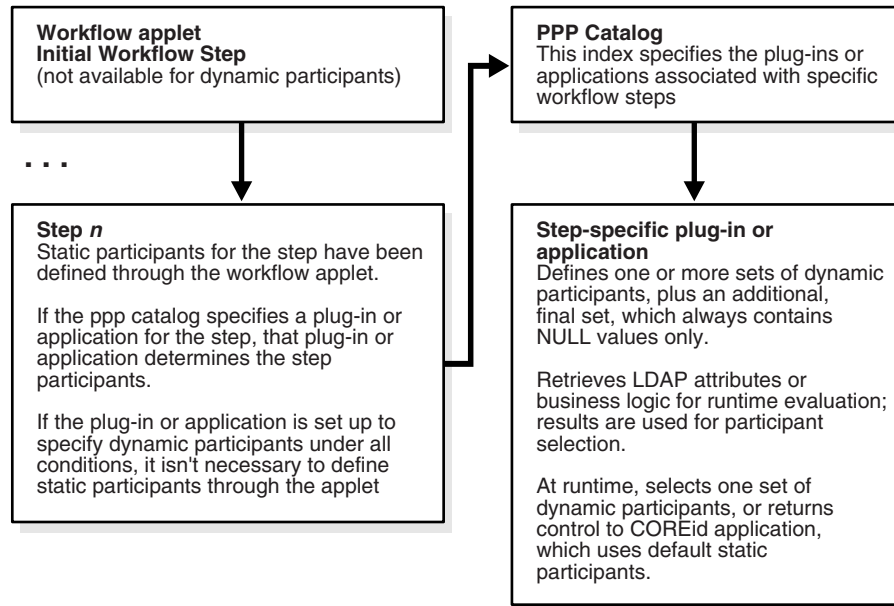
About Workflow Participants

the Identity System supports the following two types of workflow participants:

- *Static participants* are specified through the workflow applet, usually when you define a workflow step. For example, when you create a workflow to set up network accounts for new employees, you can include an approval step that routes all incoming tickets to the network security manager. Unless you later add a workflow plug-in or application, this pre-designated static participant serves as the primary recipient for all tickets generated by this workflow step.
- *Dynamic participants* must be specified in a workflow plug-in or application, rather than through the workflow applet. These conditional participants are selected based on runtime attribute values or external business logic. For example, your plug-in can specify that all purchase requests greater than \$2,500 go to the accounting manager, all requests smaller than \$50 go to the petty cash clerk, and all other requests go to any available staff accountant.

About Workflow Ticket Routing

As the following graphic illustrates, when a workflow is run and program execution reaches a step that has been enabled for dynamic participants, a workflow plug-in or application selects a set of dynamic participants and sends them a workflow ticket. In cases where the plug-in or application does not select any dynamic participants, the calling application sends the ticket to the original static participants, just as if the workflow plug-in or application had never intervened.



About Dynamic Participants

You designate dynamic participants using the same criteria you use to specify static participants. This includes specification by person, by group, by role, and by rule. For details, see the procedure step ["Use any of the following methods to specify participants."](#) on page 5-25.

You can define dynamic participants for every step in a workflow, except for the first step, which must be initiated by a static participant.

When step instances are assigned at runtime, dynamic participants inherit the same ticket-processing rights that static participants would normally receive. These rights extend only to tickets specifically assigned to the dynamic participant. In other words, the dynamic participant does not receive all the rights assigned to the original participant, as would be the case if the Substitute Rights feature were used to create a delegate. See ["Adding Substitute Administrators"](#) on page 2-10.

Since the identity of the dynamic participants is not known until the associated workflow step is run, they are not available for certain workflow services, such as post-action email generated by the previous workflow step. However, it is possible to notify dynamic participants through pre-action email generated by the workflow step that selects them. See the procedure ["To prepare a workflow step for dynamic participants"](#) on page 5-38.

About Static Participants

Under normal circumstances, workflow steps use static participants, whom you designate through the workflow applet.

In cases where the oblixpppcatalog catalog file specifies a plug-in or application for a given step, the plug-in or application receives the first opportunity to select dynamic participants according to the run-time values it evaluates. Only if the plug-in or application declines to specify a set of dynamic participants does control pass back to the main application, where the static participants are specified as the primary step participants.

About the Static Participants Not Available Button

If you know in advance that a workflow plug-in or application will always select dynamic participants for a given step, you don't have to define static participants for that step. Remain aware, however, that if you do elect not to specify static participants, you must toggle the default Static Participants Available button to Static Participants Not Available. As the following graphic illustrates, these radio buttons appear on the Participants tab of the workflow applet, which is accessible through the User, Group or Organization Manager, as appropriate to the workflow you are configuring.

The screenshot shows the 'Workflow Definition' applet interface. At the top, there are four tabs: 'Attribute Access Control', 'Delegated Administration', 'Workflow Definition' (which is active), and 'Set Searchbase'. Below the tabs, the 'Workflow Name' is 'test' and the 'Workflow Type' is 'Change Attribute'. The 'Defined Steps' section contains a text box with '[step 1] Request' and buttons for 'New', 'Modify', 'Delete Step', and 'Insert Step'. Below this is the 'Step Properties' section with tabs for 'Action', 'Subflows', 'Attributes', 'Participants' (which is active), 'Out of Office', 'Escalation', and 'Mail Notification'. Under the 'Participants' tab, there is a 'Select Participants*' section with two radio buttons: 'Static Participants Available' (which is selected) and 'Static Participants Not Available'.

Enabling Dynamic Participants

The dynamic participants feature is not enabled by default. To activate this feature, you must complete the procedures listed in the following task overview:

Task overview: Assigning dynamic participants to a workflow step

1. Use either the QuickStart tool or the workflow applet to create a workflow containing the steps that will utilize dynamic participants.

See ["Using the QuickStart Tool"](#) on page 5-16 and ["Using the Workflow Applet"](#) on page 5-19 for details.

2. If any possibility exists that a given step will use static participants, you must define static participants for that step as described in ["Use any of the following methods to specify participants."](#) on page 5-25.

On the other hand, if you know in advance that a given step will always use dynamic participants, you don't need to define static participants for that step. See ["About the Static Participants Not Available Button"](#) on page 5-37 for details.

3. If you wish, set up email notification for dynamic participants.

See ["To prepare a workflow step for dynamic participants"](#) on page 5-38 for details.

4. Add a pointer to the oblixpppcatalog catalog file so that the proper plug-in or application is called at runtime to select dynamic participants.

See ["To modify oblixpppcatalog.lst"](#) on page 5-39 for details.

5. Create a pre-action plug-in or application to select dynamic participants at runtime.

A typical plug-in or application might contain code sections to perform the following:

- Specify at least three sets of dynamic participants, the last of which always contains NULL values only
- Specify attributes or business logic to be evaluated at run time
- Select dynamic participants based on the evaluation
- Ensure that the selected participants actually exist in the Identity System directory; otherwise, the dynamic participant selection process will fail, but the workflow engine will not return an error message
- Pass the list of dynamic participants back to the calling the Identity System application

See ["Task overview: Creating a plug-in or application to select dynamic participants"](#) on page 5-40 for details.

WARNING: Plug-ins and applications that enable dynamic participants must be of type "pre-action." They can never be of type "post-action."

To prepare a workflow step for dynamic participants

1. In the User Manager, Group, Manager, or Organization Manager, navigate to Configuration, then click Workflow Definitions.
2. From the list marked Workflows, select the workflow containing the step being prepared for dynamic participants, then click Modify.
3. If any possibility exists that the current workflow step can ultimately use static participants, define static participants by Role, Rule, Persons, or Group, as described in ["Use any of the following methods to specify participants."](#) on page 5-25.

If the preceding condition does not apply, or you previously defined static participants for this step, proceed directly to step 4.
4. If the current workflow step will use a plug-in or application to specify dynamic participants, and there are no conditions under which static participants can ultimately receive the workflow ticket for the current step, activate the Static Participants Not Available button, as described in ["About the Static Participants Not Available Button"](#) on page 5-37. Otherwise, proceed directly to step 5.
5. If you wish, set up email notification by clicking the Mail Notification tab, clicking the Select Participants to Prenotify button, then selecting Current Step Participants in the Select Role box. If they are ultimately selected, the dynamic participants will receive e-mail announcing that workflow tickets have been assigned to them.

Note: The Select Participants to Prenotify button turns on email notification for static participants when the Static Participants Available switch is active. By contrast, it sends notifications to dynamic participants when the Static Participants Not Available switch is active.

6. Commit the step-specific information by clicking Save Step and then clicking OK to dismiss the prompt that asks you to confirm the operation.
7. Commit the information pertaining to the entire Workflow by clicking Save Workflow and then clicking OK to dismiss the prompt seeking to confirm the

operation. If an additional prompt asks whether you want to enable the workflow, click Yes.

To modify oblixpppcatalog.lst

1. Complete the following sub-steps to determine the workflow ID of the workflow containing the step for which you wish to set dynamic participants:

- a. Launch the User, Group, or Organization Manager, as appropriate for the workflow you wish to modify.
- b. Click the Configuration tab, then click Workflow Definitions.
- c. Select the workflow you wish to modify, then click View.
- d. Make a note of the value reported in the Workflow Definition View for obworkflowid, which will appear in a string similar to the following:

Workflow DN : obworkflowid=5985de47196a4a728a629a429b6a5194

2. Use any plain-text editor to open the file oblixpppcatalog.lst, which is located in the following directory:

IdentityServer_install_dir\identity\oblix\apps\common\bin

where *IdentityServer_install_dir* is the root installation folder for the Identity Server that runs your workflow.

3. Add one of the following strings to oblixpppcatalog.lst:

```
obworkflowid_workflowstep_preaction;lib;;
Component_install_dir\identity\oblix\path\pluginName.dll;
functionName;
```

Or

```
obworkflowid_workflowstep_preaction;exec;;
Component_install_dir\identity\oblix\path\applicationName.exe;
functionName;
```

where:

- *obworkflowid* is the workflow identification number you noted in step 1d of this procedure
- *workflowstep* is the step for which you wish to define dynamic participants
- *path* is the path under *Component_install_directory*\identity\oblix\ leading to pluginName.dll or applicationName.exe, which are the code objects that select the dynamic participants at runtime
- *functionName*—If you specify a plug-in, you must also specify functionName, which is the function within the dynamic link library plug-in that sets the criteria for the dynamic participants.

You insert the first line of code if you are using a plug-in to specify the dynamic participants; you insert the second line if you are using an executable program, instead of a plug-in.

In any case, the line you insert must end with a semi-colon, and you may place it anywhere in the oblixpppcatalog.lst file, as long as that line does not interrupt an existing line.

The line you insert should be similar to the following:

```
wfqs20040901T17251953156_2_preaction;lib;;
Component_install_dir\identity\oblix
\unsupported\ppp\ppp_dll\ppp_dll.dll;
WorkflowPreActionSetDynamicParticipantsTest;
```

4. Save oblixpppcatalog.lst in its original location.

See also *Oracle Access Manager Developer Guide* for details about the oblixpppcatalog.lst file.

Task overview: Creating a plug-in or application to select dynamic participants

1. Use C++ to create a plug-in or application that specifies the dynamic participants selected when program execution reaches the workflow step you specified in "[To modify oblixpppcatalog.lst](#)" on page 5-39.
2. You can use various conjunctions of LDAP attributes or proprietary business logic to specify the conditions under which one group of dynamic participants is chosen over the others at runtime.
3. Include in the plug-in or application, the following header files, which enable the pre-action processing necessary for dynamic participant selection:

```
obppp.h
obpppwf.h
obpppdata.h
```

4. Within the application or plug-in, define three or more sets of dynamic participants using any combination of roles, rules, persons, or groups. The final item in each array must always be NULL. For instance:
 - a. If you wish to specify persons, insert lines similar to the following:

```
PPPSetsVals[0] = "cn=Van Oman, ou=Sales, ou=Dealer1k1,
ou=Latin America, ou=Ford, o=Company,c=US";
PPPSetsVals[1] = "cn=Fabien Esser, ou=Sales, ou=Dealer1k1,
ou=Latin America, ou=Ford, o=Company,c=US";
PPPSetsVals[2] = NULL;
data->Set("DynamicParticipant.Persons", PPPSetsVals);
```

- b. If you wish to specify groups, insert lines similar to the following:

```
PPPSetsVals[0] = "cn=Basic group1k1, ou=Groups, ou=Dealer1k1,
ou=Latin America, ou=Ford, o=Company,c=US";
PPPSetsVals[1] = "cn=Basic group1k2, ou=Groups, ou=Dealer1k1,
ou=Latin America, ou=Ford, o=Company,c=US";
PPPSetsVals[2] = NULL;
data->Set("DynamicParticipant.Groups", PPPSetsVals);
```

- c. If you wish to specify roles, insert lines similar to the following:

```
PPPSetsVals[0] = "ob_self";
PPPSetsVals[1] = "manager";
PPPSetsVals[2] = NULL;
data->Set("DynamicParticipant.Roles", PPPSetsVals);
Remember, of course, that only certain roles are valid for particular
workflow types. See page 200.
```

- d. If you wish to specify rules, insert lines similar to the following:

```
PPPSetsVals[0] = "(cn=rohit*)";
PPPSetsVals[1] = "(cn=beth*)";
```

```

PPPSetsVals[2] = NULL;
data->Set("DynamicParticipant.Rules", PPPSetsVals);

```

Specifying Surrogates

You can configure the Identity System applications so that when a static or dynamic participant is not available to perform the actions assigned for a particular workflow step, that participant can set an Out of Office flag in his or her user profile, causing incoming workflow tickets to go to one or more designated surrogate participants. The surrogate is granted whatever rights the original participant had to process the rerouted tickets.

Only tickets created after activation of the Out of Office flag are rerouted to the surrogate. The original participant retains responsibility for processing all tickets created prior to activation of the Out of Office flag.

When the Out of Office flag is turned on, it applies to all of the steps in all of the workflows for which the participant has been designated as a static participant or potential dynamic participant.

When the Out of Office flag is reset to Off, newly created tickets are once again routed to the original participant. The surrogate retains responsibility for processing all tickets routed to him or her while the Out of Office flag was active, but no new tickets are routed to the surrogate unless the original participant once again activates his or her Out of Office flag.

The same workflow applet setting that sends ticket assignments to original participants also notifies surrogates and others that the workflow ticket has been rerouted because of the Out of Office flag.

Task overview: Enabling surrogates

1. Associate the attribute of your choice with the Out of Office semantic type through the Common Configuration tab of the Identity System Console.

You only need to do this once. See ["To associate an Out of Office attribute with the Out of Office semantic type"](#) for details.

2. Specify one or more surrogates through the Out of Office tab in the workflow applet.

See ["Use any of the following methods to specify participants."](#) on page 5-25 for details.

3. Individual users activate their Out of Office flags in their User Profiles.

See ["To make use of the Out of Office flag"](#) on page 5-43 for details.

To associate an Out of Office attribute with the Out of Office semantic type

1. Choose an attribute in the LDAP directory to associate with the Out of Office semantic type.

It must be an attribute with a boolean value that indicates whether the user is in or out of the office. For convenience, Oracle supplies the attribute `obOutOfOfficeIndicator`, but you can use any suitable attribute in your directory.

2. Navigate to Identity System Console, select Common Configuration, select Object Class, select the person object class (for example, `gensiteorgperson`), then select Modify Attributes.

3. From the attribute list, select the attribute you wish to associate.

4. In the Semantic Type field, select "Out of Office - Indicator."

Only some attributes will have this indicator. For example, for gensiteOrgPerson, the genuserid attribute will have this indicator.

ORACLE Identity Administration Help About Logout

User Manager Group Manager Org. Manager Identity System Console

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration

Logged in user: Rohit Valiveti

Object Classes

- Workflow Panels
- Master Audit Policy
- Global Auditing Policies

Modify attributes

Through Modify attributes, you can modify or configure the display name, semantic type, display type, and attribute value(s) for the attributes in the gensiteorgperson object class. After modification, please click done button to save the attribute information.

Attribute	genPhoto genSiteDn genSiteUid genType genUserID givenName homePhone	Display Name	genUserID
Data Type	String(Case-insensitive)	Semantic Type	Group Dynamic Member Location Coordinates Out Of Office - Indicator Photo
Display Type	Boolean	Attribute Value(s)	☒ Single ☐ Multiple

No Display Type Properties Available

5. In the Display Type box select Boolean, then click Done to commit the change.

Note: This procedure only needs to be performed once.

To specify a surrogate

1. As appropriate to the particular workflow containing the step for which you wish to specify one or more surrogates, log onto the User, Group, or Organization Manager, then navigate to Configuration, then select Workflow Definitions.
2. Select the workflow containing the step for which you wish to specify surrogates and click Modify.
3. Click Next once if you are modifying a Change Attribute workflow.
Click Next twice if you are modifying any other type of workflow.
4. Select the step for which you wish to specify surrogates, then click Modify.

The page is refreshed with information about this step. If the page does not refresh, be sure that the step is highlighted. If it is not, click it again, then click Modify.

You can specify a surrogate for any workflow step associated with a user action.

5. Click the Out of Office tab.
6. Specify one or more Surrogates using any combination of the Person, Group, Role, and Rule tools.

See ["Use any of the following methods to specify participants."](#) on page 5-25 for details.

The Select Indirect Roles box provides check boxes to select whatever roles are currently defined in your directory. These roles are considered indirect, because they apply to the current participant, rather than the workflow target.

7. Click Save Step to commit the changes for that step.

If you receive a warning to configure attributes, make sure that you have selected attributes for this step.

8. Repeat the preceding steps for each workflow step for which you wish to specify a surrogate.
9. Click Save Workflow to save the entire workflow.

To make use of the Out of Office flag

1. Verify that you, as a potential static or dynamic user, have been granted sufficient privileges (search, read, and write) to perform the operations described in this procedure.
2. Verify that the Out of Office flag has been configured for the attribute.
3. Navigate to User Manager, select My Profile, then click Modify.
4. In the Personal Information section, toggle the Out of Office Indicator to True. (This attribute is False, by default).
5. Click Save to commit the change, then click OK to dismiss the pop up that seeks to confirm the operation.

Enabling Time-based Escalation

If the participant or participants assigned to process a workflow ticket do not do so within a specified interval, you can have the Identity System escalate the ticket by rerouting it to a different participant. The original participant can no longer process the escalated ticket: it must now be processed by the escalation participant, who receives all rights previously given to the original participant for processing the ticket.

If the escalation participant does not process the ticket within the allotted time, the ticket is escalated again, and so on, until it is escalated to the Identity System administrator, who is the last possible participant to be assigned responsibility for the escalated ticket.

By default, you can enable time-based escalation on any workflow step, provided the following two conditions hold true:

- The escalated step is not the initial step in the workflow
- The action associated with the step has been enabled for escalation. By default, only Approval and Provide Information and Approval are enabled for escalation, but you can enable other actions by adding lines to the appropriate workflow parameter file. See the procedure ["To modify the workflow parameter files"](#) on page 5-45 for details.

To enable time-based escalation

1. As appropriate to the particular workflow for which you wish to set up time-based escalation, log onto the User, Group, or Organization Manager, then navigate to Configuration, then click Workflow Definitions.
2. Select the workflow for which you wish to set up escalation and click Modify.

If a pop up appears to warn you that only certain settings can be modified while pending tickets exist for the workflow, dismiss it by clicking OK. If you are modifying a Change Attribute workflow, click Next once; if you are modifying any other type of workflow, click Next twice.

3. Highlight the step for which you want to enable escalation, then click Modify.

The page is refreshed with information about this step. If the page does not refresh, be sure that the step is highlighted. If it is not, click it again, then click Modify.

The step you select must be associated with an action that is enabled for escalation. By default Approval and Provide Information and Approval are enabled. To enable additional actions to support time-based escalation, see the procedure "[To modify the workflow parameter files](#)" on page 5-45 for details.

4. Click the Escalation tab.
5. Specify the interval after which the ticket will be escalated. You can set the interval in days, minutes, or hours. This interval applies to all escalation levels.

ORACLE Identity Administration

My Profile | Reports | Create User Identity | Deactivate User Identity | Substitute Rights | Requests | **Configuration**

Search: Full Name That Contains All 8 Results Go Advanced Logged in User: I

Attribute Access Control Delegated Administration **Workflow Definition** Set Searchbase

Workflow Definition

Workflow Name : time-based escalation Workflow Type : Create User

Defined Steps:

- [step 1] Initiate
- [step 2] Provide Information and Approval
- [step 3] Enable

New Modify Delete Step Insert Step

Step Properties

Action Subflows Attributes Participants Out of Office **Escalation** Mail Notification

Idleness interval for escalation 5 Day(s)

Number of escalation levels 1 levels

Route to participants

Select Indirect Role ☒ Manager ☐ Secretary

6. Specify the participant or participants to whom the ticket will be escalated. Roles are indirect in the sense that they are evaluated not with respect to the workflow target, but with respect to the participant who did not process the ticket in time to prevent the most recent escalation. For instance, if Manager is checked in the Select Indirect Roles box, and the accountant who initially receives the ticket does not process it quickly enough, the ticket is escalated to that accountant's manager (and not the manager of the workflow target).
7. Specify the number of times (levels) a ticket can be escalated. This does not include the final escalation level, which always routes the ticket to the Identity System administrator.

You can only specify one set of escalation participants. This single set applies to all escalation levels. If you specify a unique user, for example, the ticket is escalated to that person each time escalation is triggered. If that escalation participant does not process the ticket at any level, the ticket is ultimately escalated to the Identity System Manager.

If, on the other hand, you specify a role that is held by a different person at each level, the ticket will be escalated to a different person at each level. For instance, if you specify Manager, the ticket will be escalated to the manager of the person to

whom the ticket was originally issued, then to the manager of that manager, and then to that manager's manager, and so on.

8. Click Save Step to commit the setting you have entered on the escalation tab.
9. Specify the people who will be notified about the escalation by clicking the Mail Notification tab.

ORACLE Identity Administration

My Profile | Reports | Create User Identity | Deactivate User Identity | Substitute Rights | Requests | **Configuration**

Search: Full Name That Contains All 8 Results Go Advanced Logged in User: I

Attribute Access Control Delegated Administration **Workflow Definition** Set Searchbase

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, and participants. To create a workflow using QuickStart [Click here](#)

Workflow Name : time-based escalation Workflow Type : Create User

Defined Steps:

- [step 1] Initiate
- [step 2] Provide Information and Approval
- [step 3] Enable

New Modify Delete Step Insert Step

Step Properties

Action Subflows Attributes Participants Out of Office Escalation **Mail Notification**

Select Participants:

- ☒ Select Participants to PreNotify
- ☐ Select Participants to PostNotify
- ☐ Select Escalation Notifies

10. Click Select escalation notifiers
11. Select the people to be notified by Person, Group, Role, or Rule. The available roles are the following:
 - **Previous step owners:** This is the participant who completed the previous step.
 - **Current step participants:** These are the people currently assigned to process the just-escalated ticket.
 - **Next step participants:** These are the people who will be assigned to process the next step. Only the static participants defined for the next step can be notified, since the email notifications are sent before the execution flow reaches the next step, and the identity of the dynamic participants is determined.
 - **Initiator:** This is the user who initiated the work flow.

To modify the workflow parameter files

1. Complete this procedure only if you want to enable time-based escalation for a user action other than Approval or Provide Information and Approval. For a list of the user actions that can be enabled for time-based escalation, see [Table 5-9](#).
2. Using any plain text editor, launch the workflow parameter file appropriate to the workflow containing the actions for which you are enabling time-based escalation.

[Table 5-10](#) lists the workflow parameter files that apply to workflows associated with the various Identity System applications:

Table 5–10 Workflow parameter file names and paths

Application	Workflow parameter file name and path
User Manager	<i>IdentityServer_install_dir</i> /identity/oblix/apps/userservcenter/bin/usc_wf_params.xml
Group Manager	<i>IdentityServer_install_dir</i> /identity/oblix/apps/groupservcenter/bin/gsc_wf_params.xml
Organization Manager	<i>IdentityServer_install_dir</i> /identity/oblix/apps/objservcenter/bin/osc_wf_params.xml

3. Locate the compound list for the action you want to support time-based escalation. The first half of this compound list should resemble the listing in [Example 5–1](#):

Example 5–1 The workflow parameter compound list (partial listing only)

```
<CompoundList ListName="actionName">
  <SimpleList >
    <NameValPair ParamName="occurrence" Value="n"/>
    <NameValPair ParamName="useraction" Value="true"/>
    <NameValPair ParamName="initialStep" Value="false"/>
    <NameValPair ParamName="time_based_escalation" Value="true"/>
  </SimpleList>
  . . .
</CompoundList>
```

where *actionName* is the name of the action for which you want to enable time-based escalation. For a list of actions that can be enabled to support time-based escalation, see [Table 5–9](#).

4. Add the following string in the position indicated by the preceding listing:


```
<NameValPair ParamName="time_based_escalation" Value="true"/>
```
5. Repeat the preceding steps for all the user actions you want to support time-based escalation.
6. Save and exit, the file.

Performing Asynchronous Operations

An asynchronous workflow moves from step to step without completing pending subflows. An asynchronous operation is pre and post processing code that is part of an Identity Event, as described in the *Oracle Access Manager Developer Guide*. The *asynch_user* parameter determines who may resume the pending asynchronous action. The default is Anyone.

To allow a user to perform an asynchronous operation

1. Open the *asynchparams.xml* file in:

```
IdentityServer_install_dir/oblix/apps/asynch/bin/asynchparams.xml
```

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

2. Set the *asynch_user* parameter to one of the following:

- Anyone: Anyone can perform asynchronous operations (default).
- DN: A particular user can perform asynchronous operations. Provide the DN of a user.

Only one DN can be accepted by the parameter.

3. Close the asynchparams.xml file.

Notes on Asynchronous Workflows

The User, Group, and Organization Managers are not automatically loaded when an asynchronous workflow is resumed. If there is a request to the Identity Server to resume an asynchronous workflow when the application is not loaded, the workflow engine may not register error conditions.

For example, suppose you create a Deactivate User workflow in the User Manager. This workflow only has Initiate and Disable steps. Suppose also that you create an event plug-in for the workflow that returns STATUS_PPP_WF_ASYNC code to make the workflow instance become asynchronous during the Initiate step, pending a command to instruct the workflow to resume and run the Disable step. If there is an IdentityXML request to resume this workflow while the Identity System is being restarted, the workflow engine would mistakenly return success. The Disable step would return with a status of complete when in fact the user was not disabled.

Note: Be sure the User Manager, Group Manager, and Organization Managers are pre-loaded when the Identity Server is restarted.

To preload the User, Group, and Organization Managers

1. Open the Identity System parameter file:

```
Identity_install_dir/identity/oblix/engine/obengineparams.xml
```

2. In this file, find the configuration information for the Identity System applications:

- <ValNameList ListName="groupservcenter">
- <ValNameList ListName="userservcenter">
- <ValNameList ListName="objservcenter">

3. Change the Dll_Load parameter from 0 to 1 as in following example for Group Manager.

```
<ValNameList ListName="groupservcenter" >
<NameValPair ParamName="Dll_Name" Value="groupservcenter"/>
<NameValPair ParamName="Dll_Dir" Value="oblix/apps/groupservcenter/bin"/>
<NameValPair ParamName="Dll_Load" Value="1"/>
<NameValPair ParamName="Work_Dir" Value="oblix/apps/groupservcenter/bin"/>
```

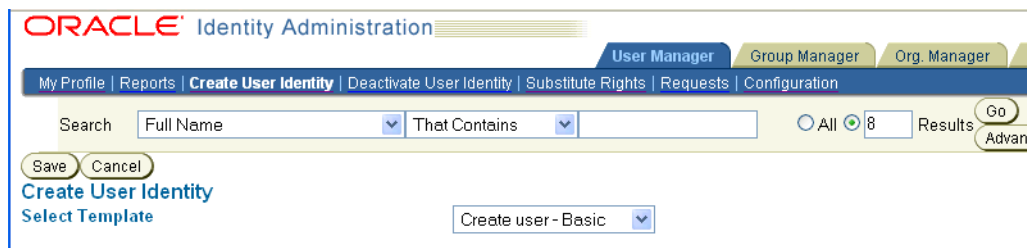
Using a Workflow

Once a workflow has been defined, users can invoke the workflow from the associated function in the User Manager, Group Manager, or Organization Manager. Participants in steps other than the Initiate step of the workflow can find and process tickets. Users can delete workflow requests, archive requests, and monitor the progress of a workflow.

Note that to be able to perform the actions specified in the workflow definition, participants in a workflow must be granted permission to view and modify the attributes affected by the workflow. See ["Allowing Users to View and Change LDAP Data"](#) on page 4-22.

Invoking a Workflow

Once a workflow is defined, it becomes a piece of embedded functionality in the User, Group, or Organization Manager. The workflow can be invoked by any user who has been defined as a participant in the Initiate step for this workflow. For example, suppose you define a Create User workflow. Users who are in the domain specified for the workflow can invoke this workflow from the Create User function in the User manager. If multiple workflows have been defined for a create operation, you will see a list on the create page for that object.



Users can also initiate a Change Attribute workflow. Change attribute workflows are available on profile pages that the user is permitted to access. For example, suppose a workflow has been defined for the manager attribute displayed on a profile page. When users change departments, they may need to issue a request to change the name of their manager. This request can be handled by a Change Attribute workflow.

To invoke a change attribute workflow

1. From the User Manager, click My Identity, then click Modify.

Your user profile page is displayed using editable fields for all attributes that you can change.

2. For attributes on your profile page that have the Request to Remove or Request to Modify buttons, you may request to remove or change that attribute value.

These buttons are displayed when a Change Attribute workflow or subflow with a Request to Remove or Request to Modify action has been created.

Your request is sent in the form of a ticket. The person who processes this ticket may approve or reject the request. See ["Finding and Processing a Ticket"](#) on page 5-48 for details.

Finding and Processing a Ticket

Once a workflow has been initiated, subsequent steps are generated by processing a ticket. You can find pending workflow tickets in the User Manager, Group Manager, and Organization Manager.

To find a workflow ticket

1. From the User Manager, Group Manager, or Organization Manager, click Requests.

2. From the Requests page, click either Incoming Requests, Outgoing Requests, or Monitor Requests.

Note that outgoing requests are tickets that have already been processed.

3. In the Search list, select the application for which you want to view requests.
4. Specify a number of days in the text field, or leave this field blank to view all requests.
5. Click Go.

A list of workflow tickets is displayed. The list will match your search criteria.

To process a ticket

1. From the User Manager, Group Manager, or Organization Manager, click Requests.
2. From the Requests page, click Incoming Requests.
3. In the Search list, select the application for which you want to view requests.
4. Specify a number of days in the text field, or leave this field blank to view all requests.
5. Click Go.
6. A list of workflow tickets is displayed. The list will match your search criteria.
7. Click a link for an incoming request.
8. On the details page for the request, click the Process button.

If you are a participant for this workflow, a page is displayed showing the attributes configured for this step of the workflow.

9. Supply any required attributes for this workflow.

For example, a Create User step may prompt you to supply an email address for the new user. Any information you need to supply on this page is determined by how this step of the workflow was configured.

10. Click the appropriate button for completing this step of the workflow.

For example, on a Create User request, the detail page for this ticket may contain Approve and Reject buttons.

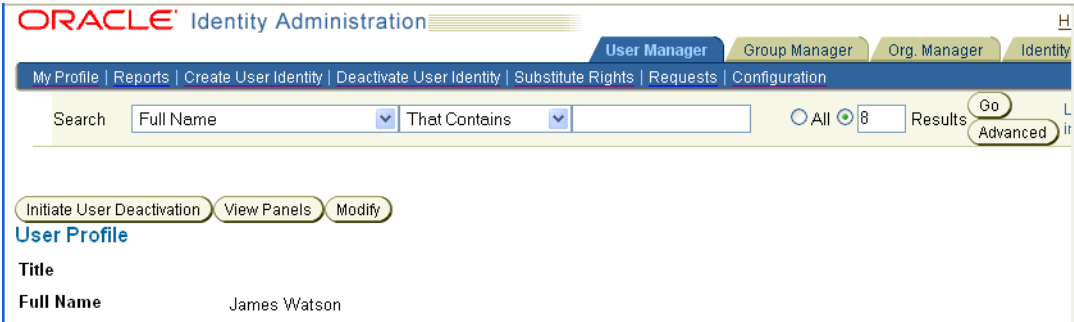
Deactivating and Reactivating Users

Once a user has been enabled in the Identity System, they can be deactivated and reactivated. Deactivation makes a user unable to log in and unavailable for viewing in the Identity System. It takes effect once the user logs out of the current session. An administrator with Monitor Requests privilege can view deactivated users and either permanently delete them or reactivate them.

Note: All configured directories will be searched to remove references to the Deactivated/Deleted user, including groups to which the user belongs. When you have stored user data and configuration data separately, both directories will be searched concurrently.

The steps for defining a workflow for deactivating a user are provided in ["Using the Workflow Applet"](#) on page 5-19. Once the workflow has been defined, users with

sufficient privileges will see an Initiate User Deactivation button on a user's profile page.



The steps for deactivating the user will conform to the actions you specified on the user deactivation workflow.

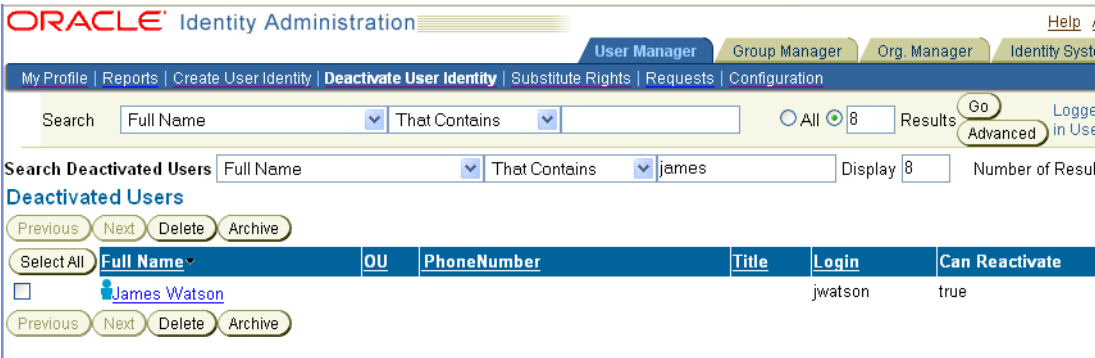
Reactivating a Deactivated User

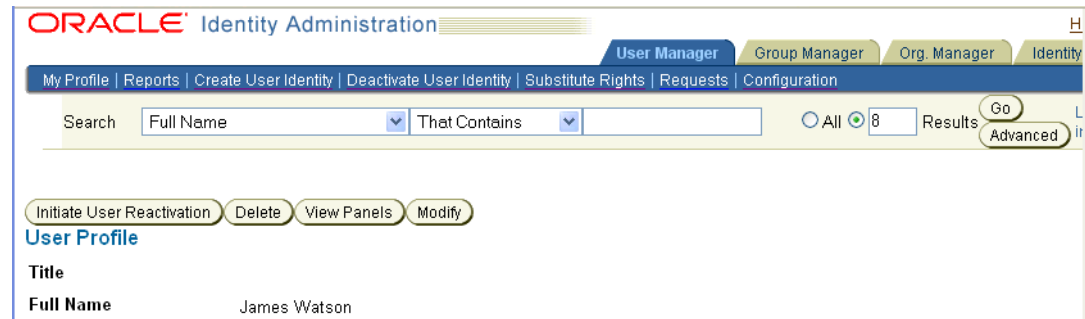
At times you may want to reactivate a deactivated user. For example, you may want to deactivate an employee during a leave of absence, and reactivate the employee when he or she returns to work.

To reactivate a deactivated user

- 1. Define a Reactivate User workflow for this purpose.
For a summary of actions permitted on a reactivation workflow, see "Workflow Types, Steps, and Actions" on page 5-7. Once the workflow has been defined, users with sufficient privileges will see an Initiate User Reactivation button on a deactivated user's profile page.
- 2. In the User Manager, click the Deactivated User Identity sub-tab to display the Search Deactivated Users page.
- 3. Search for, then select the deactivated user name for the identity you want to reactivate.

The View Profile page appears.





4. Click the Initiate User Reactivation button on the View Profile page.

The user is reactivated.

Note: When you reactivate a user, you must manually add the user to any groups he or she belonged to, and re-set attribute policies and the searchbase for the user.

Monitoring a Workflow

Users who have the right to monitor a workflow can view the progress of a workflow, including request tickets owned by others.

Only requests within your management domain are listed. See "[Delegating Administration](#)" on page 2-5 for details.

To monitor a workflow

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

Note: For subflows, if the first step has not been processed, the Date Processed field is empty.

3. In the Search fields, select your search criteria and click Go.
The results appear after the search fields.
4. Click Next or Previous as necessary to see other results.
5. Click a ticket's Request Number to open the Request page for that ticket.

This page lists the workflow's current step number.

To delete an incomplete workflow that is not responding, use the Monitor Requests function to locate the workflow and then click the Terminate button. To terminate a completed workflow, use the Delete button in the Monitor Workflow functionality.

Archiving Requests

Archiving the workflows enables you to keep a record of participants and times but remove a large amount of data from the directory. You may want to archive workflows

periodically to prevent the Oblix tree from becoming too large. You can archive only completed workflows. Multiple archive operations add information to this file.

Archived workflows are stored in LDIF format. The default location for storing the archive data is the following:

Identity_Server_install_dir/oblix/data/common/wfinstance.ldif

You may want to change the default location to prevent the archived data from being overwritten during an upgrade. The following files control the archive file location:

Filename	Application
usc_wf_params.xml	User Manager
gsc_wf_params.xml	Group Manager
osc_wf_params.xml	Org. Manager

Oracle recommends that you only archive a maximum of a few hundred workflow instances at a time. If you select a large number of instances, it can take several seconds and consume server resources during the archive operation.

To archive a workflow

1. View workflow requests, as described in ["Monitoring a Workflow"](#) on page 5-51.
2. Select requests in the Select All column.
3. Click Archive.
4. When the archive confirmation page appears, click Back to return to the previous page.

Note: You must restart the Identity Server after changing parameter files.

Deleting Requests

You can remove workflow requests.

To delete requests

1. View requests as described in ["Monitoring a Workflow"](#) on page 5-51.
2. Select requests in the Select All column and click Delete.
3. When the delete confirmation page appears, click Back to return to the previous page.

Preventing Other Administrators from Working on a Workflow Ticket

At run time, multiple users may receive a workflow ticket. For example, suppose an IT group receives a ticket for a Create Workflow request. An administrator who processes this request can lock the ticket so that other users can view the information on the locked ticket but they cannot work on the ticket. Only the person who locked the Ticket, the Master Identity Administrator, and people who have been granted permission to Monitor Requests can unlock the ticket.

To lock or unlock a ticket

1. Open a ticket as described in "To process a ticket" on page 5-49.

The Lock and Unlock buttons are displayed on the workflow page when you process the workflow ticket.

2. Select Lock or Unlock, as appropriate.

Managing Workflows

Once you have defined workflows, you can view, copy, modify, delete, and export them.

Viewing and Exporting a Workflow Summary

You can view a summary of a workflow, including its steps, participants, and so on, and export this report to a comma-delimited value (CSV) file.

Note: The following is of interest if you use Microsoft Internet Explorer and you are protecting the Identity System interface (WebPass) with a WebGate. To enable the Export to CSV feature, you must configure the following two WebGate parameters as follows:

CachePragmaHeader: Leave blank

CacheControlHeader: Specify Private or leave blank

See *Oracle Access Manager Access System Administration Guide* for details.

To view and export a workflow summary

1. Access the User, Group, or Organization Manager.
2. Click Configuration, then click Workflow Definition.
3. From the workflows menu, select the workflow you want to view.
4. Select View.

The Workflow Definition View page appears.

Workflow Definition View

Workflow Name : Create user - Basic

Workflow Type : Create User

Workflow DN : obworkflowid=wfgs20051206t18100081532,obcontainerid=workflowdef

Workflow Status : Enabled

No. of Steps : 3

Step ID	Step Name	Entry Condition	Attribute Name	Attribute Kind	Attribute Properties
1	Initiate		Full Name	Required	
			Last Name	Required	
			Login	Optional	
			User Password	Required	
2	Enable	1:True:False			
3	Error Report	1:False:False,2:False:False			

Close Export to .csv file

Java Applet Window

5. Enlarge the Workflow Definition View page or scroll to the right to see all of the workflow contents.
6. Click Export to CSV to save a comma-delimited value file of your workflow.
7. Click Close to close the page.

A sample CSV file, when opened in a spreadsheet, may appear as follows:

	A	B	C	D	E	F	G
1	Workflow Name	Create user - Basic					
2	Workflow Type	Create User					
3	Workflow DN	obworkflowid=wfgs20051206t18100081532,obcontainerid=workflowdef					
4	Workflow Status	Enabled					
5	No. of Steps	3					
6	Descriptor	Simple create user workflow - generated using workflow QuickStart					
7	Workflow DN	company:o=company,c=us					
8	Workflow DN	o=company,c=us:					
9							

Copying a Workflow

You can use a copy of a workflow as a starting point for a new workflow. You can also copy a workflow if you would like to modify a workflow that has too many pending tickets for the modify function to be practical.

To copy a workflow as a starting point for a new workflow

1. Access the User, Group, or Organization Manager.
2. Click Configuration and click Workflow Definition.

3. From the Workflows menu, select the workflow you want to copy.
4. Click Copy.
5. A copy of the workflow appears in the list.
It is named Copy of original name. Oracle recommends renaming the copied workflow, although you are not required to do so.
6. Change information as necessary to create a new workflow.

To copy a workflow as an alternative to modifying it

1. Copy the workflow, as described in the previous procedure.
2. If the workflow has external actions, update the oblixpppcatalog catalog file to reference the new workflow.

See *Oracle Access Manager Developer Guide* for details.

3. Modify the copy.
If you intend the workflow to only be invoked from the Identity System, you are done.
4. If the workflow is embedded as a portal insert an another application Web page, update the link for the workflow to point to the new workflow ID.

Modifying a Workflow

After creating a workflow, you can change its parameters. If the selected workflow has pending instances, you can only modify the list of Targets, the Participants for any step, or the Pre/Post Notification recipients for any step.

Note: Since only certain parts of a workflow can be modified when there are pending tickets, on a very active system you may need to copy the workflow and make changes to it. See ["Copying a Workflow"](#) on page 5-54 for details.

To modify a workflow

1. Access the User, Group, or Organization Manager.
2. Click Configuration and click Workflow Definition.
3. From the Workflows menu, select the workflow you want to modify.
4. Click Modify.
The selected workflow information appears. Clicking Modify disables this workflow so that it cannot be used while being modified.
5. Change the workflow settings as necessary.
6. Click Save Workflow to save your changes.
7. Click Yes when prompted to enable the workflow.

Deleting a Workflow

You can delete a workflow unless the workflow has pending requests.

To delete a workflow

1. Access the User, Group, or Organization Manager.
2. Click Configuration and click Workflow Definition.
3. From the Workflows menu, select the workflow you want to delete.
4. Click Delete.
5. Click OK at the confirmation message.

Exporting Workflows

You can export all workflows to a comma-delimited value (CSV) file. This is a text file that can be printed.

Note: The following is of interest if you use Microsoft Internet Explorer and you are protecting the Identity System interface (WebPass) with a WebGate. To enable the Export to CSV feature, you must configure the following two WebGate parameters as follows:

CachePragmaHeader: Leave blank

CacheControlHeader: Specify Private or leave blank

See *Oracle Access Manager Access System Administration Guide* for details.

To export workflows

1. Access the User, Group, or Organization Manager.
2. Click Configuration, then click Workflow Definition.
3. From the workflows menu, select the workflow to export.
4. Click Export All to be prompted to save a comma-delimited value file that includes all of your workflows.

Viewing Workflow Panel Settings

As described in "[About User, Group, and Organization Manager](#)" on page 4-2, you configure what appears in the User, Group, and Organization Manager applications. The User and Group Manager applications consist of one tab and the Organization Manager consists of one or more tabs. Tabs are a collection of profile pages, which themselves are collections of panels. Panels are groups of attributes and values.

You can view and modify the workflow panels that appear on the profile pages for these applications.

To view current workflow panel settings

1. From the Identity System Console, click Common Configuration.
The Common Configuration page appears.
2. Click Workflow Panels.

The Workflow Panels page displays configured workflow panels.

ORACLE Identity Administration

User Manager Group Manager Org. Manager

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | **Common Configuration**

- Object Classes
- Workflow Panels**
- Master Audit Policy
- Global Auditing Policies

Configure Workflow Panels

Panel Name	Description
Workflow monitor table	Used for workflow monitor search results
Workflow profile panel	Used for workflow instance page
Workflow steps profile panel	Used for workflow steps information
Ticket information panel	Used for the ticket information page
Ticket search table	Used for ticket search results

The following table describes each panel.

Panel	Description
Workflow monitor table	The columns included in the results page when a user performs a workflow search from the Monitor Requests page.
Workflow profile panel	The information displayed about a workflow instance in the Monitor Requests page.
Workflow steps profile panel	The information displayed about a workflow instance's steps in the Monitor Request page.
Ticket information panel	The information displayed in the Ticket Information page from the Incoming Requests or Outgoing Requests page.
Ticket search table	The information displayed in the results page when a user performs a workflow search from the Incoming Requests or Outgoing Requests page.

- Click the panel you want to view.

The View Panel page displays the items that are displayed on the panel.

Panel field	Description
Panel Label	Name of the panel as displayed in the Identity System. This field can be localized.
Description	Description of what this panel does. This field can be localized.
Attributes	Attributes used for the panel columns and their display names. This field can be localized.

Modifying the Appearance of Workflow Panels

You can modify, but cannot delete, workflow panels.

To modify a workflow panel

- From the Identity System Console, click Common Configuration.

The Common Configuration page appears.

- Click Configure Workflow Panels.

The Configure Workflow Panels page displays configured workflow panels.

3. Click the panel you want to view.
4. Click Modify.

The Modify Panel page appears.

ORACLE Identity Administration Provisioning Help About

User Manager Group Manager Org. Manager Identity System Console

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | **Common Configuration** Logged in user: Master C

- [Object Classes](#)
- **[Workflow Panels](#)**
- [Master Audit Policy](#)
- [Global Auditing Policies](#)

In the Modify Workflow Panel screen, you can modify attribute information that displays in the Workflow Ticket Search Results, Workflow Steps Information, and Ticket Information pages.

Panel Label

Description

Attributes

Step Number	Step Number
Action	Action
Action Taker	Action Taker
Status	Status
Subflow Number	Subflow Number

5. In the Panel Label field, type a new name for this panel as it will appear in the application.
6. In the Description field, type a description.
7. In the Attributes fields, select attributes to display on the application in the order in which they will appear.
8. Click Save.

Localizing Workflow Panels

You can localize workflow panels if you want to display the panel information in more than one language. To do this, you must do the following:

- Install the appropriate language packs.
- Manually enter the panel display information in the Administration Console for each language that you installed.

See ["Making Schema Data Available to the Identity System"](#) on page 3-1 and ["Configuring User, Group, and Organization Manager"](#) on page 4-1 for information on localizing attributes.

To view language-specific workflow panel information

1. From the Identity System Console, click Common Configuration.
The Common Configuration page appears.
2. Click Configure Workflow Panels.

The Configure Workflow Panels page displays configured workflow panels.

3. Click the panel you want to view.

The details of the workflow panel such as the panel name, description, and attributes are displayed on the page.

4. Click Translate.

Note: The Translate button appears only if more than one language pack is installed.

The Summary of Panel Display Names page appears. The language-specific display name for the panel fields and attributes are displayed. Fields that has not been translated for a language are marked as Not Configured.

To configure language-specific workflow panel information

1. In the Summary of Panel Display Names page, click Modify

The Configure Panel Display Names page appears. This page contains panel information and links for the languages that you have installed

2. Click the language for which you want to configure the workflow panel.

The Configure Panel Display Names page for the selected language appears.

3. Enter the following information:

- **Panel Label:** Enter the language-specific display name for the panel.
- **Description:** Enter a brief description of the panel. This is optional.
- **Attributes:** Enter the language-specific text for each attribute display name.

4. Click Save to save your changes; click Cancel to exit the page without saving your changes.

Workflow Performance

Access to the directory server access can be reduced by setting the WfInstanceNotRequired flag to true in the oblix/data/common/workflowdbparams.xml file. This flag indicates that no workflow instances should be written to the directory server unless necessary. It is set to false by default, which means workflow instances are written to the directory server.

For more information about workflow performance see *Oracle Access Manager Deployment Guide*.

The Identity Administrator's Modify Rights

As defined in "[Specifying Identity System Administrators](#)" on page 2-1, only an Identity Administrator can manage the User, Group, and Organization Manager.

By default, an Identity Administrator bypasses attribute access controls. As a result, if you define a Change Attribute workflow, the attribute access controls in this workflow are not checked for Identity Administrators. These administrators automatically have modify rights where other users have only the right to request to modify an attribute.

The parameter to control this feature is BypassAccessControlForDirAdmin, located in *IdentityServer_install_dir/identity/oblix/apps/common/bin/globalparams.xml*. If you

want to not automatically provide modify rights to the Directory Administrator, set this flag to false and restart the Identity Server.

You can give the Identity Administrator the right to modify an attribute and request to modify an attribute in a Change Attribute workflow. In each application parameter file, for instance, *IdentityServer_install_dir/identity/oblix/apps/userservcenter/bin/userservcenter.xml*, you can set the parameter `checkChangeAttributeEvenAllowModify` to true. This setting provides that even if the administrator is allowed to modify an attribute, this person will see both input and workflow buttons. This parameter applies to administrators who have both modify and initiate workflow rights. Note that this feature can introduce performance overhead.

Advanced Workflow Options

The following advanced options are available:

- Attaching custom code to workflow actions
- Configuring the behavior of workflow actions

Pre and Post Actions

You can use the Identity Event Plug-in API to attach custom code to workflow actions. Common scenarios for using the Identity Event Plug-in API with workflows include:

- Automatically generating a value (such as a unique ID) from an external system
- Validating data for a workflow step
- Updating data in an external system

Once you write custom code, you must tell the Identity System to execute it either before the workflow action (pre action), or after the workflow action (post action) in the *oblixpppcatalog* file.

See the *Oracle Access Manager Developer Guide* for more information.

External Actions

An external action serves the same purpose as pre and post actions, but differ from Identity Event Plug-in API actions in two ways:

- An external action is not attached to an existing action.
- Routing paths can be fully configured based on the external action's exit condition.

You implement the external action code as a hook in the *oblixpppcatalog* file. See the *Oracle Access Manager Developer Guide* for more information.

Customization of Data and Actions in a Workflow

The User, Group, and Organization Manager each have a workflow parameter file that controls the data displayed to participants and the actions that can be selected.

Each parameter file has three sections:

- Create Object
- Delete Object
- Change Attribute

The files are located in

`IdentityServer_install_dir/identity/oblix/apps/applicationname/bin/`

where *IdentityServer_install_dir* is the Identity Server installation directory and *applicationname* is one of the following:

- `usc_wf_params.xml`: User Manager
- `gsc_wf_params.xml`: Group Manager
- `osc_wf_params.xml`: Org Manager

The following table describes each parameter:

Parameter	Description	Sample Setting
occurrence	Indicates how many times this action may be used within a workflow.	[1] [n] 1: action can be used once. n: Action can be used multiple times.
useraction	Indicates whether or not the step is interactive.	[true] [false] true: Action requires user interaction. false: This is a background step and requires no user interaction.
forceCommit	Indicates whether an implicit commit takes place for this step, even though this action is not a commit. An implicit commit writes all collected data to the specific target entry	[true][false] true: Implicit commit takes place. false—Implicit commit does not take place.
pre_action	Indicates that the current action can be specified if the previous step's action is in this list.	[list of actions]
exit_condition	Indicates the possible results for the given action.	[list of exit conditions] For example: true: 1 false: 0
relevant_data	Indicates which types of relevant data can be configured for this step. Background steps do not contain any relevant data.	[list of relevant data] Can be any combination of Required, Provisioned or Optional.
initialStep	A parameter you can apply to an initiate, self registration, or approval step.	Values are true and false.

Adding Roles to a Workflow

By default, only the role of Anyone is available in a workflow definition applet. To add the roles that have been defined in the directory to the workflow definition applet, you can modify the workflow parameter files for the User Manager, Group Manager, and Organization manager.

The following procedures cause all DN roles to show up in the workflow applet.

To configure a role

1. Open the Modify Attributes applet as described in ["Configuring Attributes"](#) on page 3-17.
2. For a person object class or an auxiliary object class associated with the person object class, select an attribute with a DN data type.
For example, you might select the Manager attribute.
3. From the Display Type list, select the Object Selector display type for this attribute.
4. Select the person object class, for example, gensiteOrgPerson, in the Target Object Class list.
The attribute will display as a role in the workflow applet, provided all roles are enabled as described in the following procedure.
5. Click Save.

To add roles to a workflow definition applet

1. Edit the WF parameter file in the following location:
User Manager: Open usc_wf_params.xml.
Group Manager: Open gsc_wf_params.xml.
Organization Manager: Open osc_wf_params.xml.
2. Go to the section `<CompoundList ListName="Roles">`
3. Find the appropriate Workflow Type.
For example, to modify a create object workflow, you would need to find `<CompoundList ListName="CREATE_OBJECT">`
4. Find the Participant or Notiffee section in this file.
For example, you could edit the section `<ValNameList ListName="Participant" >`
5. Add the following line:
`<NameValPair ParamName="dns" Value="dns"/>`

Creating a Self-Registration Workflow

Self-registration enables users to add themselves or their organizations to the Identity System directly from a Web page. The Identity System does not provide a user interface for self-registration. You must configure a URL that displays a registration form.

When users self-register, they may be prompted to reset their passwords after their initial login attempt. This depends on settings provided for the Change On Reset field, as described in ["Configuring Password Policies"](#) on page 7-49. If more than one user self-registers using the same browser session and the Change On Reset option is chosen, all users after the first are prompted to change their passwords after their first login.

If you want users to be automatically logged in to the Identity System after self-registering, you must set the SelfRegGeneratesSSOCookie to true in the basedbparams.xml file. See the *Oracle Access Manager Customization Guide* for details.

Additional information on customizing self-registration pages is provided in the *Oracle Access Manager Customization Guide*.

The following procedure illustrates a self-registration workflow for Basic authentication.

To create a self-registration workflow

1. From the Identity System Console, select the User, Group, or Organization Manager.

If the Organization Manager has more than one tab, select the appropriate tab.

2. Click Configuration, then click Workflow Definition.
3. Define a Create User or a Create Organization workflow using self-registration as the first step.
4. Access this workflow and record the workflow's Distinguished Name and the target domain.

You will add this information to the self-registration URL.

5. Add the self-registration URL to an HTML document as follows:

```
https://domain_name:port/identity/oblix/apps/userservcenter/bin/
userservcenter.cgi?program=workflowSelfRegistration&ObWorkflowName=workflow_DN
&ObDomainName=target_domain
```

Variables are as follows:

- *Domain_name:port*: host system's domain name and port number.
- *Workflow_DN*: the DN for the workflow.
- *Target_domain*: the target path, without a name.

The value of ObDomainName *target_domain* is a target domain that was defined in the self-registration workflow. See ["Defining an LDAP Target for Create Object Workflows"](#) on page 5-22 for details.

For organization self-registration, use this format:

```
https://domain_name:portnumber/identity/oblix/apps/objservcenter/
bin/objservcenter.cgi?program=workflowSelfRegistration&tab_id=tab_
name&ObWorkflowName=workflow_DN&ObDomainName=target_domain
```

Variables are as follows:

- *Domain_name:portnumber*: host system
- *Tab_name*: the name of the tab
- *Workflow DN*: the workflow's DN
- *Target_domain*: the target path, without name

The URL for self-registration must be to a page that does not require authentication. The self-registration URL is not the usual `/identity/oblix/apps/userservcenter/bin/userservcenter.cgi`. Typically, when a user accesses the Identity System, the Access System asks the user to authenticate. However, the WebGate should be set up to not request authentication for people accessing self-registration and lost password pages.

6. Replace reserved characters with URL-compatible text substitutes.

When providing a DN path in the dynamic expansion URL, you must encode URL-reserved characters (non-alphanumeric) with a % followed by the character's ASCII hexadecimal equivalent, as follows:

- %3D: Equal sign (=)
- %2C: Comma
- %20: Space

For example:

```
cn=Engineering Team, ou=Engineering, o=Company, c=US
```

is replaced by:

```
cn%3DEngineering%20Team%2C%20ou%3DEngineering%2C%20o%3DCompany%2C%20c%3DUS
```

7. Save the HTML file.

The following is an example of a self-registration URL:

```
http://silicon/identity/oblix/apps/userservcenter/bin/userservcenter.cgi?program=workflowSelfRegistration&obdomainname=o%3DCompany%2Cc%3DUS&obworkflowname=obworkflowid%3D20020605T1132216476%2CobcontainerId%3Dworkflowdefinitions%2co%3Doblix%2Co%3Dconfigdata
```

Note: If you are using Sun's iPlanet directory, note that self-registration passwords cannot use UTF-8 characters. If the user supplies UTF-8 characters, the iPlanet directory default 7-bit plug-in fails the operation. By default the 7-bit plug-in requires the uid, mail, and userpassword attribute values to be 7-bit. To resolve this problem, turn off the plug-in or remove the userpassword attribute from the configuration. Note also that this issue applies as well to Create User and Modify Profile operations.

Creating a Location Workflow

In the Organization Manager, you can create workflows to manage business locations and allow specific users to manage those locations. You can select individual users or users who play a specific role such as Facilities Manager, or you can select specific groups such as IT Operations.

To enable users to view the location of the organization, you can attach .gif images of the location map to the workflow. When users click a location, the location profile displays a map of the area where building is located.

You can create a new location workflow and then create a location object using the new workflow. To do this, use the Create Org Profile feature in the Organization Manager. Alternatively, you can create a location object first and then link it to an existing workflow. Once you create a location object, you can assign other objects such as users to specific locations on the map.

Note: If Location ID has the Semantic type DN Prefix it is important to note Active Directory and ADAM do not allow multi-valued RDNs (although iPlanet/SunOne do). For Active Directory and ADAM, ensure that the Attribute Value(s) selection is Single in the meta-attribute configuration.

After you have created a location object, you must enable the Location functionality and enable users with the appropriate permissions to view the user or object's location.

Task overview: Enabling Location functionality and users

1. Modify the Location tab for the Organization Manager, then add location attributes to the Profile pages for User Manager and the Organization Manager.
See ["Enabling the Location Tab in Organization Manager"](#) on page 4-38 and ["Configuring Tab Profile Pages and Panels"](#) on page 4-12 for details.
2. Configure read permissions for location attributes.
See ["Allowing Users to View and Change LDAP Data"](#) on page 4-22 for details.
3. Define a Create Location workflow as described in ["Task overview: Defining a Create Location workflow"](#) on page 5-65.
4. Create a new location and establish the location's hierarchy in relation to other locations if applicable.
See ["Adding Object Classes"](#) on page 3-8 for details.
5. Assign a value for the Location attribute for a user or object profile.
See ["Using the Workflow Applet"](#) on page 5-19 for details.

Note: Attribute values can also be added and modified on object profile pages as well as through a workflow.

Task overview: Defining a Create Location workflow

1. Initiate a workflow as described in ["Starting a New Workflow Definition"](#) on page 5-20.
2. Create one or more subflows, if necessary, as described in ["About Subflows"](#) on page 5-15.

Note: You must create subflows before you initiate the main workflow. This enables you to link a subflow to the main workflow.

3. Select the attributes that you want to associate with the workflow as described in ["Defining Step Attributes"](#) on page 5-26.

The available default location attributes are Location ID, Location Name, Location Title, and Map Image. Location ID and Location Name are required attributes.

4. Specify participants as described in ["Defining the First Step in a Workflow"](#) on page 5-24.
5. Define the workflow process as described in ["About Step Actions"](#) on page 5-10.
6. Save the workflow.
7. Enable the workflow as described in ["Enabling a Workflow"](#) on page 5-30.
8. Test the workflow to ensure its validity as described in ["Testing a Workflow"](#) on page 5-31.

Sending Non-LDAP Data to External Applications

The User, Group, and Organization Manager are Identity System applications that enable users to view and modify information about themselves, other people, groups, inventory, and any other item that you, the administrator, choose to make available. As explained in the chapter, you can apply business logic to actions performed in the Identity System applications, so that, for example, review and approval must be performed before information about a user can be modified.

With the Identity System's object template functionality, you can extend an Identity workflow so that information that is added, deleted, or changed is propagated to other applications. With templates non-LDAP schema can be managed in the Identity System, allowing for non-LDAP data to be made available to workflows and to the Identity Event API.

This chapter covers the following topics:

- [About Configuring Non-LDAP Data](#)
- [Summary of Using Non-LDAP Data in a Workflow](#)
- [About Template Objects](#)
- [About Template Object Data and Workflows](#)
- [Object Template Configuration](#)
- [Sample Object Template File](#)
- [Creating an Identity Event Plug-In for Template Attributes](#)

About Configuring Non-LDAP Data

The User Manager, Group Manager, and Organization Manager applications rely on information in an LDAP directory or in an object template:

- **LDAP Directory**--You configure the information in the directory to display data on profile pages and to enable configuration of workflows that manipulate data about users, groups, and objects. The LDAP directory is the authoritative data source for the Identity System.
- **Object Template**--You can manually configure information in an object template to propagate data that is entered during an Identity workflow step to different target data sources. For example, you can configure an Add or Modify User workflow that uses non-LDAP data and makes this available to the Identity Event API (see *Oracle Access Manager Developer Guide* for details).

Summary of Using Non-LDAP Data in a Workflow

For sending non-LDAP data to a back-end system, the process is as follows:

Task overview: Configuring non-LDAP data for a back-end application

1. Configure an object template, as described in ["Object Template Configuration"](#) on page 6-4

The template should contain objects and attributes that can be understood by the back-end application.

2. Store this file in:

IdentityServer_install_dir\oblix\config\template\xxx.tpl

where *IdentityServer_install_dir* is the directory where the Identity System is installed and *xxx* is the name of the .tpl file.

3. Configure the template objects and attributes in the Identity System Console, as described in ["Making Schema Data Available to the Identity System"](#) on page 3-1 and note the following:

- Users can only specify values for template attributes in the context of a workflow step.
- Template attributes are not searchable in the Identity System.
- You cannot set View or Modify permissions for template attributes.

You configure access control for template attributes by defining workflow step participants. Only individuals who are step participants have access to these attributes.

- Template attributes cannot be configured as derived attributes.

Note: After you configure template objects and attributes in the Identity System Console, do *not* modify the template file. This restriction is the same as for an LDAP schema, which you also should not change after configuration. Such changes can cause unpredictable behavior and are not supported.

4. Associate one or more template attributes with panels on a tab in an Identity System application, for example, the User Manager, as described in ["Configuring User, Group, and Organization Manager"](#) on page 4-1.
5. Create a workflow and associate the template attribute with one or more workflow steps, as described in this chapter and ["Chaining Identity Functions Into Workflows"](#) on page 5-1.

The attribute display name appears on the profile page associated with the workflow. However, the attribute value is not shown. This is because the data flow is only one-way from the Identity System to the target application. (In a future release, the data flow will be two-way, which will permit the display of these attribute values.) See ["About Template Object Data and Workflows"](#) on page 6-3 for details.

6. Ensure that the workflow has separate enable, commit, and other steps to write the data to each schema that is used in the workflow, as described in ["Configuring User, Group, and Organization Manager"](#) on page 4-1.

7. Configure an external action using the Identity Event API and IdentityXML to send the data in the object template to the back-end application, as described in the *Oracle Access Manager Developer Guide*.

Note: For non-LDAP attribute values, the IdentityXML actions of Add and Replace do not apply. The action Replace All is the only action that is used. If you create an IdentityXML statement using Add or Replace, the statement is processed as if you used Replace All.

The rest of this chapter discusses how object templates work and how to configure an object template.

About Template Objects

The Identity System provides a generic object template schema file, located in:

`Identity_install_dir\oblix\config\template\`

This is the required location for this file and for any other object template file that you configure.

Template objects created in this file are similar to LDAP objects. The primary difference is that you use LDAP objects and attributes to display data on user profile pages and to configure workflows, whereas template objects are only used in workflows.

You configure a workflow that contains one or more steps that perform actions on the template attributes. When a user invokes the workflow, the Identity System formats the data entered during the relevant step according to the requirements of the object template schema.

The workflow temporarily stores the template attribute values in the step instance. Once the commit step is performed, the data is written to the target back-end system. The data flow is one-way, so that the Identity System does not continue to store this data once it has been written to the back-end system.

Finally, you must create an Identity Event plug-in to send object template data to the target back-end system.

About Template Object Data and Workflows

As described in the previous discussion, you can configure a workflow step that uses template attributes to send non-LDAP data to back-end applications. Attribute values that the user enters as part of a workflow step, once committed, cannot be displayed on a profile page. This is because the Identity System sends data to, but does not retrieve the data from, the target application.

When you configure a workflow that uses an object template, you may want to configure commit steps that write both object template data and LDAP data. This would permit you to use your directory to display the LDAP attribute value on a profile page, and to use the template attribute value to send the data to the back-end application.

Since the flow of data from the Identity System to the back-end application is one-way, the user will not be able to verify the data in the Identity System. The user will need to view the target application or its logs to view the data created in the application from the workflow.

If there is an error writing the data to the back-end system, an Identity Event API plug-in can send a message back to the Identity System user.

Note: You cannot commit data for all data sources in one workflow step. You must configure one commit step for each domain.

Object Template Configuration

An object template file contains schema-like definitions for objects and attributes in XML format. The objects and attributes that you configure in an object template file correspond to values that can be understood by a back-end application to which you want to write data.

All object template files must reside in:

Identity_install_dir/oblix/config/template

where *Identity_install_dir* is the directory where the Identity System is installed. The file extension for any object template is .tpl.

The Identity Server reads the template file upon startup. If your installation uses multiple servers, you must copy the same template files to each server.

You can define multiple template object classes in a single file or in multiple files. If you create multiple files within the same domain, be aware that the Identity System enforces uniqueness of attributes and classes across files. If an attribute or a class already exists within a domain, the template file is not registered when the Identity Server starts up, and the objects cannot be shown in the System Console.

Similarly, the Identity Server cannot register the template file if it contains any syntax error. Instead, a log entry is generated.

Format of the Object Template File

The template file begins with a schema domain statement. The schema domain resolves ambiguity between object classes that have the same names but are used by different data sources, for example, the user object in LDAP and the user object in a back-end application.

The following is an example of a schema domain statement:

```
ObTemplateDefinition domain="exchange" version="1.0"/>
```

At startup time, the Identity Server reads the domain statement. For display purposes, template objects and attributes are shown in the Identity System Console in the following format:

attribute.class.domain

where the *domain* name is taken from the domain statement in the .tpl file.

Note that all domain statements must be unique. If the Identity System detects a non-unique domain, it fails to read the entire .tpl file. Note that the following domain names are reserved and cannot be used in your .tpl file:

- MIIS
- LDAP

The object template file enables you to define arbitrary name/value pairs. These name/value pairs must match those understood by the target application.

The template definition file is in XML format. The following is an example:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ObTemplateDefinition domain="ABC_APPLICATION" version="1.0">

  <!-- ObAttributeDefinition -->
  <ObAttributeDefinition name="cn" syntax="OB_CIS" maxlen="20"/>
</ObAttributeDefinition>
  <ObAttributeDefinition name="sn" syntax="OB_CIS" maxlen="20"/>
  <ObAttributeDefinition name="mail" syntax="OB_CIS" maxlen="20"/>
  <ObAttributeDefinition name="phone" syntax="OB_CIS" maxlen="20"/>

  <!-- ObClassDefinition -->
    <ObClassDefinition name="User">
      <ObAttributeReference name="cn" required="true">
      </ObAttributeReference>
      <ObAttributeReference name="sn" required="false">
      </ObAttributeReference>
      <ObAttributeReference name="mail" required="false">
      </ObAttributeReference>
      <ObAttributeReference name="phone" required="false">
      </ObAttributeReference>
    </ObClassDefinition>

    <ObClassDefinition name="Group">
      <ObAttributeReference name="cn" required="true"/>
      <ObAttributeReference name="sn" required="false"/>
      <ObAttributeReference name="uniqueMember" required="false"/>
    </ObClassDefinition>
  </ObTemplateDefinition>
```

How Template Objects Appear in the Identity System

Objects and attributes defined in the object template file appear as follows in the Identity System:

- Each object that you define in the .tpl file becomes selectable from the page displayed when you select the following from the Identity System Console:

Common Configuration, Configure Object Class, Add.

The name of the object class as displayed in the Identity System Console is in the format *class.domain*. The class is taken from the definition that you provide in the .tpl file.

- Each attribute that you associate with an object in the .tpl file becomes selectable from the page displayed when you select **Common Configuration, Configure Object Class, object class link, Modify Attributes**.

The name of the attribute as displayed in the Identity System Console is in the format *attribute.class.domain*. The *attribute* name is taken from the definition that you provide in the .tpl file.

- In each attribute statement, the syntax element determines the attribute data type that is displayed in **Common Configuration, Configure Object Class, object class link, Modify Attributes**.

You cannot choose the data type for a template object attribute from the Identity System Console. You must configure the data type in the syntax element of the attribute definition.

- Whether this attribute is single- or multi-valued, as seen in **Common Configuration, Configure Object Class, object class link, Modify Attributes**, is determined by the attribute definition in the .tpl file.
- Other characteristics of the attribute, such as the display name and semantic type, are configured from the Identity System Console.

The Identity System enforces the use of only one semantic type for each domain. For example, you can only assign the Login and Password semantic types once each in a domain.

- Unlike LDAP attributes, the attributes you configure in the .tpl file are not searchable and cannot be configured as derived attributes.

Elements in an Object Template File

The elements of the object template file are as follows.

The object template file begins with a list of attribute definitions. These attributes are referenced in the object definitions later in the file:

Table 6–1 Elements of ObAttributeDefinition

Element Name	Description
name	Name of the attribute. This corresponds to the attribute name displayed in the Identity System Console. This parameter is required. Length: 32 (max.) Format: [(a-z) (A-Z)][(a-z) (A-Z) (0-9)]
syntax	This is the attribute syntax. It corresponds to the attribute Data Type in the Identity System Console. This parameter is required. Format: ■ OB_DN--LDAP DN. This is synonymous with an LDAP DN attribute. This parameter enables you to configure an Object Selector display type in the Identity System Console. This parameter corresponds to an attribute data type of distinguished name in the Identity System Console. ■ OB_BIN--A binary. This corresponds to an attribute data type of binary in the Identity System Console. ■ OB_CES--Case Exact String. This corresponds to an attribute data type of string (case-sensitive) in the Identity System Console. ■ OB_CIS--Case Insensitive String. This corresponds to an attribute data type of string (case-insensitive) in the Identity System Console. ■ OB_INT--Integer. This corresponds to an attribute data type of integer in the Identity System Console. ■ OB_TEL--This corresponds to an attribute data type of telephone in the Identity System Console. ■ OB_POSTAL_ADDRESS--This corresponds to an attribute data type of postal address in the Identity System Console.

Table 6–1 (Cont.) Elements of ObAttributeDefinition

Element Name	Description
cardinality	The cardinality may be single or multi-valued. This corresponds to an attribute value of single or multi in the Identity System Console. If you set this value to multi, you can re-set it to single in the Identity System Console. However, if you set it to single in the .tpl file, you cannot reset it in the System Console. This parameter is optional. Default: <i>multi</i> unless specified otherwise. Format: [single multi]
maxlen	The maximum data length for the attribute value. This parameter is optional. Default: -1 unless specified otherwise. This setting indicates that no maximum length is enforced. Format: -1 or 1 - n where <i>n</i> is an integer that represents a reasonable maximum length.

Examples:

```
<ObAttributeDefinition name="c" syntax="OB_CIS" cardinality="single" />
<ObAttributeDefinition name="cn" syntax="OB_CIS" cardinality="single" />
```

In the .tpl file, a list of object classes occurs after the list of attribute definitions. Each object class contains an ObClassDefinition statement, followed by a list of ObAttributeReference statements.

Table 6–2 Elements of ObClassDefinition

Element Name	Description
name	Name of the class. This must be a unique name within the domain. This parameter is required. Length: 32 (max.) Format: [(a-z) (A-Z)][(a-z) (A-Z) (0-9)]

Examples:

```
<ObClassDefinition name="User">
<ObClassDefinition name="Group">
```

You associate an attribute with an object by including the attribute in an AttributeReference statement in the object class definition in the .tpl file. Each ObAttributeReference statement must refer to an attribute defined in an ObAttributeDefinition statement:

Table 6–3 Associations Between Attributes and Objects

Element name	Description
name	Name of the template attribute. This parameter is required. The attribute reference must be unique within the object class. The 'name' must be the name of an existing attribute definition (ObAttributeDefinition) in this domain. Length: 32 (max.) Format: [(a-z) (A-Z)][(a-z) (A-Z) (0-9)]
required	Specifies whether the attribute is required or optional in the context of the class definition. This parameter is optional. Default: 'false'. Format: ['true' 'false']

Examples:

```
<ObAttributeReference name="cn" required="true">
<ObAttributeReference name="mail" required="false">
```

Sample Object Template File

The following is an example of an object template file:

```
<?xml version="1.0" encoding="iso-8859-1"?>

<ObTemplateDefinition domain="myapplication" xmlns:dsml="http://www.dsml.org/DSML"
xmlns:oblix="http://www.oblix.com/">
  <ObAttributeDefinition name="c" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="cn" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="department" syntax="OB_CIS" cardinality="single"/>
  <ObAttributeDefinition name="l" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="location" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="mail" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="ou" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="uid" syntax="OB_CIS" cardinality="single" />
  <ObClassDefinition name="person">
    <ObAttributeReference name="c" required="false" />
    <ObAttributeReference name="cn" required="false" />
    <ObAttributeReference name="department" required="false" />
    <ObAttributeReference name="l" required="false" />
    <ObAttributeReference name="location" required="false" />
    <ObAttributeReference name="mail" required="false" />
    <ObAttributeReference name="ou" required="false" />
    <ObAttributeReference name="uid" required="false" />
  <ObClassDefinition name="organizationalUnit">
    <ObAttributeReference name="l" required="false" />
    <ObAttributeReference name="ou" required="false" />
  </ObClassDefinition>
  <ObClassDefinition name="locality">
    <ObAttributeReference name="l" required="false" />
  </ObClassDefinition>
  <ObClassDefinition name="country">
    <ObAttributeReference name="c" required="false" />
  </ObClassDefinition>
```

```
<ObClassDefinition name="computer">
  <ObAttributeReference name="cn" required="false" />
  <ObAttributeReference name="l" required="false" />
  <ObAttributeReference name="location" required="false" />
  <ObAttributeReference name="ou" required="false" />
</ObClassDefinition>
<ObClassDefinition name="group">
  <ObAttributeReference name="cn" required="false" />
  <ObAttributeReference name="mail" required="false" />
  <ObAttributeReference name="ou" required="false" />
  <ObAttributeReference name="uid" required="false" />
</ObClassDefinition>
<ObClassDefinition name="role">
  <ObAttributeReference name="l" required="false" />
  <ObAttributeReference name="ou" required="false" />
</ObClassDefinition>
</ObTemplateDefinition>
```

Creating an Identity Event Plug-In for Template Attributes

The *Oracle Access Manager Developer Guide* describes how to create a plug-in to send data from an Identity System workflow to a back-end application. Keep in mind the following when creating this plug-in:

- It is no longer possible to do bulk reactivations using the Identity Event API.
- When using IdentityXML and the Identity Event API, note that the only permitted action when sending attributes from the Identity System is Replace All. If you create an IdentityXML statement using Add or Replace, the statement is processed as if you used Replace All.

Configuring Global Settings

This chapter covers tasks that affect the basic functionality of Oracle Access Manager, including configuring multiple languages and directory and database server configuration.

This chapter also discusses how to control the appearance and functionality of Identity System applications. For example, through the searchbase and stylesheet, you may want to control what users can view or the actions they can perform in Identity System applications. You may want to add additional Identity Servers or WebPasses for better performance.

To help you manage these tasks, you can specify other Oracle Access Manager Administrators and Master Identity Administrators, as described in ["Specifying Identity System Administrators"](#) on page 2-1.

This chapter contains the following topics:

- [Configuring Styles for Identity System Applications](#)
- [Configuring Multiple Languages for Oracle Access Manager](#)
- [Configuring Identity Server Settings](#)
- [Managing Identity Servers](#)
- [Managing Directory Server Profiles](#)
- [Managing RDBMS Profiles](#)
- [Configuring WebPass](#)
- [Configuring Password Policies](#)
- [Configuring the Access Manager SDK for the Identity System](#)
- [Cloned and Synchronized Components](#)

Note: You must be a Master Administrator to configure the Identity System. Most tasks in this chapter are performed through the Identity System Console.

See also ["Changing Transport Security Modes"](#) on page 8-1, ["Implementing .NET Features"](#) on page D-1, ["Logging"](#) on page 10-1, ["Auditing"](#) on page 11-1, and ["SNMP Monitoring"](#) on page 12-1.

Configuring Styles for Identity System Applications

You use styles to change the appearance or limit functionality across Identity System applications. A style is a named collection of stylesheets, graphics files, and scripts that define a certain user interface for the system. A style is based on stylesheets that define the appearance of elements in application pages, including the names of fields and functions, the GIF images used to specify the colors, shapes, and sizes of tabs and buttons, and the fonts used for tab and button names.

Styles can be cosmetic or functional. A cosmetic style determines the appearance of Identity System applications, such as color, or the appearance of tabs. A functional style determines the functionality of Identity System applications. That is, you can add, modify, or remove particular functions on an Identity System application page. For example, you can remove the Substitute Rights function from all three Identity System applications. The Identity System provides a default style named Classic Style but you can use *PresentationXML* to develop other styles to change the appearance of the Identity System.

You can use the Customize Styles option in the Identity System Console to set the default style, create styles, modify styles, or delete styles. However, when you create or modify a style through the Identity System Console, the system copies the existing stylesheet and renames it. You must then open the stylesheet and manually make the necessary changes.

See the information on designing the GUI with PresentationXML in the *Oracle Access Manager Customization Guide* for details about creating and modifying stylesheets.

Note: You can change only the appearance of Identity System application pages. The System Console uses a style setting that cannot be modified.

This section discusses the following topics:

- [Viewing a Style](#)
- [Adding a Custom Style Directory](#)
- [Deploying a Style](#)
- [Changing a Style Name](#)
- [Modifying a Style](#)
- [Deleting a Style](#)
- [Setting the Default Style](#)

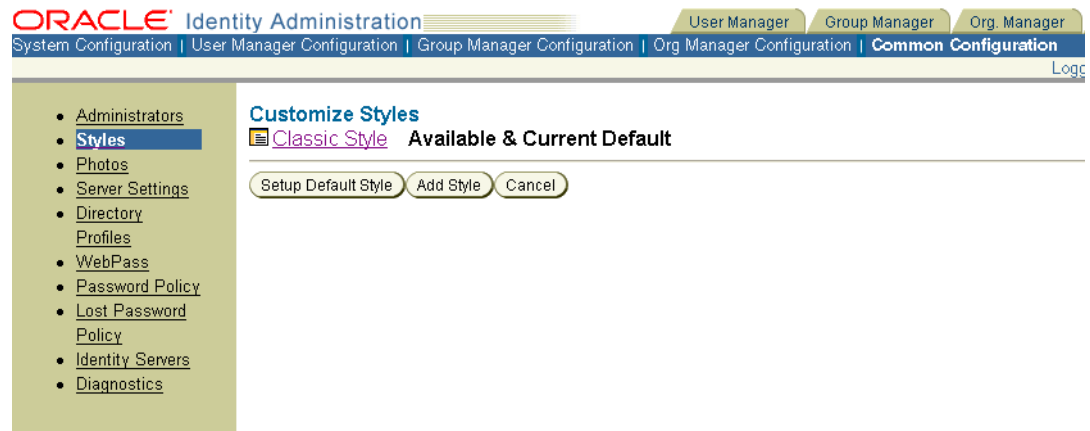
Viewing a Style

You can use the following procedure to view currently configured styles. This leads to the Customize Styles page, which is the starting point for style-related procedures.

To view currently configured styles:

1. From the Identity System Console, select System Configuration.
2. In the left navigation pane of the System Configuration page, select Styles.

The Customize Styles page appears. The following example shows the Classic Style, which is the default provided by the Identity System.



3. Click the style's link to view a style's parameters.

You will see the style name, directory where style files are stored, and the source of those files (in the Copy from field).

Adding a Custom Style Directory

The Identity System out-of-the-box provides one default presentation style, known as Classic Style. The *IdentityServer_install_dir/identity/oblix/lang/en-us/style0* directory contains XSL wrapper stylesheet files for the Classic Style. Most of these files point to global shared stylesheet template files for all languages in the *IdentityServer_install_dir/identity/oblix/lang/shared* directory.

The process of creating a custom style for presentation of Identity System pages for user applications begins by adding a new style to the Identity System, as described here. The result is a new custom style directory with XSL wrapper stylesheet files. Then you may either copy and modify an existing style or create an entirely new style based on new stylesheets.

Note: You may only change styles for user applications. The System Console always uses the default style.

In either case, adding a style (and custom style directory) to the Identity System follows the same method: you provide a style name and a directory name for your style files. You may also choose an existing style on which to base your new style. After selecting your new style as the default, you can begin customizing copies of Identity System stylesheets or creating your own. To complete the process, you need to copy your new stylesheets and GIFs to all Identity Servers and WebPass machines, respectively.

Before adding your first new style, there are a few things to take into account:

Multiple Languages: To support multiple languages, the Identity System provides a specific named directory for each installed language. For example, */lang/en-us* is the default English language directory, */lang/fr-fr* is the French language, and so on. Both the Identity System default and your custom style directories are stored within each installed language directory.

Suppose you have a French Language Pack installed. In this case, both *lang/en-us* and *lang/fr-fr* directories include the */style0* directory. When you add a style to the

Identity System, your new style directory is added in both the lang/en-us and lang/fr-fr directories:

IdentityServer_install_dir/identity/oblix/lang/en-us/NewStyle

IdentityServer_install_dir/identity/oblix/lang/en-us/style0

IdentityServer_install_dir/identity/oblix/lang/fr-fr/NewStyle

IdentityServer_install_dir/identity/oblix/lang/fr-fr/style0

Your Style Name: The Identity System uses the style name you supply internally. As a best practice, your style name should match your custom style directory name and should be easily recognizable. Do not include white spaces, &, *, or parentheses () in the name.

Your Style Directory Name: The directory name you specify will be used to create a directory for related wrapper stylesheet files. This name should match your style name and follows the same rules for naming.

In addition, your custom style directory name is also assigned to an XML document (a duplicate of style0.xml) that is created to identify the status and origin of your new style. For example, if your new directory is named Pastel, a file is created and stored as:

IdentityServer_install_dir/identity/oblix/config/style/Pastel.xml

No other files are created during this process. However, the styles.xml file will be updated to include a NameValPair specifying the directory and style name and directory name that you supply. For example:

IdentityServer_install_dir/identity/oblix/config/style/styles.xml

The style information files in the config/style are not included on WebPass. For more information, see the *Oracle Access Manager Customization Guide*.

Copy from an Existing Style: You may copy stylesheets from an existing style directory or select None to build a style that is not based on an existing style or to customize only selected stylesheets.

Note: If this is the first style being added, the only available style is the default Classic Style.

Selecting None: If you select None, the directory that is created is empty and you must manually create a set of stylesheets for your new style or selectively copy files from the /style0 directory to work with.

If you select None, your new style's status will appear as "Under Construction" within the Identity System until you select a style for it. An empty style directory is created automatically, and a duplicate of style0.xml is created in *IdentityServer_install_dir*/identity/oblix/config/style/style0.xml.

Selecting a Style: When you select a style to copy from, a duplicate of the directory you copied from is created under the custom directory name you specify. The copied files retain relative references to the directory you copied from (/style0 or a custom style that you chose to copy from).

During customization, you only change references that refer to the changed version of the stylesheet in the your new style directory.

Results: Suppose you added a new style named Pastel in a directory named Pastel and you copied from the default Classic Style. In this case, the Pastel directory is created in

each langTag directory and populated with duplicate files from Classic Style's directory, /style0:

```
IdentityServer_install_dir/identity/oblix/lang/en-us/Pastel
```

The Classic Style directory, /style0, remains intact as:

```
IdentityServer_install_dir/identity/oblix/lang/en-us/style0
```

In addition, an XML document that duplicates style0.xml is created when the new style is selected as the default, named after the directory you created, and stored with style0.xml in config/style:

```
IdentityServer_install_dir/identity/oblix/config/style/Pastel.xml
```

```
IdentityServer_install_dir/identity/oblix/config/style/Pastel.xml.lck
```

For additional information and a look at the content of various files, see the *Oracle Access Manager Customization Guide*.

To add a style

1. From the Identity System Console, select System Configuration, then select Styles to display the Customize Styles page.
2. Click the Add Style button to display the Add Style page.
3. Fill in the fields on the Add Style page, for example:

Name: Pastel

Directory Name: Pastel

Note: The style name you specify here is used internally by the Identity System and should match the directory name you supply.

4. In the Copy From field, select an existing style to use as a template for your new style.

For example:

Classic Style

5. Click Save to save the new style.

The new style name is listed in the Customize Styles page. One or more directories are created to hold the new wrapper stylesheets.

6. Select the new style as your default style, as follows:
 - a. Click the Setup Default Style button to display the Set Default Style page.
 - b. Click the Make Default button beside your new style name, then click Save.

7. Check your file system for the new style directory name you specified.

Next you customize styles. For details about customizing Identity System pages, copying styles, testing, and propagating styles, see the *Oracle Access Manager Customization Guide*.

Deploying a Style

The following procedure enables you to make a style available to end users:

To deploy a style

Append the directory name containing the stylesheets to the URL of the first page where users enter the Identity System, as follows:

```
&style=directory_name
```

where *directory_name* is the name of the directory that contains the stylesheets for the Identity System.

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

Changing a Style Name

You can change the name of a style using the steps in the following procedure as a guide.

To change a style name

1. From the Identity System Console, select System Configuration, then select Styles to display the Customize Styles page.
2. In the Customize Styles page, click the name of the style that you want to rename.
The View Style page appears.
3. Click Modify.
The Modify Style page appears.
4. Change the style name.
5. Click Save to save your changes.
The View Style page displays the new style name.

Modifying a Style

You cannot modify Classic style, the default style provided by the Identity System, because it is used by the Identity System. However, you can modify any of the custom styles you have created.

To change a style

1. Modify the corresponding stylesheets, as discussed in the chapter on designing the user interface with PresentationXML in the *Oracle Access Manager Customization Guide*.
2. When you have completed your changes, copy the stylesheets to each Identity Server and to each WebPass linked to the Identity Server where the stylesheets were modified.

Deleting a Style

You cannot delete the Classic Style, the default style provided by the Identity System, because it is used by the Identity System. However, you can delete any of the custom styles you have created.

To delete a custom style

1. In the Customize Styles page, click the name of a style.
The View Style page appears.

2. Click Delete.
3. When prompted, confirm your deletion.
The Customize Styles page reappears.
4. Delete the new stylesheets from the new style directory in all the other Identity Servers, as well as the WebPass installation area.

Setting the Default Style

You use the Setup Default Style option to choose the default style for applications.

Note: You cannot select a style that has the *Under Construction* status.

To set the default style

1. In the Customize Styles page, click Setup Default style.
The Set Default Style page appears.
2. Click Make Default beside your choice.
3. Click Save.

In the Customize Styles page, the words "Available & Current Default" appear next to the style you selected.

Configuring Multiple Languages for Oracle Access Manager

As discussed in the *Oracle Access Manager Installation Guide*, the English language is installed automatically. However, you can install one or more Oracle-provided Language Packs. Language Packs enable you to provide localized information to end users and administrators in the languages identified in the *Oracle Access Manager Introduction*.

Some display names and attributes appear for end users in Identity System applications (User Manager, Group Manager, and Org. Manager), as well as in administrator applications (Identity System Console, Access System Console, and Policy Manager).

For end-users, Oracle Access Manager 10g (10.1.4.0.1) enables the display of static application data such as error messages, and display names for tabs, panels, and attributes in supported end -user languages. Administrative information can be displayed in only supported administrator languages.

After installation and setup, you must configure languages for use and localize attribute display names, which you can accomplish at the following levels:

- Object Class level
- Tab level
- Panel level
- Search Result Attributes level

Note: Oracle recommends that you configure display names at the Object Class level because this is the highest level. If you choose to configure display names at a lower level, ensure that you provide display names at each level for all languages.

Task overview: Configuring multi-language functionality

1. Install and set up Oracle Access Manager with one or more Oracle-provided Language Packs, as described in the *Oracle Access Manager Installation Guide*.
2. Enable the languages you have installed and want to use, as described in ["Managing Multiple Languages"](#) on page 7-14.
3. Configure the applications to use an installed language by manually entering the display names for labels and attributes. For example:
 - Localize object class attributes, as described in ["Localizing Attribute Display Names"](#) on page 3-20.
 - Localize display names for tabs, group type panels, search result attributes, and reports, as described in ["Configuring User, Group, and Organization Manager"](#) on page 4-1.
 - Localize workflow panel names, as described in ["Localizing Workflow Panels"](#) on page 5-58.
 - Localize tab names to display them in other languages, as described in ["Localizing Tabs"](#) on page 4-5.
 - Localize search result attributes, as described in ["Viewing, Modifying, and Localizing Attributes that Appear in Search Results"](#) on page 4-7.

Selecting a Language for Administrative Pages

If you have installed and configured more than one language, you can determine the language to use for administrative information in the Identity and Access System Consoles, and Policy Manager. You do this from your browser. See your browser's documentation for details.

If administrative pages are requested in a language that is not supported for administrative information, the default language that was selected during product installation is used to display administrative pages.

Language Evaluation Order for End-User Applications

After enabling installed languages, and configuring attributes to use the installed languages, the language used to display application pages to an end user is chosen according to the following evaluation order.

Evaluation Order

1. The language specified in the URL.

The user can specify a language in a URL. For instance, when a user selects the Create User function in the User Manager, he or she can append `lang=fr-fr` to display the User Manager page in French. The application first looks for a language preference specified in the URL for a resource. The user or the administrator can specify a language in the URL by appending `lang=languageTag`, where `languageTag` is a language tag in RFC 1766 format.

The following example returns the Create User Profile page in French:

```
http://localhost/identity/oblix/apps/userservcenter/bin/userservcenter.cgi?program=workflowCreateProfile&tab_id=employees&lang=fr-fr
```

2. The language stored in a parameter called LangCookie in the ObTEMC cookie.

Once you specify the language in a URL as in the previous step, this language is set in the LangCookie parameter. The ObTEMC cookie is created when the user logs in and is maintained for the duration of the user's session. If a URL does not contain the language specification, the application checks the ObTEMC cookie, which lasts for the duration of the session. The ObTEMC cookie can also be set on a form or a page.

3. The language specified in the HTTP header variable, HTTP_OBLIX_LANG.

You can create an authentication or authorization success header variable to contain this value, as explained in the chapters on authentication and authorization in *Oracle Access Manager Access System Administration Guide*. If you want to change the name of the HTTP_OBLIX_LANG header variable, you can do so in the following files:

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

PolicyManager_install_dir/access/oblix/apps/common/bin/globalparams.xml

where *IdentityServer_install_dir* is the directory where the Identity Server is installed, and *PolicyManager_install_dir* is where the Policy Manager is installed.

See the *Oracle Access Manager Customization Guide* for information on globalparams.xml.

4. The value set by the user's Web browser determines the default language. This value is specified in the header variable, Accept-Language.

If the application does not find the HTTP_OBLIX_LANG header variable, it looks for the Accept-Language header variable that is set in the user's browser.

Note: Both the HTTP_OBLIX_LANG and the Accept-Language header variables are configurable. See the *Oracle Access Manager Customization Guide* for information.

5. The default language of the Oracle Access Manager installation.

If the Accept-Language header variable is not found in the user's browser, the application looks in the obnls.xml configuration file for the default language of the Oracle Access Manager installation.

The obnls.xml file is located in the *IdentityServer_install_dir*/identity/oblix/config directory. *IdentityServer_install_dir* is the directory where the Identity Server is installed.

See "[Managing Multiple Languages](#)" on page 7-14 for details.

Configuring Identity Server Settings

Configuring an Identity Server includes specifying the duration of user sessions, specifying email addresses for user feedback, configuring mail servers for notification events, managing the cache, and enabling multiple languages.

You use the Identity System Console to view and modify server settings.

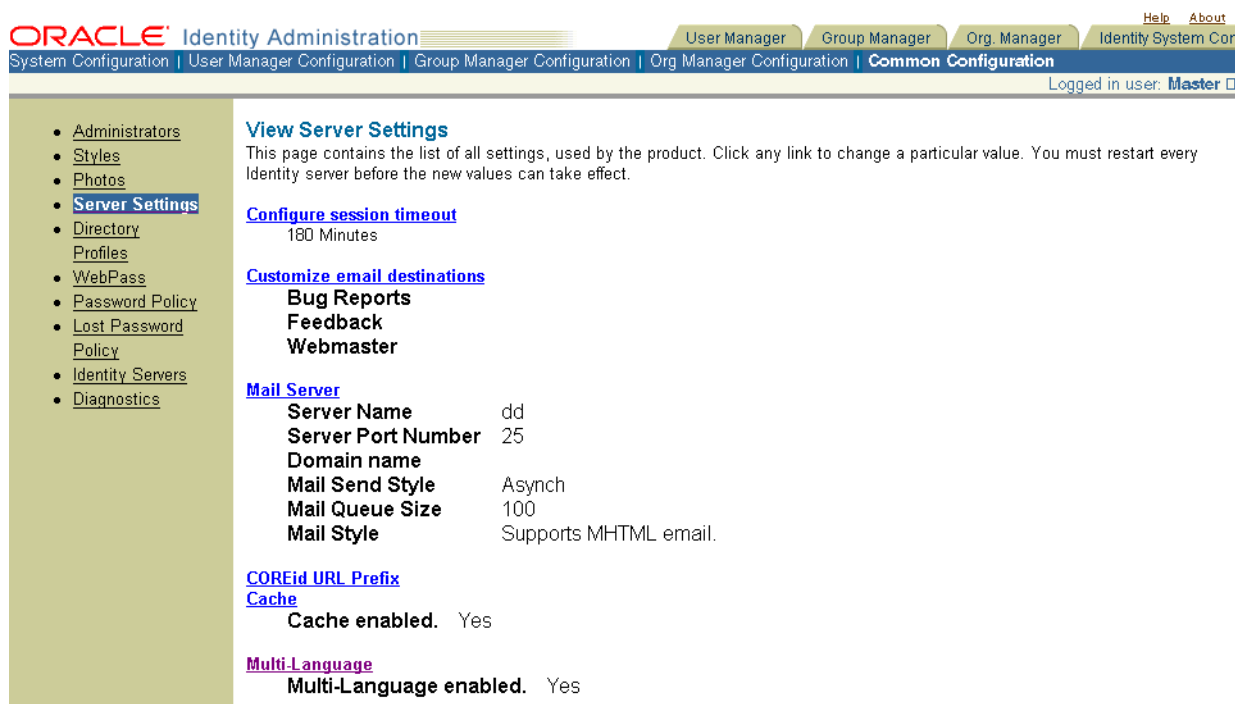
This section discusses the following topics:

- [Configuring Session Timeout](#)
- [Customizing Email Destinations](#)
- [Configuring a Mail Server](#)
- [Managing Caches](#)
- [Managing Multiple Languages](#)

To view or modify server settings

1. In the Identity System Console, select System Configuration, then select Server Settings.

The View Server Settings page appears, which looks something like the one in the following screen shot.



2. To view or modify a value for a setting, click the link for the setting (Mail Server or Multi-Language, for example).
3. Make the changes you want, if any.
4. Click Save to save your changes (or Cancel to exit without saving your changes).
5. Restart the Identity Server for the new values to take effect.

Configuring Session Timeout

Configuring session timeout enables you to specify user-idle session time (in minutes). The user session automatically ends when the specified idle time elapses.

The setting in this page applies to all users and all Identity System applications. You cannot have different settings for different users and applications.

A session timeout does not apply if you are using a Web-server-based login, such as through a WebGate, because the WebGate instance handles the timeout.

Note: Resources protected by Web single sign-on always ignore idle session timeout settings.

To configure the length of a user's Identity System session

1. In the Identity System Console, select System Configuration, then select Server Settings, then click Configure session timeout to display this page.

ORACLE Identity Administration Help About

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | **Common Configuration** | Identity System Co

Logged in user: Master T

- Administrators
- Styles
- Photos
- Server Settings**
- Directory Profiles
- WebPass
- Password Policy
- Lost Password Policy
- Identity Servers
- Diagnostics

Configure Session

With this screen, you can configure the time duration of a user's session is valid without activity for the product if it is not protected by Single Sign On. If you specify "No Timeout", the user can use the product without logging in, as long as the browser is active. Or, you can specify a specific timeout in minutes and specify number of minutes the session time should be refreshed. If Refresh Period is zero means the session time stamp is updated in every request.

IMPORTANT: "No Timeout" implies that a user's session never ends as long as they do not exit their browser.

☐ No Timeout
☒ Idle Session Timeout Minutes Refresh Period Minutes

2. Choose the timeout option:

- **No Timeout:** User sessions continue indefinitely as long as the browser is active.
- **Idle Session Timeout:** The number of minutes to wait before ending an idle session. After this period of inactivity elapses, the user must log in to the application to continue.

There are several reasons for ending an idle session after a predetermined time period. A short session protects users who leave their workstations without locking them, making them vulnerable to unauthorized use.

- **Refresh Period:** Configures how often to update the user session time stamp. A value of 0 (zero) means the session time stamp is updated on every request. Oracle recommends you set this value to 1/4 of the Idle Session Timeout value.

3. Click Save to save your changes (or Cancel to exit the page without saving).

Customizing Email Destinations

Use the Customize Email function to specify email addresses for user feedback. End users access these addresses by clicking About on the side navigation bar, then clicking Submit Admin Feedback or Submit Oracle Feedback.

To customize email destinations

1. In the Identity System Console, select System Configuration, Server Settings.
2. In the Server Settings page, click Customize Email Destinations to display this page.

ORACLE Identity Administration Help About

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | **Common Configuration** | Identity System Co

Logged in user: Master t

- Administrators
- Styles
- Photos
- Server Settings**
- Directory Profiles
- WebPass
- Password Policy
- Lost Password Policy
- Identity Servers
- Diagnostics

Customize Email Destinations

In the Customize E-mail Destinations screen, you can specify the target email addresses for various categories of user input.

Enter the email addresses that will receive Bug Reports and User Feedback.

Email address for Bug Reports

Email address for User Feedback

Specify the email address of the webmaster or COREid administrator. This is the internal address for feedback and requests, not the Oracle address.

Webmaster's email address

- Type email addresses for the following fields:
 - Email address for Bug Reports:** You must change this address if you plan to send it to a person or alias in your organization. This person or department can either solve the problem or contact Oracle for help.
 - Email address for User Feedback:** If users in your company cannot send email outside the local network, you can type an internal address in the Bugs and Feedback fields. Provide the address of a user who is responsible for forwarding the information to Oracle.
 - Webmaster's email address:** Type the email address of the user in your company responsible for administering Oracle Access Manager.
- Click Save to save your changes (or Cancel to exit the page without saving).

Configuring a Mail Server

The Identity System can issue emails alerts during request ticket processing and group management, notification of password expiration, or modification of a Profile attribute. Use the SMTP server configuration function to configure how the Identity System handles these emails.

When configuring a mail server, one of your options is Supports MHTML email. MHTML stands for MIME encapsulation of aggregate documents, such as HTML.

MHTML lets you send an HTML document with in-line graphics, applets, and linked documents in a MIME multipart/related body format. You can provide links to other parts included in the HTML document by using the CID (content-ID) URLs or any other kind of URL. The linked body part is identified in its header by either a content-ID (linked to by CID URLs) or a content-location (linked to by any other kind of URL).

The main difference between HTML and MHTML is that with MHTML, graphics are in-line in the email instead of referenced with a link as in HTML format.

To configure a mail server

- In Identity System Console, select System Configuration, then select Server Settings
- In the Server Settings page, click Mail Server to display this page.

ORACLE Identity Administration Help About

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | **Common Configuration**

Logged in user: Master

- Administrators
- Styles
- Photos
- Server Settings**
- Directory Profiles
- WebPass
- Password Policy
- Lost Password Policy
- Identity Servers
- Diagnostics

SMTP Server configuration

Server Name

Server Port Number

Domain name

Mail Send Style:

☐ Synchronous Mailer.

☒ Asynchronous Mailer. Mail Queue Size

Mail Style:

☐ Supports Text-only email.

☐ Supports Rich HTML email.

☒ Supports MHTML email.

- In the Server Name field, enter your SMTP server name.
- In the Server Port Number field, type the mail server's port number.
- In the Domain name field, type the Web server's domain name.

Note: This field is optional, but specifying the domain name allows the SMTP connection to be set up according to RFC 821.

- Select an option under Mail Send Style:
 - Synchronous Mail**—Sent from the process, such as Modify Attribute, that triggered the email. If an error occurs connecting to the mail server or the server is down, the email is not sent and cannot be regenerated.
 - Asynchronous Mail**—Uses a thread to queue emails from all applications and sends them one at a time. If the mail server cannot be reached, the thread re-sends the email. Queued mails are saved to disk. If you select Asynchronous Mailer, specify the mail queue size.
 - Select an option under Mail Style.
 - Click Save to save your changes (or Cancel to exit the page without saving).

Managing Caches

You can view the contents of the Identity Server cache, load the cache with new information, and clear the memory cache to resolve inconsistencies.

To view Identity System cache details

- In the Identity System Console select System Configuration, then select Server Settings.
- In the Server Settings page, click Cache to display the page.

3. Select the option you want to view the cache contents, or load or clear the memory cache.

See the *Oracle Access Manager Deployment Guide* for more information about managing caches.

Managing Multiple Languages

In new installations, the Multi-Language feature is enabled by default. You can enable, disable, and specify preferred languages in the Identity System Console.

Note: When you upgrade from an older version, the Multi-Language feature is disabled. To enable this feature complete the steps in the following procedure.

To manage a language

1. From the Identity System Console, select System Configuration, then select Server Settings.
2. From the View Server Settings page, select Multi-Language.
The Manage Multiple Languages page appears. Details such as available languages, the order of preference, and whether a language is enabled or not appear on this page.
3. Determine which languages to enable or disable.
 - Enable—Select a language and click Enable to enable it.
 - Disable—Select a language and click Disable to disable it.
4. Click Back to go back to the Server Settings page.

See Also: [""Configuring Multiple Languages for Oracle Access Manager"](#) on page 7-7

Managing Identity Servers

Managing Identity Servers consists of tasks such as adding or deleting Identity Servers, and modifying an Identity Server's parameter values. See the *Oracle Access Manager Installation Guide* for details about installing an Identity Server. To remove a server completely, you must un-install it.

This section includes the following topics:

- [Setting Up Multiple Identity Servers](#)
- [Adding an Identity Server](#)
- [Viewing and Modifying Identity Server Parameters](#)
- [Deleting Identity Server Parameters](#)
- [Managing an Identity Server Service from the Command Line](#)

Setting Up Multiple Identity Servers

The following overview outlines how to set up multiple Identity Servers.

Task overview: Setting up multiple Identity Servers

1. Install the first Identity Server and a WebPass, then set up the Identity System as described in the *Oracle Access Manager Installation Guide*.
2. Add a new Identity Server instance in the Identity System Console, as described in the procedure "[Viewing and Modifying Identity Server Parameters](#)" on page 7-18.
3. Associate the new Identity Server instance with a WebPass and specify the priority as Primary, as described in "[Managing Associations Between Identity Servers and WebPass](#)" on page 7-45.
4. Modify the WebPass instance to set the maximum connections to the appropriate number to communicate with all primary Identity Servers, as described in "[Adding or Modifying a WebPass](#)" on page 7-41.

You must wait at least one minute before performing step 5 to ensure that the WebPass configuration file, `webpass.xml`, is updated with the new instance information. Otherwise, the WebPass instance may not receive the new information and cannot connect to the new Identity Server instance.

The actual amount of time you must wait depends on the interval set for this refresh, which is described in "[Configuring the WebPass Poll Tracking Parameter](#)" on page 7-47.

5. Wait at least one minute before stopping all installed Identity Servers.
6. Install the new Identity Server and indicate that this is not the first Identity Server for this directory server, as described in the *Oracle Access Manager Installation Guide*.

You do not need to update the schema again.

7. Set up the new Identity Server you installed, as explained in the *Oracle Access Manager Installation Guide*.

Adding an Identity Server

When you want to add a new Identity Server instance to your installation, use the following procedure.

To add an Identity Server

1. In the Identity System Console, click System Configuration, then select Identity Servers.
The List all Identity Servers page appears with links to existing Identity Servers.
2. Click the Add button.

The Add a New Identity Server page appears.

ORACLE Identity Administration Help About

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration Logged in user: Master

- Administrators
- Styles
- Photos
- Server Settings
- Directory Profiles
- WebPass
- Password Policy
- Lost Password Policy
- **Identity Servers**
- Diagnostics

Add a new Identity Server

Name	
Hostname	
Port	
Debug	☒ Off ☐ On
Debug File Name	
Transport Security	☒ Open ☐ Simple ☐ Cert
Maximum Session Time (hours)	24
Number of Threads	20
Audit to Database Flag (auditing on/off)	☒ Off ☐ On
Audit to File Flag (auditing on/off)	☒ Off ☐ On
Audit File Name	
Audit File Maximum Size (bytes)	100000
Audit File Rotation Interval (seconds)	7200
Audit Buffer Maximum Size (bytes)	25000
Audit Buffer Flush Interval (seconds)	7200
Scope File Name	/oblix/logs/scopefile.lst
SNMP State	☒ Off ☐ On
SNMP Agent Registration Port	80

3. Fill in the Name through Number of Threads fields as follows:

- Name: Type the name of the Identity Server.
- Hostname: Type the name of the machine on which the Identity Server is running.
- Port: Type the port number of the Identity Server.
- Debug: Specify whether you want to store debug information on the low-level traffic between the Identity Server and the WebPass.
- Debug File Name: Type the name and path of the debug file. The default path is *IdentityServer_install_dir/oblix/logs/debugfile.lst*, where *IdentityServer_install_dir* is the directory where the Identity Server is installed.
- Transport Security: Select the security method used for communications between the WebPass and the Identity Server:

Open: Used if security is not required. No transport security.

Simple: Provides basic security. Communications are encrypted using TLS v1 (Transport Layer Security, RFC 2246). Communicating elements authenticate one another using a password-based mechanism. All elements that use simple security must use the same password throughout the installation. The Identity System provides the certificate that performs the authentication.

Cert: Used if you manage an internal Certificate Authority (CA). Communications are encrypted using TLS v1. In addition, each element, both client and server, must present an X.509 certificate when establishing a connection. The certificate must be provided by a third party such as VeriSign.

Note: Cert—Used if you manage an internal Certificate Authority (CA). Communications are encrypted using TLS v1. In addition, each element, both client and server, must present an X.509 certificate when establishing a connection. The certificate must be provided by a third party such as VeriSign.

- Maximum Session Time (Hours): Type the maximum period of time that a connection between the WebPass and Identity Server can remain open.
When the time expires, the connection closes and a new one is opened.
- Number of Threads: Type the maximum for number of concurrent requests that the Identity Server is allowed.

4. Complete the Audit information for your environment.

- Audit to Database Flag (Auditing On/Off): Selecting On directs audit information to a configured database. Off is the default.
- Audit to File Flag (Auditing On/Off): Selecting On directs audit information to a file whose name you specify in the next field. Off is the default.
- Audit File Name: Type the name of the auditing file where the Identity Server's auditing information is written.

You can specify an absolute or a relative path for the audit file for the Access or Identity Server. To specify a relative path, use either "." or ".." at the beginning of the path. For example, you can input the following relative path:

`./auditfile.lst\`

This relative path creates an audit file in the following location:

`Component_install_dir\oblix\apps\common\bin\auditfile.lst`

Where *Component_install_dir* is the root installation directory for the associated Access or Identity Server.

The following relative path:

`../../../../logs/auditfile.lst`

Creates `Component_install_dir\oblix\logs\auditfile.lst`.

Note: For IIS deployments, in order for your audit files to be visible, you must grant write permissions to the IIS user (the system user running the Web server) for the %TEMP% and %TMP% directories and to the audit file destination directory.

- Audit File Maximum Size (Bytes): Type the number of bytes an audit file can contain. When this amount is reached, the audit file is time stamped and saved, and a new file is created.
- Audit File Rotation Interval (Seconds): Type a number representing the number of seconds that elapse before audit file rotation occurs. Rotation means that the audit file is time stamped and a new file is created. The default is 7200. A setting of 0 means that the audit file never times out, and audit information continues to be added to the file.

5. Fill in the Scope File Name field as follows:

Scope File Name: Type the name of the file that logs bug reports. When a bug report is generated, the information displayed on the page also is logged to a file. This parameter specifies the name of the file for Bug Report or OB_SCOPE messages.

6. Enter details for the SNMP state and agent registration port for your environment.

See ["SNMP Monitoring"](#) on page 12-1 for details.

- SNMP State: Selecting On enables SNMP monitoring. Off is the default.
- SNMP Agent Registration Port: The port that the SNMP agent listens on.

Note: Even if SNMP monitoring is On, to retrieve SNMP statistics you must configure your Network Management Station (NMS) to process the information defined in the Management Information Base (MIB). See details on the SNMP Agent MIB variables, later in this book.

7. Click Save to finish defining your new Identity Server (or Cancel to exit without saving).

Viewing and Modifying Identity Server Parameters

You use the following procedure in the Identity System Console to view or modify parameters.

To view or modify an Identity Server's parameters

1. In the Identity System Console, select System Configuration, then select Identity Servers.

A list of existing Identity Servers appears, displaying each server's name, hostname, and port number.

2. Click the name of an Identity Server to view its parameters.

The Details for Identity Server page appears. The server's parameters are listed on this page.

3. Click Modify.

The Modify Identity Server page appears.

4. Modify the parameters as necessary.

See ["To add an Identity Server"](#) on page 7-15 for information about each parameter.

5. Click Save to save your changes (or Cancel to exit without saving).

Deleting Identity Server Parameters

You use the following procedure in the Identity System Console to delete Identity Server parameters.

Note: If you delete an Identity Server from the Console, an attempt to start that server from a command line will fail because the Identity Server's parameters have been deleted from the Console.

To delete an Identity Server's parameters

1. In the Identity System Console, click System Configuration, then select Identity Servers.
A list of existing Identity Servers appears, displaying each server's name, hostname, and port number.
2. In the List all Identity Servers page, select the Identity Server you want to delete.
3. Click Delete.
4. When asked to confirm your decision, click OK.

The server's name is removed from the list of servers.

Managing an Identity Server Service from the Command Line

You can use the command line tool `config_ois` to manage tasks related to the Identity Server Service in the Windows Service window.

You can install the Identity Server Service and perform other tasks such as starting or stopping the service with the following commands:

Table 7-1 Commands for `config_ois`

Command	Operation
<code>[-i install_dir]</code>	Specifies the installation directory for the Identity Server Service.
<code>-v</code>	Specifies the Service name.
<code>[-a <start, stop, query, install, remove>]</code>	Specifies the action to be performed.

```
C:\IdentityServer_install_dir\identity\oblix\apps\common\bin\
config_ois.exe -q -i c:\IdentityServer_install_dir\identity
-v Identity_ServiceName -a query
```

where `IdentityServer_install_dir` is the directory where Identity Server is installed and `Identity_ServiceName` is the name of the Identity Server Service.

The query displays the following information:

```
Sample_Srv configuration:
Service Type: 0x110
      Start Type: 0x2
Err Control: 0x1
Binary path:
c:\COREid\identity\oblix\apps\common\bin\ois_server.exe
Load order group:
Dependencies:
Dependencies: LocalSystem
```

Managing Directory Server Profiles

When installing components that communicate with a directory server, you specify the directory server with which the component communicates. Each component communicates with the directory for a specific purpose:

- Identity Server: When you install an Identity Server, you designate an LDAP directory server where configuration data is to be stored, and you designate where

user data is stored. The user data may be on the same directory server where the configuration data is stored, or it may be a different directory server.

- **Policy Manager and Access Server:** When installing a Policy Manager or an Access Server, you also designate where user data and configuration data are stored. Configuration data includes system configuration, attribute definition, attribute access control, workflow definition, and workflow instance data. Additionally, you designate where access policy data is stored.

Note: As of release 7.0, you may have user data stored on one directory server type and configuration and policy data stored together on a different directory server type. For data storage details, see the *Oracle Access Manager Installation Guide*.

The following topics provide more information:

- [About LDAP Directory Server Profiles](#)
- [Creating an LDAP Directory Server Profile](#)
- [Viewing an LDAP Directory Server Profile](#)
- [Modifying an LDAP Directory Server Profile](#)
- [Rerunning Setup Manually](#)
- [Adding Database Instances to LDAP Directory Server Profiles](#)
- [Deleting an LDAP Directory Server Instance](#)
- [Working With Multiple Directory Searchbases](#)

About LDAP Directory Server Profiles

For each type of data that Oracle Access Manager requires—configuration data, user data, and policy data—an LDAP directory server profile identifies the precise location of the data. The location of policy and configuration data is also stored in.xml files for the Identity Server, Access Server, and Policy Manager. A directory server profile contains the connection information for one or more directory servers that share the same namespace and operational requirements for Read, Write, Search, and so on. The connection information includes a name, a domain or namespace to which it applies, a directory type, and a set of operations. A default directory server profile is created automatically each time you install the Identity Server and specify new directory server connection information.

You can create additional LDAP directory server profiles for load-balancing and failover. You can create directory server profiles that correspond to the partitions of your directory information tree (DIT). Partitioning can potentially increase performance by freeing CPU cycles to perform read and write operations on a specific portion of the DIT. This can be especially beneficial in installations with multiple directory servers and machines.

You can also create LDAP directory server profiles that specify different operations for master and replicated copies of the DIT. For example, you could specify that write operations take place only in the master, and the replica can accept only read operations.

Note: You must always support read, search, modify, create, and delete operations for the directory server profile containing the Oblix tree. You cannot create a read-only or write-only directory server profile for the Oblix tree. If you change settings for the configuration or policy data directory profile, you need to rerun Identity Server and Policy Manager setup and reconfigure the Access Server. For details, see ["Rerunning Setup Manually"](#) on page 7-29.

Locations for Directory Server Profiles

When you install and configure a new Oracle Access Manager component, various profiles are stored in a configuration directory server as follows:

`obcontainerId=DBAgents, oblix-root-node`

Where `oblix-root-node` is the top-level node for Oracle Access Manager data. If user, policy, and configuration data are stored in the same branch of the directory server, a single profile that contains all of the data can be created for each component that requires a profile. If user, policy, and configuration data are stored in disjoint branches or in different directory servers, separate profiles are required.

[Table 7-2](#) summarizes creation of the directory server profiles. Note that no new profiles are created for additional instances of the Access Server.

Table 7-2 Directory Server Profiles That Are Created During Installation

Installation and Configuration Scenario	Profiles Created for the First Identity Server and Policy Manager	Profiles Created for the Second Identity Server	Profiles Created For the Second Policy Manager
User, configuration, and policy data are stored in one directory server. The user search base is the top node, the configuration base is a descendent of the user search base, and the policy base is a descendent of the configuration base.	New profiles: - Identity Server - Policy Manager - Access Server	Profile for the new Identity Server	No new profiles
Configuration and policy data are stored in same directory, while user data is stored in a separate directory. Or, the policy base is a descendent of the configuration base, and the configuration base is disjoint from the user search base.	New profiles: - Configuration data for the Identity Server - User data for the Policy Manager - Configuration data for the Policy Manager - User data for the Access Server - User data for the Identity Server	New profiles: - User data for the new Identity Server - Configuration data for the new Identity Server	No new profiles

Table 7–2 (Cont.) Directory Server Profiles That Are Created During Installation

Installation and Configuration Scenario	Profiles Created for the First Identity Server and Policy Manager	Profiles Created for the Second Identity Server	Profiles Created For the Second Policy Manager
User, configuration and policy data are located in three different directories servers. Or, user, configuration, and policy data are each in a different disjoint domain. None of these data stores is a descendent of another one.	New profiles: ■ User data for the Identity Server ■ Configuration data for the Identity Server ■ Policy data for the Identity Server ■ User data for the Policy Manager ■ Configuration data for the Policy Manager ■ User data for the Access Server	New profiles: ■ User data for the new Identity Server ■ Configuration data for the new Identity Server	One new policy data profile is created for all new Identity Servers if there is no existing profile for the policy base.

Note: If you install a second Identity Server and notice that an exiting database profile has changed, be sure that you specified a unique ID for the new Identity Server. Using an existing ID can cause profiles to be overwritten.

Creating an LDAP Directory Server Profile

The following screen shot shows the Configure Profiles page in the Identity System Console.

The top portion of Configure Profiles page shows details for the directory server that contains user data and configuration data. The central portion of the page includes links you can use to configure LDAP Directory Server Profiles. The bottom portion of the page includes links to configure RDBMS profiles. For details about RDBMS profiles, see "[Managing RDBMS Profiles](#)" on page 7-36.

ORACLE Identity Administration Help About

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration Logged in user: Master

- Administrators
- Styles
- Photos
- Server Settings
- **Directory Profiles**
- WebPass
- Password Policy
- Lost Password Policy
- Identity Servers
- Diagnostics

Configure Profiles

The following contains the Configuration Base and Search base settings. Click on the link to change a particular value.

Directory Server

Machine	staqh24
Port number	389
Root DN	cn=orcladmin
Root password	<Not Displayed>
Search Base	o=company,c=us
Configuration base	o=Obliz,o=company,c=us
Directory Server Security Mode	Open
Disjoint Search Base	

The following table contains the list of all Directory Profiles. Click on any link to change a particular profile. You must stop and restart every COREid Identity server before the new values can take effect.

Configure LDAP Directory Server Profiles

Name	Name Space	Primary Servers	Secondary Servers
☐ default-ID Server 10.1.3 M3 stagh24 6021	o=company,c=us	default	
☐ OracleContext-ID Server 10.1.3 M3 stagh24 6021	cn=Products,cn=OracleContext	default	
☐ AccessManager setup user profile	o=company,c=us	default	
☐ AccessServer default user profile	o=company,c=us	default	

Configure RDBMS Profiles

Name	Primary Servers	Secondary Servers

Clicking the Directory Server link displays the Directory Server Configuration page. If you change the communication mode for the directory server, or host name or port number, you must also change this information on the Directory Server Configuration page and re-run the setup program. See ["Changing Transport Security Modes"](#) on page 8-1 for details about this type of change.

The middle portion of the Configure Profiles page is titled "Configure LDAP Directory Server Profiles" followed by a list of links to LDAP directory server profiles for user data, configuration data, and policy data. You can click a profile link to review specifications and supported operations for the profile. You can specify all operations or specific operations, as listed in [Table 7-3](#).

Table 7-3 Supported Directory Server Operations

Category	Operation	Comments
	All Operations	All operations are allowed (the default).
Search	Search Entries	The Authenticate User operation allows users to authenticate within the name space of the directory server profile. Selecting this option results in a list of the login pages for Authentication Domain.
	Authenticate User	

Table 7–3 (Cont.) Supported Directory Server Operations

Category	Operation	Comments
Read	Read Entry	This operation enables the directory server profile to support "Read Schema" as well.
Write	Create Entry Modify Entry Delete Entry Change Password	The Change Password operation allows users to change their password over an ADSI or SSL connection while assigning other more frequently used operations like search to different directory server profiles.

The following procedure shows how to create a directory server profile.

To create a directory server profile

1. From the Identity System Console click System Configuration, then click Directory Profiles.
2. Click Add to create a new LDAP directory profile and display the Create Directory Server profile page.

Note: To modify a directory server profile, click the name of the profile in the list under Configure LDAP Directory Server Profiles. In this case, the Modify Directory Server Profile Page appears as described in ["Modifying an LDAP Directory Server Profile"](#) on page 7-29.

ORACLE Identity Administration User Manager Group Manager Org. Manager Identity

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration Logged in as

- Administrators
- Styles
- Photos
- Server Settings
- Directory Profiles**
- WebPass
- Password Policy
- Lost Password Policy
- Identity Servers
- Diagnostics

Create Directory Server Profile

Name*

Name Space*

Directory Type

☐ Sun Directory Server 5.x
☒ Oracle Internet Directory
☐ Novell Directory Services (NDS eDirectory)
☐ IBM Directory Server
☐ Siemens DirX
☐ COREid Data Anywhere
☐ Microsoft Active Directory Application Mode
☐ Microsoft Active Directory (using ADSI)
☐ Use LDAP for Authentication
☐ Microsoft Active Directory
 AD-Change password using: ☐ ADSI ☒ SSL

Dynamic Auxiliary

☐ Yes ☒ No
☒ All Operations
☐ Selected Operations

Operations

Search	☒ Search Entries	☒ Authenticate User
Read	☒ Read Entry	
Write	☒ Create Entry	☒ Modify Entry
	☒ Delete Entry	☒ Change Password

Used By

☒ All COREid Components
☐ Identity servers

All servers
 ID_Server_10.1.3_M3_staqh24_6021

☐ Access servers

All servers
 M3_AAA_staqh24
 dummy_Access_Server

☐ Access Managers

Note: Fields marked with an * are required.

- In the Name field, enter a name for the directory server profile.

This name is for informational purposes only. The Identity System uses the naming convention default *<Identity Server id>* for all default directory server profiles automatically created during Identity System installation.

- In the Name Space field, enter the searchbase for the directory server profile.

Note: Use caution that this namespace does not overlap with other directory server profile namespaces. Overlapping namespaces result in duplicate entries. Exceptions to overlapping name spaces include a directory server profile for a Microsoft Active Directory sub-domain, and the directory server profile containing the configuration DN.

- Select the type of directory server.

Configuring Global Settings 7-25

Siemens DirX and Sun: When using either Siemens DirX or Sun (formerly iPlanet) exclusively, you have the option to store data either separately or together, as discussed in the *Oracle Access Manager Installation Guide*.

Oracle Data Anywhere: Requires integrating with the Oracle Virtual Directory Server (VDS).

Oracle Data Anywhere is a data management layer that aggregates and consolidates user data from multiple sources (including RDBMS and LDAP directories) into a virtual LDAP tree that can be managed by the Identity System and used to support authentication and authorization using the Access System.

The LDAP directory branches containing configuration and policy data must reside on one or more directory servers other than the one hosting VDS or user data. Identity System applications only recognize configuration and policy information that resides outside the VDS virtual directory.

Caution: Before installing for use with Oracle Data Anywhere and VDS, be sure to read the chapter about integrating with VDS in the *Oracle Access Manager Integration Guide*.

Active Directory: If you select Active Directory, specify whether to use ADSI (Active Directory Service Interfaces) for change password operations. Selecting the ADSI option implies you do not have to set up an LDAP/SSL connection for password changes. If you do not use ADSI, Oracle Access Manager uses an SSL connection to change the password. See "[Configuring for ADSI](#)" on page B-1.

If you have already set up LDAP/SSL for all other regular operations to the directory server, you do not need to set up the certificate server, import the CA certificate, and so forth. Otherwise, you need to configure LDAP/SSL for the password change.

See the *Oracle Access Manager Installation Guide* for more information.

Dynamic Auxiliary Classes—If you are using dynamic auxiliary classes with Active Directory, select Yes for Dynamic Auxiliary to associate a dynamic auxiliary class with a structural object class in Active Directory 2003.

See "[Deploying with Active Directory](#)" on page A-1 for more information.

Note: You can enable either dynamic or static auxiliary classes in Active Directory 2003.

6. Specify the supported operations for this directory server profile, as listed in [Table 7-3](#).
7. Indicate which servers are to use this profile.
 - **All Oracle Access Manager Components:** Select this option if you want each component server in this installation to share the same profile.
 - **Identity Servers:** Select this option if you want only the Identity Servers to share this profile. If you want a particular Identity Server to use this profile, select the server name from the list box provided.
 - **AAA Servers:** The AAA Server option represents the configuration option for the Access Server. You are prompted to create a database profile whenever you

add a new Access Server. For details about adding an Access Server instance, see the *Oracle Access Manager Access System Administration Guide*.

8. Click Add to associate a directory server instance (database instance) with this profile, and assign the server type as primary or secondary.

See ["To add or modify a database instance for an LDAP directory server profile"](#) on page 7-32 for details.

9. Specify the number of maximum active servers you want (the number of primary and secondary database instances to connect to for load balancing).

- A default value of 1 indicates that no load balancing takes place.
- A value greater than 1 distributes database requests across all database instances, depending on which database instance has the shortest job queue. This ensures that the job is processed as quickly as possible.

For more information on load balancing, see the *Oracle Access Manager Deployment Guide*.

10. Specify the Failover Threshold.

The value specifies the minimum number of primary servers that must be running. If the number of primary servers running is less than the specified number, a failover occurs. It is recommended that this value be the same as the number of maximum active servers. This ensures that failover to any secondary server happens immediately when a primary server goes down.

The default value is 1. This indicates that failover to a secondary server only occurs when there are no primary directory servers to which the Identity Server can connect.

Note: Oracle recommends that this value match the number of maximum active servers to ensure that failover to any secondary server happens immediately when a primary server goes down. For more information on failover and related parameters, see the *Oracle Access Manager Deployment Guide*.

11. In the Sleep For (Seconds) field, enter the number of seconds before the watcher thread wakes up and attempts to reestablish a connection to one or more downed primary servers.

Note: If a primary server is available when failover occurs, the Identity Server will fail over to the primary server first.

12. In the Max. Session Time (Min.) field, specify the number of minutes that the Identity Server keeps a connection to the directory before attempting to reconnect.

The default value is 0 (unlimited). If you see the memory usage rise for the Identity or Access Server or the Policy Manager, Oracle recommends to change this value to 600 (10 hours).

13. If this profile is ready for use, select Enable Profile.

The following screen shot shows the lower half of the configuration page:

ORACLE Identity Administration Help About

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration

Logged in user: **Master**

- Administrators
- Styles
- Photos
- Server Settings
- Directory Profiles**
- WebPass
- Password Policy
- Lost Password Policy
- Identity Servers
- Diagnostics

Dynamic Auxiliary

Operations

Used By

Database Instances

☐ Use LDAP for Authentication

☐ Microsoft Active Directory

AD-Change password using: ☐ ADSI ☒ SSL

☐ Yes ☒ No

☒ All Operations

☐ Selected Operations

Search	☒ Search Entries	☒ Authenticate User
Read	☒ Read Entry	
Write	☒ Create Entry	☒ Modify Entry
	☒ Delete Entry	☒ Change Password

☒ All COREid Components

☐ Identity servers

All servers
ID_Server_10.1.3_M3_stagh24_6021

☐ Access servers

All servers
M3_AAA_stagh24
dummy_Access_Server

☐ Access Managers

Name	Machine	Port number	Server Type
Add			

Maximum Active Servers:

Failover Threshold:

Sleep For (Seconds):

Max. Session Time (Min.):

☒ Enable Profile

Note: The fields marked with an asterisk(*) are required fields.

14. Select Save, Cancel, or Reset as follows:

- Click Save to save your changes.
- Click Cancel to exit this page without saving.
- Click Reset to reset all settings to the default settings.

15. Click OK to confirm your addition.

16. Restart your Identity Servers to enable the new profile.

Viewing an LDAP Directory Server Profile

The middle section of the Configure Profiles page, under the heading Configure LDAP Directory Server Profiles, contains a list of configured directory server profiles.

To view an LDAP directory server profile

- From the Identity System Console, click System Configuration.
- On the System Configuration page, click Directory Profiles.

The Configure Profiles page appears.

The middle section of the page, under the heading Configure LDAP Directory Server Profiles, contains a list of configured directory server profiles.

3. Click the link for the directory server profile that you want to view.

The Modify Directory Server Profile page appears.

Modifying an LDAP Directory Server Profile

There may be occasions when you need to modify an existing LDAP directory server profile.

To modify an LDAP Directory Server Profile

1. From the Identity System Console, select System Configuration.
2. On the System Configuration page, click Directory Profiles.
3. On the Configure Profiles page, click the link for the directory server profile that you want to modify from those listed under the title 'Configure LDAP Directory Server Profiles'.
4. Refer to ["Creating an LDAP Directory Server Profile"](#) on page 7-22 for details about parameters.
5. Make the changes you need, then click Save to confirm them.
6. Restart your Identity Servers to enable the new profile.

Rerunning Setup Manually

You need to rerun the setup after completing any of the following operations on a directory server profile for configuration and policy data:

- Change directory server configuration options in the System Console.
- Create a new directory profile for configuration and policy data.
- Delete a directory profile belonging to configuration and policy data.
- Modify a directory profile for configuration and policy data.
- Add or change a directory instance within a profile.

Note: You also need to rerun setup when you make specific changes (those marked with an asterisk, *) on the Directory Server Configuration page.

Rerunning setup must occur in a specific sequence.

Task overview: Rerunning system setup

1. Rerun Identity System setup, as described in ["Rerunning Identity System Setup"](#) on page 7-30.
2. Rerun Policy Manager setup, as described in ["Rerunning Policy Manager Setup"](#) on page 7-30, if needed.
3. Reconfigure the Access Server, as described in ["Reconfiguring the Access Server"](#) on page 7-31.

Rerunning Identity System Setup

Modifying or removing the status parameter in setup.xml tells the Identity System that installation is not complete and permits you to rerun setup.

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 in the following 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 will 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.
9. Complete the next procedure to rerun Policy Manager setup.

Rerunning Policy Manager Setup

After rerunning setup for the Identity System, if your implementation includes the Access System, you are ready to setup the Policy Manager manually. Modifying or removing the status parameter in setup.xml permits you to rerun Policy Manager setup.

To rerun Policy Manager setup

1. Shut down all but one Policy Manager Web server if there is more than one running.
2. Go to the only remaining running Policy Manager host and open the setup.xml file:
PolicyManager\oblix\config\setup.xml
3. Remove the status parameter (or change the status parameter value from "done" to "incomplete"), and save the file as shown in the following example:

```
<NameValPair ParamName="status" Value="incomplete"></NameValPair>
```
4. Restart the Policy Manager Web server.
5. From your Web browser, launch the Access System Console.
You will see a Setup page similar to the one that appears during the initial Access System setup.
6. Initiate setup again and specify the new information.
7. After completing setup, restart the other Policy Manager Web servers.

The other Policy Managers should pick up the new information.

8. Rerun Access Server, as described in ["Reconfiguring the Access Server"](#) on page 7-31.

Reconfiguring the Access Server

After manually rerunning setup for the Policy Manager, you need to reconfigure the Access Server as indicated in the following procedure. For additional information on using the `configureAAAServer` tool, see the *Oracle Access Manager Access System Administration Guide*.

To reconfigure the Access Server

1. Locate the `configureAAAServer` tool.

For example:

```
AccessServer_install_dir/access/oblix/tools/configureAAAServer
```

2. Use the following command with the `configureAAAServer` tool to set up the Access Server:

```
configureAAAServer install -i AccessServer_install_dir
```

3. Specify new information.
4. Restart your Access Server.

Adding Database Instances to LDAP Directory Server Profiles

A directory server profile contains the bind information for a particular LDAP directory server, including the server name, the host machine, the port, the root DN, and the password. As part of a directory server profile, you can configure a database instance. When you define such a database instance, Oracle Access Manager validates the configured host and port against the supplied bind credential. The directory server corresponding to the database instance must be running when you configure it.

Note: A database instance within an LDAP directory server profile is distinct from a database instance within an RDBMS profile. An RDBMS profile is used to connect Oracle Access Manager to an external, ODBC 3.0-compatible relational database. See ["Managing RDBMS Profiles"](#) on page 7-36.

An LDAP directory server profile consists of one or more database instances, which are used for load balancing and failover. The directory server profile balances the load among its instances according to the maximum number of active servers; it experiences failover among its instances according to the failover threshold.

Note: Reconfiguring the Identity System to point the configuration directories to a new directory server causes `/IdentityServer_install_dir/data/common` to be reset. Specifically, in `workflowdbparams.xml`, the parameter `wfinstancenotrequired=true` is reset to false. After reconfiguring a directory server instance, manually reset the parameter `wfinstancenotrequired` to true.

LDAP Referrals

When you add a directory server instance, you can specify whether or not to enable LDAP referrals. A referral redirects a client request to another server, to find the requested information in another location. A referral contains the names and locations of objects.

If you choose to enable LDAP referrals when you add a directory server instance, you need to set the `enableLDAPReferral` parameter to true in the following file:

```
install_dir\oblix\data\common\ldapconfigdbparams.xml
```

Where *install_dir* is the installation directory for the Policy Manager, Access Server, or Identity Server.

The following is an example of this file for Active Directory:

```
BEGIN:vCompoundList
    specialAttrs:
    BEGIN:vNameList
        userPassword:( 2.5.4.35 NAME 'userPassword' DESC
'Standard Attribute' SYNTAX '1.3.6.1.4.1.1466.115.121.1.5' )
        sAMAccountName:( 1.2.840.113556.1.4.221 NAME 'sAMAccountName' DESC
'sAMAccountName' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )
    END:vNameList
    useOIDNamingAttribute:false
    dynamicAuxiliary:false
    enableLDAPReferral:true
END:vCompoundList
```

To add or modify a database instance for an LDAP directory server profile

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 Add button (to edit/modify an existing database instance, select it from the list of database instances).

The Create Database Instance (or Modify Database Instance) page appears.

Note: The fields for the Modify Database Instance page for an LDAP Directory Server Profile differ from those for the Modify Database Instance page for an RDBMS. See ["Adding or Modifying an RDBMS Database Instance"](#) on page 7-39 for details.

ORACLE Identity Administration User Manager Group Manager Org

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Conf

- Administrators
- Styles
- Photos
- Server Settings
- Directory Profiles**
- WebPass
- Password Policy
- Lost Password Policy
- Identity Servers
- Diagnostics

Create Database Instance

Name*	
Machine*	
Port number*	389
Root DN*	
Root password*	
Time Limit	0
Size Limit	0
Flags	☐ SSL ☐ Referral ☐ Fast Bind (only for AD on Windows S
Secure Port number	636
Initial Connections	1
Maximum Connections	1

Note: The fields marked with an asterisk(*) are required fields.

Changes made to this DB Instance require that you save the DB profile too.

5. Fill in the fields as follows:

- **Name:** Enter a name for the directory server instance.
- **Machine:** Enter the name of the computer hosting the directory server instance.
- **Port Number:** Enter the port number for the directory server.
- **Root DN:** Enter the Root DN (bind DN) of the directory server user with administrative privileges.
- **Root Password:** Enter the password of the directory server user with administrative privileges.
- **Time Limit:** Specify the maximum amount of time allowed for a request to the directory server.

The default value is 0 seconds, which means that the server determines the time. The database-instance setting takes precedence over this setting.

- **Size Limit:** Specify the maximum number of entries the directory server can return for a search operation.

The default value is 0 entries, which indicates that the server determines the number.

Flags: Select either of the following:

- **SSL:** Directory server processes that use SSL. This requires initial certificate configuration. Refer to your directory server documentation for information.
- **Referral:** Specifies whether the directory server profile should trace LDAP referrals for this directory server. The same bind credentials (Root DN and password) are used to log in to the referral server.
- **Fast Bind (only for AD on Windows Server 2003):** Authenticates a user name and password without returning a security token, unlike simple

bind. This is faster than simple bind for applications that only perform authentication.

- **Secure Port:** Specify the port where you access the directory server.
Leave this field blank if you are not using SSL or if you are using Active Directory with ADSI for change password.
- **Initial Connections:** Specify the initial number of connections used to connect to the directory server.

These connections are shared among all user requests. The minimum is 1.

- **Maximum Connections:** Specify the maximum number of connections allowed to the directory server.

The default is 1. A different DB agent is used for different types of operations. Note that the Maximum Connections field is implemented for a specific agent. The grand total of connections that can be opened can be much higher than the value specified in this field. See the information on configuring the directory connection pool in the *Oracle Access Manager Deployment Guide* for details.

6. Click Save to save your settings.

Clicking Cancel exits without saving, and clicking Reset reverts to the last saved settings.

Deleting an LDAP Directory Server Instance

You may want to remove an LDAP directory server instance.

To delete a directory server instance for an LDAP directory server profile

1. From the Identity System Console select System Configuration, then click Configure Directory Options.

The Configure Directory Server Profiles page appears. All the directory server profiles are listed on this page.
2. Click the directory server profile to which you want to add an instance.

The Modify Directory Server Profile page appears.
3. In the Modify Directory Server Profile page, select the Database Instance that you want to delete.
4. Click Delete.

The directory server instance is deleted.

Working With Multiple Directory Searchbases

Some directories such as the Oracle Internet Directory allow you to configure multiple searchbases, sometimes referred to as disjoint searchbases or realms. These disjoint searchbases or realms consist of nonoverlapping directory trees, for example:

- o=company,c=us
- o=oracle,c=us

If you want Oracle Access Manager to manage data in more than one of these searchbases, you must configure the Identity and Access Systems separately for each searchbase.

The following procedures assume that you have already defined the disjoint searchbases (or realms) in your directory.

To configure the Identity System to work with a disjoint searchbase

1. From the Identity System Console click System Configuration, then click Directory Profiles.
2. Add a separate directory server profile for each new disjoint searchbase that you want to support.

See ["Creating an LDAP Directory Server Profile"](#) on page 7-22 for details. Ensure that the name space that you provide matches exactly the name space in the directory.
3. Restart the Identity Server and the Web server running the Identity Server.
4. Return to the Directory Profiles page: From the Identity System Console click System Configuration, then click Directory Profiles.
5. Click the Directory Server link on the Directory Profiles page.
6. In the Disjoint Search Base field, enter the name space of the first disjoint searchbase.

This must be identical to the name space provided in the directory server profile.

7. Click Add to configure additional disjoint searchbases.
8. For each new disjoint searchbase, configure new permissions for users who have entries in the searchbase.

See ["Allowing Users to View and Change LDAP Data"](#) on page 4-22 for details.

9. Add Identity Administrators and Delegated Identity Administrators for each disjoint searchbase.

See ["Specifying Identity System Administrators"](#) on page 2-1 for details.

10. Open the following file in a text editor and ensure that the value for the `whichAttrIsLogin` parameter in this file matches the user attribute in the directory:

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

To configure the Access System to work with a disjoint searchbase

1. Complete the steps in the procedure ["To configure the Identity System to work with a disjoint searchbase"](#) on page 7-35.
2. Open the following file in a text editor and ensure that the value for the `whichAttrIsLogin` parameter in this file matches the user attribute in the directory:

PolicyManager_install_dir/oblix/apps/common/bin/globalparams.xml

3. In the Access System Console, create an authentication scheme that uses the appropriate credential mapping parameters.

The syntax for the authentication scheme is as follows:

```
obMappingBase="%domain%",obMappingFilter="(&(&(objectclass=
objectclassname)(loginattribute=%userid%))(|(!(obuseraccountcontrol=*))
(obuseraccountcontrol=ACTIVATED)))",obdomain="domain"
```

Where *objectclassname* is the name of the Person object class, for example *gensiteorgperson* and *loginattribute* is the name of the login attribute for the Person object class, for example, *genuserid*. See *Oracle Access Manager Access System Administration Guide* for details.

For example, if your disjoint searchbases use the *gensiteorgperson* as the Person object class and *genuserid* as the login attribute, you might create an authentication scheme as follows:

```
obMappingBase="%domain%",obMappingFilter="(&(&(objectclass=gensiteorgperson)
(genuserid=%userid%))(|(! (obuseraccountcontrol=*)) (obuseraccountcontrol=
ACTIVATED))) ",obdomain="domain"
```

4. In the Policy Manager, modify the relevant authentication rules to use the appropriate authentication scheme.

See *Oracle Access Manager Access System Administration Guide* for details.

5. In the Policy Manager, configure the */access* and */identity* policy domains to return the value *obuniqueID* in the HTTP_OBLIX_UID header variable upon successful authentication.

In the authentication rules for these policy domains, configure the following authentication success action. Note that the *obuniqueid* attribute returns any value configured for the specific login attribute used by each searchbase in the directory:

Type—HEADervAR

Name—HTTP_OBLIX_UID

Return Attribute—obuniqueid

Managing RDBMS Profiles

Oracle Access Manager connects to external, ODBC 3.0-compatible relational databases through RDBMS profiles. Currently user access profile reporting and the audit-to-database features make use of RDBMS profiles. Each profile can contain multiple database instances for failover if the primary instance of the database goes down.

Note: RDBMS profiles contain database instances. These database instances are distinct from database instances that are configured as part of LDAP directory server profiles. The latter are used to load balance and failover for LDAP directories.

This section discusses the following topics:

- [Adding or Modifying an RDBMS Profile](#)
- [Adding or Modifying an RDBMS Database Instance](#)

Adding or Modifying an RDBMS Profile

The steps to either add or modify an RDBMS profile are similar and are described in the following procedure. The fields you complete are described in [Table 7-4](#).

Table 7–4 Field Descriptions for Adding or Modifying an RDBMS Profile

Field	Description
Name	Choose a self-explanatory name for your RDBMS profile.
Database Connection Type	Choose the connection type that your database uses, if you have configured auditing to a database. See "Auditing" on page 11-1 for details.
Used By	Check the box corresponding to the feature for which you will be using the RDBMS profile. Currently, the choices are user access privilege reporting and auditing to database.
Database Instances	You can create multiple copies of the database for use in failover as follows: ■ To add a database instance and click Add. When the Create Database Instance page appears, complete the fields marked by asterisks. For field details, see "To add or modify a database instance for an RDBMS profile" on page 7-39. ■ To modify an existing database instance, select it from the database instance list. ■ To set the server type for the database instance, select Primary or Secondary from the list. ■ To delete a database instance, check the box next to the instance you want to delete, then click Delete.
Maximum Active Servers	This is the maximum number of servers that can be connected to the relational database at any given time.
Failover threshold	When the number of connected primary servers falls to this number, failover occurs.
Sleep For (Seconds)	Once a connection fails, this many seconds must elapse before failover takes place.
Max. Session Time (Min.)	The connection to the database is discarded after this many minutes, even if it is functioning, and a new connection is established.
Enable Profile	Make sure to check this box if you want the profile to be active.

To add or modify an RDBMS profile

1. From the Identity System Console select System Configuration, then click Directory Profiles.

The Configure Profiles page appears. All the directory server profiles are listed on this page.

ORACLE Identity Administration Help About Log

User Manager Group Manager Org. Manager Identity System Console

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration

Logged in user: Master Admin

- Password Policy
- Lost Password Policy
- Directory Profiles**
- Identity Servers
- WebPass
- Server Settings
- Diagnostics

Configure Profiles

The following contains the Configuration Base and Search base settings. Click on the link to change a particular value.

Directory Server

Machine	avanur
Port number	3334
Root DN	cn=Directory Manager
Root password	<Not Displayed>
Search Base	o=company,c=us
Configuration base	o=Obliv, o=company, c=us
Directory Server Security Mode	Open
Disjoint Search Base	

The Configure RDBMS Profiles section is at the bottom of the Configure Profiles page.

Configure RDBMS Profiles

Name	Primary Servers	Secondary Servers
☐ Profile1	Instance1	

- Select from the RDBMS Profile list the name of the profile you want to edit (or click Add to create a new profile).

ORACLE Identity Administration Help About Log

User Manager Group Manager Org. Manager Identity System Console

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration

Logged in user: Master Admin

- Password Policy
- Lost Password Policy
- Directory Profiles**
- Identity Servers
- WebPass
- Server Settings
- Diagnostics
- Administrators
- Styles
- Photos

Create RDBMS Profile

Name*

Database Connection Type* ☒ ODBC ☐ OCI

Used By* ☐ Reporting ☐ Auditing ☐ MIIIS

Name	Server Type
Add	
Maximum Active Servers	1
Failover Threshold	1
Sleep For (Seconds)	60
Max. Session Time (Min.)	0

☒ Enable Profile

Note: The fields marked with an asterisk(*) are required fields.

- Complete or modify the fields on the Add RDBMS Profile (or Modify RDBMS Profile) page, as described in Table 7-4.
- When you are satisfied with the information in the fields, click Save to commit the changes.

Adding or Modifying an RDBMS Database Instance

The steps to create or modify a database instance for an RDBMS database profile are so similar that they are combined in the following procedure. In either case, you must complete fields for the information in [Table 7–5](#).

Table 7–5 *Field Descriptions to Add or Modify a Database Instance in an RDBMS Profile*

Field	Description
Name	The name of the database instance
DSN Name or Global Database Name	If you configure database auditing with an ODBC connection type, the DSN Name field appears. It identifies a unique data-source definition to all the clients that access a given data source. (The term DSN is often used incorrectly to denote an entire ODBC data-source definition.) If you configure database auditing with an OCI connection type, you specify a Global Database Name (GDN) in the database instance definition. See "About RDBMS Profiles for Database Auditing" on page 11-14 for details.
User name	The name of the administrator with access privileges to this database instance
Password	The password for this database instance
Time Limit	The number of minutes after which the connection to the database is broken and then replaced with a fresh connection
Size Limit	The maximum size of the database
Initial Connections	The number of primary and secondary servers connected to this database instance when it is initialized
Maximum Connections	The total number of primary and secondary Access Servers that can be connected to this database instance

To add or modify a database instance for an RDBMS profile

1. From the Identity System Console select System Configuration, then click Directory Profiles.
The Configure Directory Server Profiles page appears.
2. In the Configure RDBMS Profiles section, click Add to create a RDBMS profile, or select from the list the name of the RDBMS profile you want to modify.
Depending on your selection, either the Add RDBMS Profile or Modify RDBMS Profile page appears.
3. In the Database Instances section, click the Add button to create a new instance (or select from the list the name of the instance you want to edit).
4. Complete the fields on the Modify Database Instance or Add Database Instance page.
Field descriptions appear in [Table 7–5](#).

ORACLE Identity Administration Provisioning

User Manager Group Manager Org. Manager Identity

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration

Logged in as

- ◆ Password Policy
- ◆ Lost Password Policy
- ◆ **Directory Profiles**
- ◆ Identity Servers
- ◆ WebPass
- ◆ Server Settings
- ◆ Diagnostics
- ◆ Administrators

Create Database Instance

Name *	
DSN Name *	
Database Name	
User name	
Password	
Time Limit	0
Size Limit	0
Initial Connections	5
Maximum Connections	5

Note: The fields marked with an asterisk(*) are required fields.

Changes made to this DB Instance require that you save the DB profile too.

- Click Save to commit the changes when you are satisfied with the information in the fields on the page.

Configuring WebPass

You first install WebPass after installing the Identity Server. After you set up the Identity System, you can install and configure multiple WebPass instances. Each WebPass instance is installed and configured separately. When a WebPass instance is installed, you supply several required parameters. A Master Administrator can modify these parameters and supply additional information, such as the failover threshold, in the Identity System Console.

When a user requests access to a Web server resource, WebPass redirects the request to an Identity Server, which then checks the user's identity through the directory server. You must configure a WebPass plug-in for each Web server.

See the *Oracle Access Manager Installation Guide* for information about installing WebPass. Topics in this section include:

- [Viewing a Configured WebPass](#)
- [Adding or Modifying a WebPass](#)
- [Removing a WebPass](#)
- [Modifying a WebPass from a Command Line](#)
- [Managing Associations Between Identity Servers and WebPass](#)
- [Disassociating a WebPass from an Identity Server](#)
- [Configuring the WebPass Poll Tracking Parameter](#)

Viewing a Configured WebPass

WebPass configuration occurs using the Identity System Console, Configure WebPass function.

To view a configured WebPass

- From the Identity System Console select System Configuration, then select WebPass.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. To view information about a WebPass, click the link for the WebPass.

The Details for WebPass page appears. All the information about the WebPass instance is listed on this page.

Adding or Modifying a WebPass

Adding a new WebPass involves adding the instance in the Identity System Console, installing WebPass on the Web server host, and updating the Web server configuration to establish communications between the WebPass and the Web server. Use the following procedure to add the instance. See the *Oracle Access Manager Installation Guide* for other details.

To add a WebPass

1. From the Identity System Console click the System Configuration sub-tab, then click WebPass in the left navigation pane.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. From the Configure WebPass page, click Add.

The Add a new WebPass page appears.

3. In the Name field, type a name for this WebPass instance.

Note: You cannot change the name you save with this instance. To change the name, delete this instance and reconfigure it with a different name.

4. In the Hostname field, type the name of the Web server instance hosting this WebPass.

5. In the Web Server Port field, type the port number the Web server instance is listening to.

The maximum value is 65535.

6. In the Maximum Connections field, specify the maximum number of connections this WebPass opens to Identity Servers.

The minimum number of connections is 1. You may want to specify more connections for load balancing and failovers.

7. In the Transport Security field, you can modify the security mode that was specified when Oracle Access Manager was installed.

The transport security mode specifies the degree of security during communications between the WebPass and the Identity Server. See ["Changing Transport Security Modes"](#) on page 8-1 for details.

The supported transport security modes are as follows:

- Open: No transport security.
- Simple: Provides basic security. Communications are encrypted using Transport Layer Security, RFC 2246 (TLS v1). Communicating elements authenticate one another using a password-based mechanism. All elements

that use simple security must use the same password throughout the installation. Oracle Access Manager provides the certificate that performs the authentication.

- **Cert:** Used if you manage an internal certificate authority (CA). Communications are encrypted using TLS v1. Both client and server must present an X.509 certificate from a third party (such as VeriSign) when establishing a connection.

Note: Your Identity Servers and WebPasses must use the same transport security mode. Repeat these steps as necessary for each installed component.

8. In the Maximum Session Time (hours) field, specify the maximum period of time in hours before the connection between the WebPass and Identity Server is closed and a new one is opened.

9. In the Failover Threshold field, specify the minimum number of connections to Primary Identity Servers.

If this number cannot be met using primary servers, WebPass attempts to do so using secondary servers. For example, if you type 4 in this field, and the number of available connections to primary Identity Servers falls to 3, WebPass attempts to open a connection to a secondary server.

For details about configuring failover between WebPass and the Identity Server, see the *Oracle Access Manager Deployment Guide*.

10. In the Identity Server Timeout Threshold field, specify how long (in seconds) the WebPass attempts to contact a non-responsive Identity Server before it considers it unreachable and attempts to contact another.

Oracle recommends a timeout value of -1, which indicates that there is no timeout. If you set this value to a low number, you run a risk of the the socket connection closing before a reply from the Identity Server is received. If the Identity Server takes longer to service a request than the value of the timeout threshold, the WebPass abandons the request and attempts to contact another Identity Server with the request until the Identity Server. However, if all Identity Servers take longer to service the request than the value of the timeout threshold, the retries continue until the servers are shut down. You can configure a limit on the number of retries that the WebPass performs for a non-responsive server using the `client_request_retry_attempts` parameter in `globalparams.xml`. See ["To configure the number of retries to a non-responsive Identity Server"](#) on page 7-46 for details.

11. In the Sleep For (seconds) field, specify the interval at which WebPass checks its connection with the Identity System.

Along with checking for a minimum number of connections, the same check also tries to reestablish primary server connections when secondary connections are currently in use because the failover threshold was not met.

12. Click Save to add the WebPass plug-in (or Cancel to exit this page without saving).

If you click Save, this WebPass plug-in appears on the List all WebPasses page.

13. Associate the WebPass plug-in with one or more Identity Servers, as described in ["Managing Associations Between Identity Servers and WebPass"](#) on page 7-45.

To modify a WebPass

1. From the Identity System Console, click the System Configuration sub-tab, then click WebPass in the left navigation pane.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. In the List all WebPasses page, click the name of the WebPass that you want to modify.

The Details for WebPass page appears.

3. Click Modify.

The Modify WebPass page appears.

4. Modify the parameters as needed.

Note: See ["Adding or Modifying a WebPass"](#) on page 7-41 for more information on the parameters you will modify.

5. Click Save to save your changes (or Cancel to exit this page without saving).

Removing a WebPass

Removing a WebPass means that you remove it from the list of configured WebPass instances. To delete a WebPass from the Web server instance, you must uninstall it.

To remove a WebPass

1. From the Identity System Console, click the System Configuration sub-tab, then click WebPass in the left navigation pane.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. In the List all WebPasses page, select the checkbox next to the WebPass instance that you want to remove.

3. Click Delete.

4. When prompted, click OK to confirm the action.

The WebPass instance is removed from the list of configured WebPasses.

Note: If you remove a WebPass instance in the Identity System Console but do not run the uninstall program, it will be added to the directory server again when you restart the Web server.

Modifying a WebPass from a Command Line

Occasionally you may need to modify the parameters of a WebPass. You modify some parameters, such as Maximum Session Time and Failover Threshold, through the Identity System Console. You can use the command line tool `setup_webpass` to change other parameters, such as the host machine name and transport security mode.

Typically, you use the command-line tool to change the transport security mode. This tool can be used in both Windows and Solaris installations.

To modify a WebPass through the command line

1. Navigate to

`WebPass_install_dir\identity\oblix\tools\setupWebPass`

where `WebPass_install_dir` is the directory where WebPass is installed.

2. From the `setupWebPass` directory, run the `setup_webpass` tool.

You can specify parameters using the commands in [Table 7-6](#).

Table 7-6 Commands for `setup_webpass`

Command	Operation
<code>[-i install_dir]</code>	Specifies the installation directory for the WebPass
<code>[-q] [-n WebPass_ID]</code>	Specifies the WebPass ID
<code>[-h Identity_Server_Host_Name]</code>	Specifies the machine name where the Identity Server is installed
<code>[-p Identity_Server_port_#]</code>	Specifies the port number of the machine where the Identity Server is installed
<code>[-s open simple cert]</code>	Specifies the transport security mode
<code>[-P simple cert mode password]</code>	Specifies the password for simple or cert transport security mode
<code>[-c request install]</code>	Specifies a certificate request or installation

To reconfigure transport security mode through the command line

1. To reconfigure a WebPass transport security mode, run the following command from the command line:

```
setup_webpass -i WebPass_install_dir -m
```

2. Select the transport security mode for WebPass:

If you select Open	If you select Simple	If you select Cert
The transport security mode is reconfigured to run in Open mode.	The system prompts you for the password.	■ The system prompts you for the certificate password. Enter the password at the prompt. ■ The system prompts you to specify whether you want to request a certificate or install a certificate. ■ If you specify a certificate request, the system prompts you for the following organization information: Country name State or Province Locality Organization name Organizational unit Common name (for example, HostName.DomainName.com) Email address

- For Cert mode, after you enter the information, a certificate request is generated and placed in *IdentityServer_install_dir\identity\oblix\config\ois_req.pem* file, where *IdentityServer_install_dir* is the directory where the Identity System is installed

You must have this certificate request signed by the Certificate Authority.

- If you specify a certificate installation, the system prompts you for the full paths to the location of the Certificate Key file, the Certificate file, and the Certificate Chain file.

After you specify the paths, the transport security mode is reconfigured. See ["Changing Transport Security Modes"](#) on page 8-1 for details.

To change the transport security mode password

1. Run the following command from the command line:

```
setup_webpass -i WebPass_install_dir -k
```

2. Enter the following information:

- The old password
- The new password
- Reconfirm the new password

The password is changed.

Managing Associations Between Identity Servers and WebPass

You must select one or more Identity Servers to receive requests from a WebPass. A single Identity Server can be associated with multiple WebPasses. You can view a list of primary and secondary Identity Servers that are associated with a WebPass instance. You can also modify the number of connections that have been configured between an Identity Server and a WebPass for load balancing and failover purposes, as described in the following procedures:

- [To view Identity Servers associated with a WebPass](#)
- [To modify an Identity Server's connections to a WebPass](#)
- [To associate an Identity Server with a WebPass](#)

To view Identity Servers associated with a WebPass

1. From the Identity System Console, click the System Configuration sub-tab, then click WebPass in the left navigation pane.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. Click the link for a WebPass.

The Details for WebPass page appears.

3. Click the List Identity Servers button.

A page appears that lists the primary and secondary servers configured for the WebPass.

4. Click the link for an Identity Server to view details for it.

The Details for Identity Server page appears.

To modify an Identity Server's connections to a WebPass

1. From the Identity System Console click the System Configuration sub-tab, then click Identity Servers in the left navigation pane.

The List all Identity Servers page appears. From this page you can add, modify, or delete a WebPass.

2. Click the link for the appropriate server.

The Details for Identity Server page appears.

3. In the Details for Identity Server page, click Modify.

The Modify Identity Server page appears, listing the Identity Server details.

4. Change the value in the Number of Threads field, as needed.

5. Click Save to save your changes.

6. Restart the Identity Server.

To associate an Identity Server with a WebPass

1. From the Identity System Console, click the System Configuration sub-tab, then click WebPass in the left navigation pane.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. Click a link for the appropriate WebPass.

3. In the Details for WebPass page, click List Identity Servers.

The next page lists the Primary and Secondary servers associated with the WebPass.

4. Click Add.

The Add a new Identity Server to the WebPass page appears.

5. In the Select Server list, select an Identity Server.

6. Indicate whether this Identity Server is a Primary or Secondary server.

This information is required for load balancing and failovers.

7. In the Number of connections box, specify the maximum number of connections the WebPass instance opens to this Identity Server.

The minimum is 1. You may want to add more connections for load balancing and failover.

8. Click Add to associate this Identity Server with the WebPass.

To configure the number of retries to a non-responsive Identity Server

1. Open the following file in a text editor:

identity_installdir/apps/common/bin/globalparams.xml

2. Add the `client_request_retry_attempts` parameter to this file.

The following is an example:

```
<SimpleList>
  <NameValPair
    ParamName="client_request_retry_attempts" Value="3">
  </NameValPair>
```

```
</SimpleList>
```

If the Identity Server takes longer to service a request than the value of the Identity Server Timeout Threshold, the WebPass continues to try to contact the Identity Server with the request. This parameter enables you to set a limit on the number of retries that the WebPass attempts. This parameter takes an integer value. The default of -1 means that an unlimited number of retries are possible.

Disassociating a WebPass from an Identity Server

Occasionally, you may need to disassociate a WebPass instance from an Identity Server. For example, the machine resources in your division may be reallocated. In this scenario, the associations between WebPass and an Identity Server may no longer be valid. So you must disassociate them from each other. If you do not disassociate them, WebPass continues to poll for the Identity Server and slows down the Web server's performance.

Note: You cannot disassociate an Identity Server if it is the only primary server configured for a WebPass.

To disassociate an Identity Server from a WebPass

1. From the Identity System Console, click the System Configuration sub-tab, then click WebPass in the left navigation pane.
2. Click an existing WebPass.
3. Click List Identity Servers.
4. Select the check box next to the Identity Server you want to disassociate.
5. Click Delete.
6. When prompted, click OK to confirm your decision.

The WebPass instance will no longer communicate with the Identity Server.

Configuring the WebPass Poll Tracking Parameter

WebPass sends an update request to the Identity Server based on an interval specified in the webpass.xml file. The default value is 60 seconds.

When responding to an update request, the Identity Server reads configuration information that was specified using the Identity System Console and stored in the directory. If the `refresh` parameter in webpass.xml is set to `true`, the Identity Server updates webpass.xml with the latest information, which includes the latest settings for:

- **Maximum Connections:** The maximum number of connections this WebPass opens to Identity Servers
- **Maximum Session Time (Hours):** The maximum period of time that a connection between the WebPass and Identity Server can remain open
- **Timeout Threshold:** The amount of time the WebPass attempts to contact a non-responsive Identity Server before it considers it unreachable and attempts to contact another
- **Failover Threshold:** If the number of primary servers running is less than the specified number, a failover occurs

- Sleep For (Seconds): The interval before failover occurs after a connection fails
- Primary server hosts, ports, and number of connections
- Secondary server hosts, ports, and number of connections

In a multi-process environment, many processes may poll the Identity Server. For example, if 100 processes are running, each process could result in an update to the webpass.xml file. However to eliminate any possible overload, Oracle Access Manager ensures that only one process can acquire the poll tracking information for a particular time. If any configuration changes are detected, the webpass.xml file is updated for the specific time.

Beginning with Patchset 1, 10g (10.1.4.2.0), administrators can configure the `PollTrackingRefreshInterval` parameter in the webpass.xml file. The process that acquires the poll-tracking information triggers an update to the webpass.xml file only if one or more configuration specifications have changed in the directory. Otherwise, the process compares the last webpass.xml file time stamp to the current webpass.xml file time stamp.

A time stamp difference indicates that an administrator has modified the webpass.xml file. In this case, the process reads the `PollTrackingRefreshInterval` parameter and waits until the next interval specified by the value. All other processes read the webpass.xml file on the host computer and wait until the next interval specified by the `PollTrackingRefreshInterval` parameter value.

The polling interval is not specific to any Web server. However, the impact of the interval that is specified using this parameter is more obvious and advantageous with multi-process Web servers such as Apache.

Setting the `PollTrackingRefreshInterval` parameter to an interval greater than 60 seconds ensures that the Identity Server obtains the latest configuration information, and enables the directory server and webpass.xml file to remain in sync. Increasing the value to an interval greater than 60 seconds eliminates repetitive Identity Server updates to webpass.xml and eliminates the need for numerous Web server restarts.

The `PollTrackingRefreshInterval` parameter is shown below as it appears in the webpass.xml file. You may specify a value as high as you like (up to 32767 seconds, approximately 9 hours). However, a large value may result in a WebPass that does not remain in sync with configuration changes made using the System Console and recorded in the directory.

```
<SimpleList>
  <NameValPair ParamName="PollTrackingRefreshInterval" Value="60" />
</SimpleList>
```

This parameter cannot be configured using the Identity System Console. There is no explicit indication of a change in the `PollTrackingRefreshInterval` value. However, you can verify the behavior by setting `LOGLEVEL_DEBUG1` in the `oblog_config_wp.xml` file. A message is logged each time the WebPass waits for the poll tracking interval. For more information about logging configuration, see [Chapter 10](#).

To update the WebPass poll tracking refresh parameter

1. Locate the webpass.xml file in the WebPass Web component installation directory. For example:

```
oam1014\webcomp\nsapi\identity\oblix\apps\webpass\bin\webpass.xml
```

In the path name, *oam1014\webcomp\nsapi* refers to the directory where the Oracle Access Manager WebPass for a Sun (formerly Netscape/iPlanet) Web server is installed. This portion of the path name is also known as *WebPass_install_dir*.

2. Open the file in an editor and change the `PollTrackingRefreshInterval` as desired. For example:

```
<SimpleList>
  <NameValPair ParamName="PollTrackingRefreshInterval" Value="180" />
</SimpleList>
```

3. Save the file.

The parameter takes affect immediately, based on the value of the interval that you set. No reconfiguration or restart of the Identity Server or the Web server is required.

Configuring Password Policies

Password policies consist of a set of rules that govern the kinds of passwords that users create and the validity period for passwords. Password policies also govern how users are notified of password expiry, how users reset expired passwords, and how users retrieve lost passwords.

You create password policies in the Identity System. These policies apply to users who try to log in to the Identity and Access Systems. These policies also apply to users who try to access resources protected by the Access System, as described in ["Implementing Password Policies in the Access System"](#) on page 7-64.

Password policies control the characteristics and life cycle of a password, including the following:

- Rules for legal passwords.

This includes the minimum number of characters that can be used in a password and what types of characters must be used. For example, you can require both numbers and letters.

You configure password properties from the Identity System Console.

If you want to define additional rules for forming a legal password, an Identity Event API provides an external hook for password policy implementation. See the *Oracle Access Manager Developer Guide* for details.

- Challenge phrases and responses for lost password management.

You can require users to respond to one or more challenge phrases when retrieving a lost password. You can also configure rules for challenge phrases, for example, you can prevent users from providing the same response to more than one challenge. Upon successfully responding to the challenge phrase or phrases, the user is redirected to a password reset page. After resetting the password, the user is logged in.

- Settings for password expiry and password reset.

You can specify a password validity period and notify users of pending password expiration through email or at login time. You can also configure a URL redirect to return users to the originally requested resource after resetting a password.

- Account lockout after incorrect password entry.

You can configure how many times a user can enter an incorrect password in a particular time interval before being locked out.

For the Access System, you can also configure a lockout URL that has no user ID or requested resource information.

- stylesheets for password reset and lost password management pages.
Although the Access System can redirect users to these pages, the pages themselves reside on the Identity System. You can configure different Identity System-provided stylesheets for these pages.
- Unique password policies for individual domains in the directory.
You can create separate sets of password policies for different branches of the directory tree.
- Logs of the most recent successful and unsuccessful login attempts.
These are easily accessed logs written to the directory. They are provided in addition to the historical data provided in the audit logs.

This section discusses the following topics:

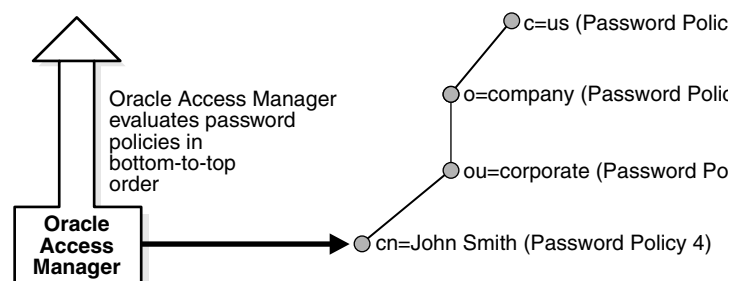
- [Order of Password Policy Evaluation](#)
- [Managing Password Policies](#)
- [Lost Password Management](#)
- [Implementing Password Policies in the Access System](#)
- [Configuring Password Redirect URLs](#)
- [Updates to the Access Server Cache](#)

Order of Password Policy Evaluation

You can configure different password policies for different domains. A domain is an area under a particular node in the directory tree.

A user can qualify under more than one policy in a domain. In this situation, password policies are evaluated in a bottom-to-top order. The first policy that applies to the user is selected, as illustrated in [Figure 7-1](#).

Figure 7-1 Password Policy Evaluation Order



In this example, four password policies apply to John Smith. Password Policy 4 is implemented because it is at the lowest (cn) level of the directory tree, so it is evaluated first.

Managing Password Policies

You configure password policies in the Identity System Console. You can create default password policies that apply to all domains. You can also define password policies for specific directory domains, and you can define multiple policies within a domain.

This section discusses the following topics:

- [Viewing Password Policies](#)
- [Setting the Defaults or Global Settings for Different Types of Password Policies](#)
- [Creating Password Policies for a Specific Domain](#)
- [Modifying Password Policies](#)
- [Deleting a Password Policy](#)

Viewing Password Policies

You view password policies from the Password Policy Management page.

To view a list of password policies

1. From the Identity System Console, click the System Configuration sub-tab.
2. Click Password Policy in the left navigation pane
3. Click the policy's link to view its settings.

The Password Policy Management page appears. This page displays default password policies and a list of domain-specific password policies.

Setting the Defaults or Global Settings for Different Types of Password Policies

You can set defaults for password policies. These defaults include URLs that direct users to a password change page, a password expiry warning page, and a custom account lockout page. These defaults are overridden by any domain-specific policies that you create.

You can create defaults for the following:

- **Password expiry warning URL:** This applies only to resources protected by the Access System.

This URL directs the user to an expiry notice form. Optionally, this URL can also redirect the user to a password change form. Another option for this URL is to return the user to the originally requested resource after the password has been changed.

See "[To set up a default password expiry warning redirect URL](#)" on page 7-67 for details.
- **Password change redirect URL:** This setting applies only to resources protected by the Access System.

It is similar to the password expiry warning URL. This URL points to a password change page. Optionally, this URL can redirect the user to the originally requested resource.

See "[Configuring Redirection to a Password Reset Page After Password Expiry](#)" on page 7-65 for details.
- **Lost password redirect URL:** To be useful as part of your password management system, this URL must exist as a portal insert on a Web page.

In the Identity System Console, you record this URL for informational purposes.

See ["Lost Password Management"](#) on page 7-57 for details.

- **Custom account logout redirect URL:** This URL is applicable only to resources protected by the Access System.

See ["Setting Up Redirect URLs for Account Lockout"](#) on page 7-68 for details.

By default, the Identity Server has a mechanism for displaying logout information. If you want to customize account logout behavior, the Identity System also returns an error code that you can use in an IDXML program. See the *Oracle Access Manager Developer Guide* for details.

- **Successful authentication events:** This writes the time of the user's latest successful login attempts to the user directory server.

By default, the information is written to the `oblastSuccessfulLogin` attribute of the `OblixPersonPasswordPolicy` object class. This feature enables quick access to the most relevant login information. Historical information is provided in the audit logs.

- **Unsuccessful authentication events:** This writes the time of the user's last failed login attempts to the user directory server.

By default, the log authentication information is written to the `oblastFailedLogin` attribute of the `OblixPersonPasswordPolicy` object class. This feature enables quick access to the most relevant login information. Historical information is provided in the audit logs.

To create the default or global password policy

1. From the Identity System Console, click the System Configuration sub-tab.
2. In the left navigation pane, click Password Policy.

The Password Policy Management page appears.

3. Enter the following information:

Lost Password Redirect URL: This is the URL that points users to a lost password management page in the Identity System. When entered in this field, this URL is for informational purposes only. The actual URL is provided in a portal insert. See ["Lost Password Management"](#) on page 7-57 for details.

Password Change Redirect URL: This is a URL to a password change form. See ["Configuring Redirection to a Password Reset Page After Password Expiry"](#) on page 7-65 for details.

Password Expiry Warning Redirect URL: Enter the URL to the password expiry warning form. See ["Setting Up Password Expiry Warning Redirect URLs"](#) on page 7-67 for details.

Custom Account Lockout Redirect URL: Enter the URL to a lockout notification page. See ["Setting Up Redirect URLs for Account Lockout"](#) on page 7-68 for details.

4. Set the logging of authentication attempts as follows:

Successful Attempts Attribute: By default, this value is `oblastSuccessfulLogin`. This is the recommended value. To change this value, enter any string attribute of the `Person` object class, or any string attribute of an auxiliary class that is attached to the `Person` object class.

Failed Attempts Attribute: By default, this value is `oblastFailedLogin`. This is the recommended value. To change this value, enter any string attribute of the Person object class, or any string attribute of an auxiliary class this is attached to the Person object class.

5. Click Enable to enable the logging features.
6. Click Save.

Creating Password Policies for a Specific Domain

You can configure password policies for specific domains. These settings override any global defaults. These policies include password rules, password redirect URLs, and so on.

In a policy for a specific domain, you also can customize the appearance of the the lost password and change password pages that are presented to a user. The default stylesheets for these pages are stored in the following folder:

Identity_install_dir/identity/oblix/lang/language_id/style0

Where *Identity_install_dir* is the directory where the Identity Server is installed and *language_id* is the directory where the language pack that is being used. The default language pack is `en-us`, and the default stylesheets are located in the "classic style" folder named `style0`.

The files `lpm_cr.xsl` and `lpm_changepwd.xsl` are the original stylesheets. You can copy these stylesheets, customize them, and place the files in the currently active style folder. You also modify the relevant password policy to indicate the relative path of the XSL file in the active style folder. After configuring the new password policy, the Identity System applies the custom stylesheet for the custom password policy. The Identity System uses the default stylesheet for all other password policies.

See ["Configuring Styles for Identity System Applications"](#) on page 7-2 for details. Also see the *Oracle Access Manager Customization Guide*.

In the following procedure, the Defaults button at the bottom of the page populates all fields except the Password Policy Name, Password Policy Domain, and Password Policy Filter field.

To create a password policy

1. From the Identity System Console, click the System Configuration sub-tab, click Password Policy in the left navigation pane, then click Add in the Password Policy Management page.
2. In the Password Policy Name field, type the name of your policy.
3. In the Password Policy Domain field, type the domain of the LDAP directory to which this policy applies. For example:

`ou=corporate,o=company,c=us`

4. In the Password Policy Filter field, you can optionally add an LDAP filter to further define the part of the domain to which this password policy applies.

For example, `(title=System Administrator)` would restrict this password policy to a subset of people.

5. In the Lost Password Policy list, select the name of the lost password policy that you want to implement.

See ["Lost Password Management"](#) on page 7-57 for details. If you leave this field blank, a single challenge-response model is used.

6. In the Password Minimum Length field, type the minimum number of characters the password must have.

The default is 8.

7. In the Minimum Number of Uppercase Characters field, type the minimum number of uppercase characters the password must have.

The default is 2.

8. In the Minimum Number of Lowercase Characters field, type the minimum number of lowercase characters the password must have.

The default is 2.

9. In the Minimum Number of Nonalphanumeric Characters field, type the minimum number of nonalphanumeric characters the password must have.

A nonalphanumeric character is any printable character that is not a letter or a number. Examples are +,!, and @.

The default is 1.

10. In the Minimum Number of Numeric Characters field, enter the minimum number of numeric characters the password must have.

11. If external rules apply to this password policy, check Externally specified validation rules.

Oracle Access Manager provides an external hook for password policy implementation. See *Oracle Access Manager Developer Guide* for details.

12. In the Password Validity Period field, select one of the options:

- Password Never Expires.
- Number of Days in which Password expires: Enter the number of days this password is valid. There is no default. You must supply a value if you select this option.

13. In the Password Expiry Notice Period, specify the number of days prior to password expiration that users are notified.

14. In the Mode of Conveying the Expiry Notice field, select one or both options:

- At Login—When users log in, a message informs them of the number of days remaining until their password expires.

If the Identity System is protected by the Access System, you must enter a Password Expiry Warning Redirect URL. See ["Setting Up Password Expiry Warning Redirect URLs"](#) on page 7-67 for details.

- E-mail—Users are notified through email of the number of days remaining before their passwords expire. You cannot customize the message.

15. In the Password Minimum Age field, enter number of days the password must last before users can change it.

16. Select Change on Reset if you want to force users to change the password the first time they log in to the system after an administrator resets the password.

By default, the Change on Reset flag is not set. During self-registration, the Change on Reset flag is not set.

This field is applicable to both the Identity and Access Systems. For the Access System only, you can also configure a redirect URL for password change. See ["Configuring Password Redirect URLs"](#) on page 7-65 for details.

17. In the Password History field, indicate whether or not you want to maintain a password history.

Select Do not Keep Password History or enter the number of passwords to be saved for each user. Saved passwords are stored in the directory and cannot be re-used. The default is 5.

Oracle Access Manager can determine the latest passwords saved in the directory. If you delete one, Oracle Access Manager determines which remaining one is the oldest.

18. In the Number of Login Tries Allowed field, specify the number of login attempts allowed before the user's account is locked.

The default value is 3. This means that if a user attempts to login three times using an incorrect login credential, they will be locked out after the third attempt that occurs within the lockout interval specified by "Lockout Duration value". An incorrect login credential consists of a correct username but incorrect password. During the lockout interval, the user cannot login even with correct credentials.

Note: This also applies to the number of attempts for a challenge response during Lost-Password Management.

19. In the Lockout Duration field, specify the number of hours an account remains locked after a user exceeds the number of failed logins specified in the previous step.

The default is 24 hours. To clear a lockout before the lockout duration expires, an administrator can reset the user's password from the Identity System. Upon login the user is redirected to a page where he or she can choose a new password—if Change on Reset was selected in the Password Policy Management page before the administrator reset the password.

If Change on Reset was not selected when the administrator assigned a new password, the user can log in to the system with the administrator-assigned password.

20. In the Login Tries Reset field, specify the number of days to store the failed login attempts that are uninterrupted by a successful login.

For example, if this value is set to 3, and a user fails to log in once, the application keeps track of that failure for 3 days before clearing it.

21. In the Lost Password Redirect stylesheet, you can optionally enter a relative path to an XSL stylesheet.

For example, if the stylesheet file is located in *Identity_Server_installdir* /identity/oblix/lang/en-us/style0/myStyleSheet.xsl, you would enter /myStyleSheet.xsl.

See the *Oracle Access Manager Customization Guide* for details on stylesheet configuration. See ["Lost Password Management"](#) on page 7-57 for details on lost password redirection.

If you leave this field blank, the default stylesheet is used.

22. In the Password Change Redirect stylesheet you can optionally enter a relative path to an XSL stylesheet.

For example, if the stylesheet file is located in *Identity_Server_installdir* /identity/oblix/lang/en-us/style0/myStyleSheet.xsl, you would enter /myStyleSheet.xsl.

See the *Oracle Access Manager Customization Guide* for details on stylesheet configuration. See ["Configuring Redirection to a Password Reset Page After Password Expiry"](#) on page 7-65 for details on the change password form.

If you leave this field blank, the default style sheet is used.

23. In the Password Expiry Warning Redirect URL, you can optionally specify a URL to override the default.

This applies only to the Access System. See ["Setting Up Password Expiry Warning Redirect URLs"](#) on page 7-67 for details on this URL.

24. In the Custom Account Lockout Redirect URL, specify the redirect URL for users who have exceeded the number of login attempts.

This applies only to the Access System. See ["Setting Up Redirect URLs for Account Lockout"](#) on page 7-68 for details.

25. Select Password Policy Enable to enable this password policy.

If you later change the setting of this field or make any other change to this password policy, you have to flush the password policy cache. You can flush the password policy cache in the Access System Console. From the Access System Console, click Common Information Configuration, and click the Flush Password Policy Cache tab. For more information, see *Oracle Access Manager Access System Administration Guide*.

26. Click Save to save this policy and return to the Password Policy Management page.

The new policy appears in the list on the page.

Note: The Redirect URLs shown on this page apply to the Access System. For more information, see ["Implementing Password Policies in the Access System"](#) on page 7-64.

Modifying Password Policies

During this operation, you can click the Defaults button to populate all fields with default values, except the Password Policy Name, Password Policy Domain, and Password Policy Filter. See ["Order of Password Policy Evaluation"](#) on page 7-50 for information about each parameter.

To modify a password policy's parameters

- From the Identity System Console, click the System Configuration sub-tab, then click Password Policy in the left navigation pane.

The Password Policy Management page displays a list of password policies.

- In the Password Policy Management page, click the policy you want to modify.

The page with the policy's parameters appears.

- Modify the parameters as necessary.

- Click Save.

Deleting a Password Policy

The Password Policy Management page displays a list of password policies. Saved passwords are stored in your LDAP directory. Oracle Access Manager can determine the latest passwords. If you choose to delete one, Oracle Access Manager determines which is the oldest.

To delete a password policy

1. From the Identity System Console, click the System Configuration sub-tab, then click Password Policy in the left navigation pane.
2. In the Password Policy Management page, select the check box next to the policy you want to delete.
3. Click Delete.
4. Click OK when prompted to confirm your deletion.

Lost Password Management

Lost password management enables users to reset their passwords if they forget them. Lost password management is a process that consists of the following:

- An "I lost my password" link that directs users to a page where they can respond to challenge phrases.

You implement lost password management by creating a URL to a lost password management page, and placing this link on Web pages where you want users to be able to access this functionality. These URLs are known as portal inserts. See the *Oracle Access Manager Customization Guide* for details.

- A lost password management page that contains challenge phrases and response fields.

The lost password management URL routes the users to an Identity System page that contains one or more challenge questions.

- A password reset page that is displayed if the user answers the challenges correctly.

After giving the correct response, users can set a new password in the Identity System.

- Additional functionality to allow the user to be redirected back to the originally requested page.

For record-keeping purposes, you record the lost password management URL in a password policy in the Identity System Console. You can configure a lost password management policy for a particular domain.

The Identity System encrypts response values using an encryption scheme licensed from RSA. This encryption scheme is different from secure hash algorithm (SHA). You can replace the default encryption with your own by writing a custom action using the Identity Event Plug-in API. This is useful if you have existing challenge and response attributes that you want to import into the Identity System. See the *Oracle Access Manager Developer Guide* for details.

Lost password management is enabled by default.

Note: Unlike other redirect URLs that you enter in the Identity System, you enter the lost password management redirect URL for informational purposes only. Its primary location is in the portal insert.

Task overview: Implementing Lost Password Management

1. In your directory, create two new attributes: one to be used for challenges that are presented to users, and one for responses that users provide to the challenges.

To support lost password management, you define an attribute pair to store values for challenges that are presented to users and for responses to those values. For example, you can define a pair of attributes named Challenge and Response. From the Identity System Console, you assign the Challenge and Response semantic types to these attributes. See ["To configure challenge and response-type attributes in your directory"](#) on page 7-60 for details.

2. From the Identity System Console, modify the attributes in your Person object class.

Configure the attribute that is to store the challenge phrases, and the attribute that is to store the user's responses to the challenge phrases.

See ["To configure the Lost Password Management attributes"](#) on page 7-61 for details.

3. From User Manager, configure attribute access controls for these attributes.

To be able to view and view and modify challenges and responses on profile pages, users need read and modify rights for the challenge and response. See ["Allowing Users to View and Change LDAP Data"](#) on page 4-22 for details.

4. Add the challenge and response attributes to user profile pages so that they appear when users view or modify these pages.

During lost password management, users are directed to a page that presents them with challenge phrases. The page that contains the challenge phrases and response input fields is a profile page that you configure in the User Manager. See ["Configuring Tab Profile Pages and Panels"](#) on page 4-12 for details. Note that users can configure challenges during login, even if the associated attributes have no "self" read or modify rights.

5. Configure workflow step pages to use these attributes.

The challenge phrases and responses need to be configured during self-registration so that they can be used later for password retrieval. It is less likely, but possible, that a Create User workflow could also be useful to enable participants in the workflow to configure challenge phrases for new users. See ["Chaining Identity Functions Into Workflows"](#) on page 5-1 for details. The workflow step page does not require read or write permissions for challenge parameters.

6. Configure lost password management policies from the Identity System Console.

See the ["Managing Password Policies"](#) on page 7-51 for details.

7. Insert a lost password management URL in a third-party application or portal page.

See the information in this section for details. Also see the information on portal inserts in the *Oracle Access Manager Customization Guide*.

Syntax for the Lost Password Management URL

The format of a lost password management URL is as follows:

```
http://machinename:portnumber/identity/oblix/apps/lost_pwd_mgmt/bin/lost_pwd_mgmt.cgi?program=passwordChallengeResponse&login=%scheme1_uid_parameter_value%%scheme2_uid_parameter_value%%schemeN_uid_parameter_value%%&target=top
```

This is similar to a password expiry reset URL, as described in ["Configuring Redirection to a Password Reset Page After Password Expiry"](#) on page 7-65. One difference between the two types of URL is the following parameter:

```
program=passwordChallengeResponse
```

Another difference is that if you supply variables such as the user ID on this URL, you will need to modify the corresponding cgi script to pass in the user's login ID.

Alternatively, you can use the following URL syntax and require the user to re-enter a user ID after being redirected to the lost password reset page:

```
http://machinename:portnumber/identity/oblix/apps/lost_pwd_mgmt/bin/lost_pwd_mgmt.cgi?program=passwordChallengeResponse&target=top
```

In the preceding URL, `lost_pwd_mgmt.cgi` is provided with Oracle Access Manager.

About Presenting Challenge Phrases to Users

Lost password management can be configured as either a single or a multiple challenge-response system. Challenge-response pairs are displayed in the following locations:

- During a create user account workflow, a workflow step will contain entries for challenge and response attributes.
- On View and Modify Profile pages, challenge and response pairs are displayed.
- During lost password management, users must respond to one or more challenge phrases.
- When logging in to Oracle Access Manager, users are prompted to provide additional challenges if the configured number of challenges is lower than the number required in the lost password policy that applies to the user.

About Other Aspects of the Challenge and Response Page

In addition to providing challenge and response prompts on the lost password management page, you may want to provide additional information on the page. For example, you may want to configure a link (or a workflow step) that sends the user to the password reset page after the user successfully responds to the lost password management challenges.

How the User Experiences Lost Password Management with Multiple Challenges

If a lost password management policy applies to the user, when the user clicks the lost password management URL, multiple challenges and response fields are displayed. The number of challenges that the user sees is determined by the Minimum Challenges to be Answered field. See ["To configure lost password management for a password policy domain"](#) on page 7-62 for details.

As noted in ["Creating Password Policies for a Specific Domain"](#) on page 7-53, password policies apply to particular domains or groups of users. If no lost password policy applies to a user, only one challenge phrase and one response entry field are displayed. If multiple challenge and responses are configured, but no lost password

policy applies to a user, the first configured challenge phrase and response are displayed.

Users are prompted to configure additional challenge phrases during login if the number of configured challenge phrases and responses falls short of the minimum configured in the lost password policy. For example, suppose that you increase the minimum number of challenges to be configured after establishing the initial lost password management policy. In this case, during login the user is presented with an additional challenge phrase, displayed as one of the following:

- A text box: This appears if the User setting was selected in the lost password management policy.
- A select box with predefined phrases: This appears if the Predefined setting was selected in the lost password management policy.
- A combo box with predefined phrases: This appears if the User or Predefined setting was selected in the lost password management policy.

See ["To configure lost password management for a password policy domain"](#) on page 7-62 for details.

If you change the source type in the lost password management policy, for example, if you change the source from User to Predefined, or you change the minimum length, or change the Allow Duplicate Responses flag, these changes are enforced when the user modifies his or her own profile.

When being prompted to configure additional challenges, the type of message that the user receives depends on what has changed in the lost password management policy. For example, suppose that you change the minimum length of a response from 3 characters to 8 characters. When the user tries to save a change to his or her profile, an error message, "response does not meet the minimum length requirement" appears. If you increase the minimum number of challenges in the policy, the user is prompted to supply the required information during login.

Viewing and Configuring Lost Password Management Policies

The following procedures describe how to configure lost password management.

Note: You cannot delete challenge parameters from profile or workflow pages if there is a lost password management policy in effect.

To configure challenge and response-type attributes in your directory

1. Ensure that there are two unused, empty attributes in your directory to be used for user challenges and responses.

If two appropriate attributes are available, continue to the following procedure.

2. If you need to add new attributes, you can do so using whatever method you prefer.

The following is an example of creating an LDIF schema file with a new auxiliary object class and two new attributes. You could create attributes that are similar to the following using the syntax appropriate for your directory server type:

```
# ----- Attributes -----  
#  
dn: cn=schema  
changetype: modify
```

```

add: attributetypes
attributetypes: ( 1.3.6.1.4.1.9999.1.1094.204 NAME 'Challenge2' SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 )

dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( 1.3.6.1.4.1.9999.1.1094.205 NAME 'Response2' SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 )

# ----- Object class -----
#
dn: cn=schema
changetype: modify
add: objectclasses
objectclasses: ( 1.3.6.1.4.1.9999.1.1094.206 NAME 'oblixAuxPerson4LPM' DESC
'User defined objectclass' SUP top AUXILIARY MAY ( Challenge2 $ Response2 ) )

```

3. Import the LDIF file into the directory.
4. To configure the new auxiliary object class in Oracle Access Manager, in the Identity System Console, click Common Configuration.
5. Click Object Classes in the left navigation pane.
6. Click Add.
7. Select the name of the new object class from the list.
8. Select the option button for the Person structural object class.

The new auxiliary object class is now associated with the Person structural object class. You can now configure the new challenge and response-type attributes in this object class, as described in ["To configure the Lost Password Management attributes"](#) on page 7-61.

To configure the Lost Password Management attributes

1. To configure an attribute from your Person object class to store the challenge phrases, from the Identity System Console, click Common Configuration.
2. Click Object Classes in the left navigation pane.
3. Click the link for the Person object class.
4. Click Modify Attributes.
5. In the Attribute list, select the attribute that you want to use for the challenge phrase.

This must be a new, empty attribute.

6. Configure the following.

See ["About Object Class Attributes"](#) on page 3-10 and ["Configuring Attributes"](#) on page 3-17 for details:

- Set the Semantic Type of this attribute to Challenge.
- Set the Data Type to case-insensitive string.
- Set the Attribute Value(s) field to Single
- The Display Type is configured automatically, depending on the type of policy you configure.

- Assign an appropriate Display Name, for example, Challenge.

If you configure multiple challenge phrases, these are stored as a single value in the user's directory entry for the challenge attribute. The values are stored in encoded format.

7. From the Attribute list, select a second attribute from your Person object class to store the user's responses to the challenge phrases.

This must be a new, empty attribute.

8. Configure this attribute as follows.:

- Set the Semantic Type of this attribute to Response.
- Set the Data Type to case-insensitive string.
- Set the Display Type to Password.
- Set the Attribute Value(s) field to Single.
- Assign an appropriate Display Name, for example, Response.

See ["About Object Class Attributes"](#) on page 3-10 and ["Configuring Attributes"](#) on page 3-17 for details:

If you configure multiple challenges and responses as part of your lost password management policy, the values that the user provides for the responses are stored as a single value in the user's directory entry for the response attribute. The values are stored in encoded and encrypted format.

9. Add these attributes to user profile pages or workflow step panels, depending on where you want them to be displayed.

To view lost password policies

1. In the Identity System Console, click the System Configuration sub-tab, then click Lost Password Policy in the left navigation pane.
2. In the Lost Password Policy Management page, click the link for the policy that you want to view.

To enable or disable Lost Password Management

1. Locate the oblixbaseparams.xml file.

The default path for the file is as follows:

```
IdentityInstall_dir/identity/oblix/apps/common/bin/oblixbaseparams.xml
```

where *IdentityInstall_dir* is the directory where the Identity System is installed.

2. Make sure Yes is entered for the Apply_LostPwdMgmt parameter.
Yes is the default. Otherwise type No to disable this feature.
3. Save and close the file.

To configure lost password management for a password policy domain

1. From the Identity System Console, click the System Configuration sub-tab, then click Lost Password Policy.
2. In the Lost Password Policy page, click Add.
The Add Lost Password Policy Page appears.
3. In the Lost Password Policy Name field, enter a name.

Note that for environments that use Oracle Internet Directory, this name cannot contain any special characters.

4. For Challenge Phrase Source, select who is to provide the challenge phrase:
 - **User:** When creating an account, the user must supply the challenge phrases.
 - **Predefined:** When creating a user account, a list of predefined challenges are shown to the user. The user must select from among the supplied challenge phrases.
 - **User or Predefined:** When creating a user account, a list of predefined phrases are shown to the user. The user can either select from among the supplied challenge phrases, or supply new challenge phrases.
5. In the Predefined Challenge Phrases field, enter a challenge phrase and click Add. The phrase is added to a selection list. Use the Delete button to remove selected phrases from the list.

After completing the definition of this lost password policy, you can add predefined challenge phrases or delete existing ones at any time.
6. In the Minimum Challenges to be Configured field, enter the number of challenges that are to be configured when creating the user account or modifying the user profile.
7. In the Challenge Response Minimum Length field, enter the minimum number of characters permitted in the responses that a user configures.
8. The Allow Duplicate Responses checkbox configures the following:
 - **Unchecked: False.** If the user provides the same response for more than one challenge phrase, an error is displayed.
 - **Checked: True.** Turns off duplicate checking.
9. In the Minimum Challenges to be Answered field, enter the number of challenges that you want the user to respond to when resetting the password using the lost password management application.

This value must be the same or less than the one in the Minimum Challenges to be Configured field. For example, if you set the value of this field to 3 and you configure four challenge-response pairs, the user must respond to three challenges. Note that the actual number of responses that the user must provide depends on the number of correctly configured challenges and responses as well as the value of this field. For example, if you enter 2 in this field, but only one of the two challenge-response pairs is configured correctly, the user is prompted to respond to only one challenge.
10. In the Challenge Pose Type field, select whether challenges are to be displayed all at once or sequentially.
 - **All at once:** Challenges are displayed at one time. The order in which they are displayed changes each time. The user must respond correctly to all challenges.
 - **One after the other:** The user must respond to one challenge phrase before the next is displayed.
11. Select the Send Email After Password Change box if you want email to be sent to the user after the password has been reset.

By sending email to the user, if an intruder has reset the password, the user can notice that something unexpected has happened and can contact the administrator.

12. Select the Lost Password Policy Enable box if you want administrators to be able to use this policy.
13. Click Save.
14. Add the name of this policy to a password policy domain.

See ["Creating Password Policies for a Specific Domain"](#) on page 7-53 for details.

Implementing Password Policies in the Access System

You can apply the password policies configured in the Identity System Console to resources that the Access System protects. To do this, you modify the authentication scheme that protects those resources. When users authenticate to a resource protected by the Access System, the password policy is invoked for the users if they are in the password policy domain.

The rest of this section discusses the following topics:

- [Modifying Authentication Schemes to Include a Password Policy](#)
- [Configuring Redirection to a Password Reset Page After Password Expiry](#)
- [Setting Up Password Expiry Warning Redirect URLs](#)
- [Setting Up Redirect URLs for Account Lockout](#)
- [Updates to the Access Server Cache](#)

See the section ["Order of Password Policy Evaluation"](#) on page 7-50 for more information. See the *Oracle Access Manager Access System Administration Guide* for instructions on creating an authentication scheme.

Modifying Authentication Schemes to Include a Password Policy

The following procedure describes how to modify an authentication scheme to include a password policy.

To modify an authentication scheme to include a password policy

1. Log in to the Access System.
2. From the Access System Console, click Access System Configuration and then click Authentication Management in the left navigation pane.

The Authentication Management page appears, listing all the configured authentication schemes.
3. Click the link for an authentication scheme you want to change, and then click the Modify button on the page that appears.

The Modify Authentication Scheme appears.
4. Choose the validate_password plug-in, add the following information to the Plugin Parameters field, and click Save:

```
obReadPasswdMode="LDAP", obWritePasswdMode="LDAP"
```

For example, suppose the original validate_password Plugin Parameters statement for the Basic Over LDAP scheme was the following:

```
obCredentialPassword="password"
```

The new parameter set would be as follows:

```
obCredentialPassword="password",obReadPasswdMode="LDAP",
obWritePasswdMode="LDAP".
```

The new parameters must be added for password change redirection.

5. If the WebPass and WebGate reside on different servers, choose the validate_password plug-in and add the obWebPassURLprefix parameter in the Plugin Parameters field, as follows:

```
http://webpasshost:webpassport
```

Where *webpasshost* and *webpassport* are the host and port for the server where the WebPass resides.

6. Make a note of the uid parameter value for the credential_mapping plug-in. You need this value when creating the password change redirect URL. For example, the uid parameter value may be %userid%.
7. Repeat this process for all authentication schemes for which you want to set up password change redirection.

Note: If you make any change to the password policy, be sure to flush the Access Server cache. See ["Updates to the Access Server Cache"](#) on page 7-68 for more information.

Configuring Password Redirect URLs

You can configure URLs that redirect users to the following pages:

- To a password reset page
- To a password expiration warning page
- To an error page stating that the user account is locked.

These redirect URLs apply only to cases where users log in to resources that are protected by a WebGate or AccessGate. In other words, if you have protected a resource as described in *Oracle Access Manager Access System Administration Guide*, you can configure one of these URLs. You can also configure these URLs to return the user to the originally requested resource when they are done with the target page specified in the redirection URL.

These URLs are explained in the following sections:

- [Configuring Redirection to a Password Reset Page After Password Expiry](#)
- [Setting Up Password Expiry Warning Redirect URLs](#)
- [Setting Up Redirect URLs for Account Lockout](#)

Configuring Redirection to a Password Reset Page After Password Expiry

When a password has gone beyond the validity period that you configured in a password policy, when the user attempts to log in, he or she is automatically redirected to a password reset page. You can configure the URL to this password reset page. Optionally, the password reset URL can redirect the user back to the originally requested resource after the password is reset.

To enter a password change redirect URL

1. From the Identity System Console, click the System Configuration sub-tab, then click Password Policy in the left navigation pane.

The Password Policy Management page displays a list of password policies.

2. In the Password Change Redirect URL field, enter a URL with the following syntax:

```
http://machinename:portnumber/identity/oblix/apps/lost_pwd_mgmt/bin/lost_pwd_
mgmt.cgi?program=redirectforchangepwd&login=%scheme1_uid_parameter_value%
%scheme2_uid_parameter_value%%schemeN_uid_parameter_value% &target=top
```

Where:

- *machinename:portnumber* are the host and port of the Web server on which a WebPass is installed, and
- *%scheme1_uid_parameter_value% %scheme2_uid_parameter_value%%schemeN_uid_parameter_value%* is the string of uid parameter values for all the authentication schemes for which you want to set up password change redirection

For example, suppose that you have the following credential_mapping plug-in parameters for two authentication schemes:

- **Form over LDAP**—obMappingBase="o=company,c=us",
obMappingFilter="((&(objectclass=genSiteOrgPerson)
(uid=%login%)))"
- **Basic over LDAP**—obMappingBase="o=company,c=us",
obMappingFilter="((&(objectclass=genSiteOrgPerson)
(uid=%userid%)))"

The password change redirect URL that corresponds to these parameters would be the following:

```
http://machinename:portnumber/identity/oblix/apps/lost_pwd_mgmt/bin/lost_pwd_
mgmt.cgi?program=redirectforchangepwd&login=%login%%userid%&target=top
```

3. To return the user to the originally requested resource after submitting the password change form, you can code a BackURL statement in the query string for this URL. The basic syntax is:

```
http://machinename:portnumber/identity/oblix/apps/lost_pwd_mgmt/bin/lost_pwd_
mgmt.cgi?prtforchangepwd&login=%login%%userid%&backURL=%HostTarget%%RESOURCE%
&target=top
```

For example:

```
http://130.35.46.141:99/identity/oblix/apps/lost_pwd_mgmt/bin/lost_pwd_
mgmt.cgi?login=%login%%userid%&backUrl=%HostTarget%%RESOURCE%
```

At runtime, the URL of the originally requested resource is substituted for the values enclosed in the percent delimiters, for example:

```
http://130.35.46.141:99/identity/oblix/apps/lost_pwd_mgmt/bin/lost_pwd_
mgmt.cgi?login=admin&backUrl=http://www.webserver1.com/test/a.html
```

The script `lost_pwd_mgmt.cgi` contains logic to process the query parameters. The `lost_pwd_mgmt.cgi` script is provided with Oracle Access Manager. It is a dynamically generated page, so no manual configuration for it is necessary.

4. Click Save.

Setting Up Password Expiry Warning Redirect URLs

You configure a password expiry period in a password policy. When a user's password is about to expire, a redirect URL sends the user to a warning page. Oracle Access Manager does not provide this warning page. You must create the actual landing page that contains the warning. Optionally, this warning page can also contain a URL that directs the user to the password reset page.

The password expiry redirect URL applies only to resources that are protected by the Access System. In other words, if you have protected a resource with a WebGate, you can configure this URL.

The password expiry URL is similar to the password change redirect URL. The URL directs the user to an expiry notice form, optionally redirects the user to a password change form, then optionally returns the user to the originally requested resource after the password is changed.

You can configure a default password expiry URL that applies to all password policies, and you can configure this URL for an individual policy.

Users may be redirected automatically to this URL, or they can be notified by email prior to expiry. See ["Creating Password Policies for a Specific Domain"](#) on page 7-53 for details.

There is no built-in page or portal to serve as a target for this URL. You must create this page. For example, you may want to provide a page that states, "Your password will expire soon and needs to be changed."

To set up a default password expiry warning redirect URL

1. From the Identity System Console, click the System Configuration sub-tab, then click Password Policy in the left navigation pane.

The Password Policy Management page displays a list of password policies.

2. In the Password Expiry Warning Redirect URL, enter a URL with the following syntax:

```
http://machinename:portnumber/path-to-custom-page
```

Where:

- *machinename:portnumber* are the host and port of the Web server on which a WebPass is installed, and
 - *path-to-custom-page* is the path of the custom Web page that warns them that their password is about to expire.
3. To return the user to the originally requested resource after authenticating, you can code a "back URL" (that is, a backURL statement) in the query string for this URL as follows:

```
http://machinename:portnumber/notice.cgi?prtforchangepwd&login=%login%%userid%&backURL=%HostTarget%%RESOURCE%&target=top
```

For example, you could enter the following URL:

```
http://130.35.46.141:99/cgi-bin/notice.cgi?login=%login%%userid%&backUrl=%HostTarget%%RESOURCE%
```

In this example, notice.cgi contains logic to process the query parameters. You can create a simple Web page, or write a cgi or another script or JSP page to parse the parameters in the URL and display appropriate messages, process timeouts, and redirect the user to the backURL.

At runtime, the actual values of the user and the originally requested resource are substituted for the query strings. For example:

```
http://130.35.46.141:99/cgi-bin/notice.cgi?login=admin&backUrl=http://  
www.webserver1.com/test/a.html
```

In this example, notice.cgi is a script that you have written that contains logic to process the query parameters.

The custom page can also retrieve the expiration date using an ExpiryDate query parameter. The following is an example of this parameter:

```
http://130.35.46.141:99/cgi-bin/notice.cgi?ExpiryDate=%PwdExpiryDate%&backUrl=%  
HostTarget%%RESOURCE%
```

4. Click Save.

Setting Up Redirect URLs for Account Lockout

As with the other redirect URLs, the redirect URL for account lockout is applicable only to the Access Server. You can configure a lockout URL that has no user ID or requested resource information.

To implement account lockout redirection, you need to create a Web page, or write a cgi or another script or JSP page to parse the parameters in the account lockout URL. The script or JSP should display messages regarding account lockout, process timeouts, and redirect the user to the originally requested resource.

To set up the account lockout URL

1. From the Identity System Console, click the System Configuration sub-tab, then click Password Policy in the left navigation pane.

The Password Policy Management page displays a list of password policies.

2. In the Custom Account Lockout Redirect URL field, enter a URL for redirecting the user to an account lockout form.
3. Click Save.

Updates to the Access Server Cache

You can ensure that the Access Server is notified of changes made by the Identity System and that the Access System's cache is flushed automatically. However, if you choose to not implement automatic cache flush, you can still manually flush the cache when you make changes to the Password Policy Management page in the Identity System. This can be useful in avoiding a significant delay in applying password-policy management changes.

For more information about flushing the Access Server caches, see *Oracle Access Manager Access System Administration Guide* and the *Oracle Access Manager Deployment Guide*.

Configuring the Access Manager SDK for the Identity System

The Access Manager SDK consists of libraries, build instructions, and examples that you use to build an AccessGate for non-Web resources. The Access Manager SDK is automatically installed with the Identity System in *IdentityServer_install_dir* / AccessServerSDK.

The following functions in the Identity System require the Access Manager SDK. You must manually configure the Access Manager SDK for these functions:

- Automatic cache flush between the Identity System and Access System
- Automatic login to the Access System after self-registration.

Complete the following procedure if you protect WebPass with a WebGate. You do not have to repeat the procedure for each Identity System function previously mentioned.

To configure the Access Manager SDK

1. Install and set up the Identity System and Access System, as described in the *Oracle Access Manager Installation Guide*.

Note: The Access Manager SDK is installed automatically with the Identity System in *IdentityServer_install_dir* / AccessServerSDK.

2. **Windows:** Set your path to point to the Access Manager SDK by modifying the systems PATH variable as shown below:

```
set PATH = %PATH%;Identityserver_install_dir\AccessServerSDK\oblix\lib
```

3. From the Access System Console, click Access System Configuration, AccessGate Configuration.
4. Add an AccessGate.

You do not need to configure a port.

The Identity System uses the AccessGate to communicate with the Access Server for purposes of flushing the cache.

For more information about flushing the Access Server caches, see *Oracle Access Manager Access System Administration Guide* and the *Oracle Access Manager Deployment Guide*.

5. Select Off for Access Management Service if you have upgraded your WebGates. A value of On is only appropriate for legacy systems.
6. Save the AccessGate.
7. Access the *IdentityServer_install_dir* / identity / AccessServerSDK / oblix / tools / configureAccessGate directory and run the configureAccessGate script.

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

See the *Oracle Access Manager Access System Administration Guide* for details about modifying an AccessGate.

When running configureAccessGate, ensure that the AccessGate ID is the same as the AccessGate name you entered from the Access System Console in step 3.

8. From the *IdentityServer_install_dir*/identity/oblix/data/common directory, open the basedbparams.xml parameter catalog file in a text editor.
9. Change the value of the doAccessServerFlush flag to true as follows:

```
<NameValPair ParamName="doAccessServerFlush" Value="true" />
```
10. Restart the Identity Server

Cloned and Synchronized Components

Instead of using the command line or the installation GUI to install a Oracle Access Manager component, you can automatically install a component by *cloning* the configuration of an already-installed component. Cloning creates a copy of a component on a remote system using an already-installed component as a template.

Synchronizing enables you to harmonize two installations of the same 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 details.

Part III

Performing Common Administrative Tasks

Certain functions and tasks are common to both the Oracle Access Manager Identity System and the Access System.

Part III explains how to perform tasks that are common to all Oracle Access Manager applications:

- [Chapter 8, "Changing Transport Security Modes"](#)
- [Chapter 9, "Reporting"](#)
- [Chapter 10, "Logging"](#)
- [Chapter 11, "Auditing"](#)
- [Chapter 12, "SNMP Monitoring"](#)

Changing Transport Security Modes

Setting up transport security is the subject of this chapter and is one of the administrative tasks that is common to both the Identity System and the Access System.

This chapter contains the following topics:

- [About Transport Security Modes](#)
- [Changing Transport Security for the Identity System](#)
- [Changing Transport Security Modes for the Access System](#)
- [Transport Security Changes for Directory Servers](#)
- [Changing Transport Security Passwords](#)
- [Importing Multiple CA Certificates](#)
- [Changing Access Server Security Password](#)

About Transport Security Modes

A transport security mode is a method to protect communication between two points, such as a client and a server. To ensure protection, communication can be encrypted with a certificate authority (CA).

Oracle Access Manager offers the following three transport security modes for communication between components, as discussed in greater detail in the *Oracle Access Manager Installation Guide*:

- **Open:** Communication is not encrypted for protection. Use this mode when security is not an issue; for example, when testing communications between an AccessGate and the Access Server, as long as you consider your network secure. Open is the default setting.
- **Simple:** Communication is encrypted with Oracle Access Manager's internal CA. Simple mode encrypts communications using Transport Layer Security, RFC 2246 (TLS v1). This mode is less secure than Cert mode. Use this mode if you have some security concerns but do not want to manage your own CA.
- **Cert:** Communication is encrypted with an external CA. With Cert mode, communications are encrypted using TLS v1. In addition, each element, both client and server, must present an X.509 certificate (in base64 format) when establishing a connection. The certificate must be provided by you, perhaps from a third-party CA.

Note: As of version 7.0, the default certificate store format and name has changed from cert7.db to cert8.db. When you upgrade from a version earlier than version 7.0, you continue to use the old certificate store (cert7.db).

When you run the `configureAAAServer`, `setup_ois`, or `setup_accessmanager` utilities, the certificate store format and name is automatically modified to cert8.db. version 7.0 and higher versions work with both the cert7.db (upgraded environments) and cert8.db (new installations) certificate store. On non-Windows systems, you use the following tools: `start_configureAAAServer`, `start_setup_ois`, `start_setup_accessmanager`.

The following two transport security modes are used for communication between a Oracle Access Manager component and the directory server:

- **Open:** Directory server communication is not encrypted for protection. Use this mode when security is not an issue; for example, when testing communications between an AccessGate and the Access Server, as long as you consider your network secure. Open is the default setting.
- **SSL:** Directory server communication using SSL.

Specifying transport security is part of the installation process. See the differences when installing the Identity System or Access System, in [Table 8–1](#).

Table 8–1 Specifying a Security Mode During Installation

Identity System	Access System
■ Install the Identity Server component. Specify the transport security mode used to communicate with WebPass. ■ Install the WebPass component. Specify the transport security mode used to communicate with the Identity Server.	■ Install Policy Manager. Specify the transport security mode used to communicate with the Access Server. ■ Create an Access Server instance in the Access System Console. Specify the transport security mode used to communicate with the Policy Manager. ■ Define a WebGate instance in the Access System Console. Specify the transport security mode used to communicate with the Access Server. ■ Install the Access Server component. Configure the transport security mode to communicate with WebGate. ■ Install the WebGate component. Configure the transport security mode to communicate with Access Server.

See also: See the *Oracle Access Manager Installation Guide* for more information on installing components.

Transport Security Mode Between Components

Transport security can be configured between the following components:

- **Identity System:** Transport security between all Identity Servers and WebPass instances must match: either all open, all Simple mode, or all Cert.
- **Access System:** Transport security among all Policy Managers, Access Servers, and associated WebGates must match: either all open, all Simple mode, or all Cert.

Access Cache Flushing Caveat: When access cache flushing is enabled on the Identity Server, the Identity Server communicates with the Access Server. In this case, the

transport security mode among all five of the following components must be in the same mode.

- Identity Servers and WebPass instances
- Policy Managers, Access Servers, and associated WebGates

For details about managing caches, see both [Managing Caches](#) on page 7-13 of this manual and *Oracle Access Manager Access System Administration Guide*. For more information on caching, see the *Oracle Access Manager Deployment Guide*.

If you need to change the transport security mode after installation, you can change the security mode in the System Console:

Identity System (WebPass and Identity Server): You select a transport security mode for WebPass and Identity Server instances in the Identity System Console. Decide on the type of transport security mode you want to use before you configure WebPass and Identity Server instances. Again, transport security among all components must match. They must all be open, simple, or cert.

Access System (Policy Manager, AccessGate, and Access Server): You select a transport security mode for the Access System when configuring AccessGate and Access Server instances in the Access System Console. Decide on the type of transport security mode you want to use before you configure the AccessGate and Access Server instances. Again, transport security among all Access System components must match: either all open, all simple mode, or all cert.

After changing the mode in the System Console, follow the process described in:

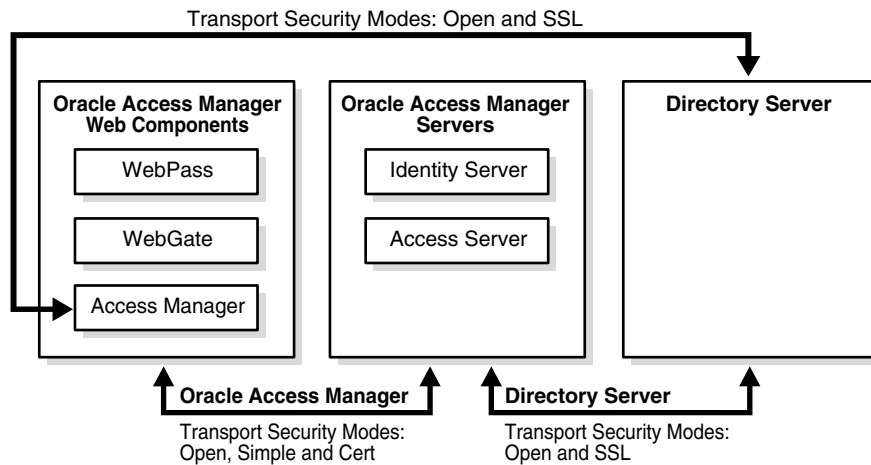
- ["Transport Security Mode Changes for the Access System"](#) on page 8-10
- ["Transport Security Changes for Directory Servers"](#) on page 8-19

You may change the security mode between a component and the Directory server after installation:

Identity or Access Server and the Directory Server: Transport security between the directory server and an Identity or Access Server can be in Open or SSL mode. You specify this transport security mode during installation. If you select SSL, you also specify the location of the SSL certificate. The directory server is automatically updated with the specified security mode information.

When configuring SSL for the directory server, note that Oracle Access Manager supports server authentication only. Client authentication is not supported. Oracle Access Manager verifies the server certificate against the Root CA certificate that you imported during product setup.

The Policy Manager is a Web component that reads from and writes to the directory server. You also specify transport security between the Policy Manager and directory server. [Figure 8-1](#) illustrates the supported transport security modes between Oracle Access Manager Web components and servers, and Oracle Access Manager components and the directory server.

Figure 8–1 Transport Security Modes

You can share directory profiles for all components running in SSL mode, even if these components were initially configured in different modes. For example, suppose the Identity Server and Access Server were installed in open mode with the directory, and the Policy Manager was installed with SSL enabled for the directory server. In this case, the `cert8.db` and `key3.db` files must exist for each component that communicates with the directory server and must reside in the `component_install_dir\identity\access\oblix\config` directory. If these files do not exist, copy them from other existing component folders or run the `genCert` (Policy Manager) or other utilities to generate them, as described in this chapter.

About CA Certificates

This discussion explains the root certificate, request, and other certificate files.

If you select the `cert` transport security mode between components during installation, you need to create and install a root certificate. The root certificate file is a chain of certificates that is generated when you submit a certificate signing request, such as a CSR to a certificate authority. This request is in the form of an `xxx_req.pem` file. You store a root certificate as a file called `xxx_chain.pem`. You download the `xxx_chain.pem` file from the Certificate Server and store it in the following directory with the key and `cert.pem` files, then specify its location during product configuration:

`Component_install_dir\identity\access\oblix\config`

- Chain file (`ois_chain.pem`)
- Certificate file (`ois_cert.pem`)
- Key file (`ois_key.pem`) the installer may know where this is.

For most components, you install certificates during product setup. You install certificates in the Policy Manager using the `genCert` utility. The command for this utility is:

```
genCert -i <install Dir> -m <cert | simple> -P <password> -c <request | install>
```

For example:

```
genCert -i c:\COREid\webcomponent\access\oblix\tools\gencert -m cert -P <password> -c install
```

You can save an approved certificate to any location that is accessible to the component installer. For example, you can save it to `/oblix/config`.

Note: When using certificates generated by a subordinate CA, the root CA's certificate must be present in the xxx_chain.pem along with the subordinate CA certificate. Both certificates must be present to ensure appropriate verification and successful Identity System setup.

The certificate request for WebGate generates the certificate-request file aaa_req.pem. You need to send this WebGate certificate request to a root CA that is trusted by the AAA server. The root CA returns the WebGate certificates, which can then be installed either during or after WebGate installation.

The following sections describe cert mode, and requesting and installing certificates.

Changing Transport Security for the Identity System

All Identity Servers and WebPass instances in your installation must run in the same transport security mode. If you specified different modes for different components during your installation, you must change them.

Task overview: Changing transport security for the Identity System

1. If you are changing to simple or cert mode, complete the process for certificate preparation.
2. Perform the steps in ["To change the Identity Server transport security mode"](#) on page 8-5.
3. Perform the steps in ["To change the WebPass transport security mode"](#) on page 8-5.

Note: The WebPass and the Identity Server will not be able to communicate with each other until you have changed the transport security mode for both.

To change the Identity Server transport security mode

1. If you are changing to simple or cert mode, complete the certificate preparation process.
2. From the Identity System Console, click the System Configuration sub-tab, then click Identity Server in the left navigation pane.
3. Click the link for the server that you want to modify, then click Modify.
4. Click the appropriate button for the transport security mode of your choice.
You can select Open, Simple, or Cert mode.
5. Click Save.
6. Restart the Identity Server.

To change the WebPass transport security mode

1. If you are changing to simple or cert mode, complete certificate preparation.
2. From the Identity System Console, click the System Configuration sub-tab, then click WebPass in the left navigation pane.
3. Select the WebPass you want to modify and click Modify.
4. Change the transport security mode

You can select Open, Simple, or Cert mode.

5. Click Save.
6. Stop the WebPass, restart the Identity Server, then restart the WebPass.

Transport Security Mode Changes for the Identity System

When changing the transport security mode after installation, specify the new mode in the Identity System Console, then change the mode in the appropriate configuration files. You repeat the steps shown in [Table 8–2](#) as needed for each component.

Table 8–2 *Transport Security Mode Changes for the Identity System*

New Security Mode	Process
Open	Specify Open mode in the Identity System Console (see " Changing Transport Security for the Identity System " on page 8-5 for details).
Simple	1. Stop the Identity Server. 2. Generate the certificate through Oracle Access Manager's internal CA (see "Changing to Simple Transport Security Mode" on page 8-6 for details). 3. Configure the mode in the Identity System Console (see "Changing Transport Security for the Identity System" on page 8-5 for details). 4. Restart the Identity Server.
Cert	1. Stop the Identity Server. 2. Generate the certificate request (see "Changing to Cert Transport Security Mode" on page 8-7 for details). 3. Get the certificate approved through an external CA. 4. Install the certificate (see "To install a certificate for Cert mode" on page 8-8 for details). 5. Configure the mode in the Identity System Console (see "Changing Transport Security for the Identity System" on page 8-5 for details). 6. Restart the Identity Server.

Note: The clocks of computers running Identity System components must be synchronized, especially when the components are using open or cert mode. A difference of a few seconds is allowed as long as the Identity Server computer's clock is ahead of the WebPass computer's clock. Otherwise, certificate time stamps are invalid, and all requests are rejected. See the *Oracle Access Manager Access System Administration Guide* for details about synchronizing system clocks.

Changing to Simple Transport Security Mode

If you want to change to simple mode, you must first generate a certificate through Oracle Access Manager's internal CA.

To generate a certificate through the CA

1. Open a Command Prompt window and go to:

IdentityServer_install_dir/identity/oblix/tools/setup

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

2. Execute one of the following commands, depending on which component you are modifying.

Table 8–3 Setup Commands

Operating System	Commands
UNIX	Identity Server: <code>start_setup_ois -i IdentityServer_install_dir/identity -m</code> WebPass: <code>start_setup_webpass -i WebPass_install_dir/identity -m</code>
Windows	Identity Server: <code>setup_ois.exe -i IdentityServer_install_dir\identity -m</code> WebPass: <code>setup_webpass.exe -i WebPass_install_dir\identity -m</code> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.

You are prompted to enter simple or cert mode.

3. Type simple and press Enter.
4. Specify and confirm the Global Pass Phrase.
This password must be the same across all Identity Servers and WebPass instances within an installation.
5. Continue with "[Changing Transport Security for the Identity System](#)" on page 8-5.

Changing to Cert Transport Security Mode

If you want to change to cert mode, you must do the following after you install a Identity Server:

- Generate a certificate request to obtain a certificate from an external CA.
- Install the signed certificate after you receive it.

To generate a certificate request for Cert mode

1. Open a Command Prompt window and change to
IdentityServer_install_dir/identity/oblix/tools/setup
where *IdentityServer_install_dir* is the directory in which the Identity System has been installed.
2. Run one of the commands in [Table 8–4](#)

Table 8–4 Identity System Request Certificate Commands

Operating System	Commands
UNIX	Identity Server: <code>start_setup_ois -i IdentityServer_install_dir/identity -m</code> WebPass: <code>start_setup_webpass -i WebPass_install_dir/identity -m</code> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.

Table 8–4 (Cont.) Identity System Request Certificate Commands

Operating System	Commands
Windows	Identity Server: <pre>setup_ois.exe -i IdentityServer_install_dir\identity -m.</pre> WebPass: <pre>setup_webpass.exe -i WebPass_install_dir\identity -m</pre> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.

You are prompted to enter simple or cert mode.

3. Type cert and press Enter.
4. Indicate that you are requesting a new certificate.
5. Enter information at the prompts for:
 - A two-letter country code (the default is US).
 - A state or province name.
 - Your city or other locality
 - An organization name (for example, your company)
 - An organizational unit name (for example, your department)
 - A common name (for example, your host name)
 - An email contact address

6. Press Enter.

You see the message:

"Your certificate request is in the file *Identity_Server_install_dir/identity/oblix/config/ois_req.pem*."

The *setup_ois* utility creates two files in this directory: *ois_key.pem*, which contains your private key, and *ois_req.pem*.

7. Submit the *ois_req.pem* file to be signed by your Certificate Authority.

To install a certificate for Cert mode

1. Open a Command Prompt window and change to:


```
Identity_Server_install_dir/identity/oblix/tools/setup
```

 where *IdentityServer_install_dir* is the directory in which the Identity Server is installed.
2. Run one of the commands in [Table 8–5](#).

Table 8–5 Identity System Install Certificate Commands

Operating System	Commands
UNIX	Identity Server: <pre>start_setup_ois -i IdentityServer_install_dir/identity -m</pre> WebPass: <pre>start_setup_webpass -i WebPass_install_dir/identity -m</pre> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.
Windows	Identity Server: <pre>setup_ois.exe -i IdentityServer_install_dir\identity -m</pre> where <i>IdentityServer_install_dir</i> is the directory in which the Identity Server is installed WebPass: <pre>setup_webpass.exe -i WebPass_install_dir\identity -m</pre> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.

You are prompted to enter simple or cert mode.

3. Type cert and press Enter.
4. Indicate that you are installing a certificate.
5. Specify the locations of the following files:

ois_key.pem

ois_cert.pem

ois_chain.pem

If you have installed certificates for an earlier Oracle Access Manager-generated request, use the default value for ois_key.pem when prompted.

Note: When using certificates generated by a subordinate CA, the root CA's certificate must be present in the ois_chain.pem along with the subordinate CA certificate. Both certificates must be present to ensure appropriate verification and successful Identity System setup.

Your certificate is installed.

6. Continue with ["Changing Transport Security for the Identity System"](#) on page 8-5.

Changing Transport Security Modes for the Access System

Before you change the transport security mode for the AccessGate or Access Server, update the transport security modes for the components in the Access System Console.

You cannot update the transport security mode for Policy Manager from the Access System Console. If you are changing from Open mode to another mode, follow the instructions in [Table 8–2](#). If you are changing to Open mode, you need not change the mode for Policy Manager because the Policy Manager automatically detects that the other AccessGate and Access Server are working in Open mode.

To specify transport security mode for Access Server

1. In the Access System Console, navigate to Access System Configuration, Access Server Configuration.
2. Select the Access Server you want to change, and click Modify.
3. Select the appropriate radio button for transport security, and click Save.
4. Restart the Access Server.

To specify transport security mode for AccessGate

1. In the Access System Console, go to Access System Configuration, AccessGate Configuration.
2. Select the AccessGate you want to change, and click Modify.
3. Select the appropriate radio button for transport security, and click Save.
4. Restart the Web server hosting the AccessGate.

Transport Security Mode Changes for the Access System

You can change the transport security mode for Access System components after you have specified the changes in the Access System Console. The process of changing modes depends on the security mode to which you are changing. If you change an Access Server's security mode, you must change the security mode of all Policy Managers and AccessGates pointing to this Access Server to match the new security mode.

If you change the security mode for one or more Access Servers, the Transport Security Mode Change Confirmation page may appear. This page notifies you of an incompatibility between the security modes used by the Access Server and one or more AccessGates.

Note: Configure the Access Server security mode before you configure the mode for an AccessGate/WebGate and Policy Manager.

Table 8–6 lists the process that you follow for each security mode. Repeat these steps as necessary for each installed component.

Table 8–6 Transport Security Mode Changes for the Access System

New Security Mode	Process
Open	Access Server: 1. Move the appropriate directory or files to a new folder (see "Changing to Open Transport Security Mode" on page 8-12 for details). 2. Configure the Access Server instance in the Access System Console (see "To specify transport security mode for Access Server" on page 8-10 for details). 3. Run the configAAAServer program to specify the new mode. For details about using the ConfigureAAAServer Tool, see the <i>Oracle Access Manager Access System Administration Guide</i>. AccessGate/WebGate: 1. Move the appropriate directory or files to a new folder (see "Changing to Open Transport Security Mode" on page 8-12 for details). 2. Configure the AccessGate instance in the Access System Console (see "To specify transport security mode for AccessGate" on page 8-10 for details). 3. Run the configAccessGate or the configWebGate program, as appropriate, to specify the new mode. To modify an AccessGate through the command line, see the <i>Oracle Access Manager Access System Administration Guide</i>. Policy Manager: 1. Restart the Web server on which the Policy Manager is installed.
Simple	Access Server: 1. Move the appropriate directory or files to a new folder (see "Changing to Simple Transport Security Mode" on page 8-6 for details). 2. Configure the Access Server instance in the Access System Console (see "To specify transport security mode for Access Server" on page 8-10 for details). 3. Run the configAAAServer program to specify the new mode. For details about using the ConfigureAAAServer Tool, see the <i>Oracle Access Manager Access System Administration Guide</i>. AccessGate/WebGate: 1. Move the appropriate directory or files to a new folder (see "Changing to Simple Transport Security Mode" on page 8-6 for details). You do not need to do this if you are changing over from Open mode. 2. Configure the new mode for the AccessGate instance in the Access System Console (see "To specify transport security mode for AccessGate" on page 8-10 for details). 3. Run the configAccessGate or the configWebGate program, as appropriate, to specify the new mode. To modify an AccessGate through the command line, see the <i>Oracle Access Manager Access System Administration Guide</i>. Policy Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <code>PolicyManager_install_dir\access\oblix\tools\gencert</code> where <code>PolicyManager_install_dir</code> is the directory in which the Policy Manager is installed.

Table 8–6 (Cont.) Transport Security Mode Changes for the Access System

New Security Mode	Process
Cert	Access Server: 1. Move the appropriate directory or files to a new folder (see "Changing to Cert Transport Security Mode" on page 8-7 for details). 2. Configure the Access Server instance in the Access System Console (see "To specify transport security mode for Access Server" on page 8-10 for details). 3. Run the configAAAServer program to specify the new mode. For details about using the ConfigureAAAServer Tool, see the <i>Oracle Access Manager Access System Administration Guide</i>. AccessGate/WebGate: 1. Move the appropriate directory or files to a new folder (see "Changing to Cert Transport Security Mode" on page 8-7 for details). You do not need to do this if you are changing over from Open mode. 2. Configure the new mode for the AccessGate instance in the Access System Console (see "To specify transport security mode for AccessGate" on page 8-10 for details). 3. Run the configAccessGate or the configWebGate program, as appropriate, to generate the certificate request and install the certificate. To modify an AccessGate through the command line, see the <i>Oracle Access Manager Access System Administration Guide</i>. Policy Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>PolicyManager_install_dir</i>\access\oblix\tools\gencert where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.

Changing to Open Transport Security Mode

If you want to change transport security mode from Simple or Cert to Open, run the appropriate configuration program.

To change to Open security mode

1. Move the following directory to a new folder:

AccessSystem_install_dir/access/oblix/config/simple (if in Simple mode)

or

AccessSystem_install_dir/access/oblix/config/*.pem and password.xml (if in Cert mode)

where *AccessSystem_install_dir* is the directory in which the Access System components are installed. For example, the Policy Manager or Access Server or WebGate.

This saves a previous configuration in case you want to revert to it.

2. Execute one of the commands in [Table 8–7](#).

Table 8–7 Access System Commands: Change to Open Mode

Operating System	Commands
UNIX	Access Server: <pre>start_configureAAAServer reconfig AccessServer_install_dir/access -R</pre> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: <pre>start_configureAccessGate -i AccessGate_install_dir/access -t AccessGate -R</pre> where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed. WebGate: <pre>start_configureWebGate -i WebGate_install_dir/access -t WebGate -R</pre> where <i>WebGate_install_dir</i> is the directory in which WebGate is installed. Policy Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <pre>PolicyManager_install_dir/access/oblix/tools/gencert</pre> where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.
Windows	Access Server: <pre>configureAAAServer.exe reconfig AccessServer_install_dir\access -R</pre> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: <pre>configureAccessGate.exe -i AccessGate_install_dir\access -t AccessGate -R</pre> where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed. WebGate: <pre>configureWebGate.exe -i WebGate_install_dir\access -t WebGate -R</pre> where <i>WebGate_install_dir</i> is the directory in which WebGate is installed. Policy Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <pre>PolicyManager_install_dir/access/oblix/tools/gencert</pre> where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.

Changing to Simple Transport Security Mode

If you want to implement Simple mode, you do not need to request or install a certificate from an external CA. Oracle Access Manager ships with its own internal CA.

To change to Simple security mode

1. Move the following files to a new folder:

AccessSystem_install_dir/access/oblix/config/*.pem

and

AccessSystem_install_dir/access/oblix/config/password.xml (if in Cert mode)

where *AccessSystem_install_dir* is the directory in which the Access System components are installed. For example, the Policy Manager or Access Server or WebGate.

This creates a backup file of your older configuration.

2. Generate a certificate through Oracle Access Manager's internal CA:

- a. Open a command prompt window and change to the appropriate *AccessSystem_install_dir*/access/oblix/tools/*componentDirectory*,

Where:

componentDirectory is the directory for the component you are modifying: *configureAAAServer*, *configureWebGate*, or *genCert* (*genCert* is the utility used by Policy Manager).

For example:

```
cd COREid/WebComponent/access/oblix/tools/configureWebGate
```

- b. Execute one of the commands in [Table 8–8](#).

Table 8–8 Access System Commands: Change to Simple Mode

Operating System	Commands
UNIX	Access Server: <pre>start_configureAAAServer reconfig AccessServer_install_dir/access -R</pre> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: <pre>start_configureAccessGate -i AccessGate_install_dir/access -t AccessGate -R</pre> where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed. WebGate: <pre>start_configureWebGate -i WebGate_install_dir/access -t WebGate -R</pre> where <i>WebGate_install_dir</i> is the directory in which WebGate is installed. Policy Manager: Run the <i>genCert</i> utility to specify the new mode. The <i>genCert</i> utility is located in the directory <i>AccessManager_install_dir</i>\access\oblix\tools\gencert where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.

Table 8–8 (Cont.) Access System Commands: Change to Simple Mode

Operating System	Commands
Windows	Access Server: configureAAAServer.exe reconfig <i>AccessServer_install_dir</i>\access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: configureAccessGate.exe -i <i>AccessGate_install_dir</i>\access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed. WebGate: configureWebGate.exe -i <i>WebGate_install_dir</i>\access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed. Policy Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>PolicyManager_install_dir</i>\access\oblix\tools\gencert where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.

- c. When you are prompted to enter Open, Simple, or Cert mode, select Simple mode and press Enter.

- d. Specify and confirm the Global Pass Phrase.

This password must be the same across all Access Servers and AccessGates and WebGates. For more information on the Global Pass Phrase, see the *Oracle Access Manager Installation Guide*.

WARNING: You need to reinstall the Policy Manager if the Simple mode password for the Policy Manager is changed, or if the Access System is changed from Simple mode to Cert mode

Changing to Cert Transport Security Mode

The following procedure describes changing the transport security mode to Cert.

Note: The certificate request for WebGate generates the certificate-request file *aaa_req.pem*. You need to send this WebGate certificate request to a root CA that is trusted by the AAA server. The root CA returns the WebGate certificates, which can then be installed either during or after WebGate installation.

To change to Cert security mode

1. Move the following to a new folder:

AccessSystem_install_dir/access/oblix/config/simple (if in Simple mode)

This creates a backup of your old configuration

2. Generate a certificate request.
 - a. Open a Command Prompt window and change to the following directory:
`AccessSystem_install_dir/access/oblix/tools/componentDirectory`
 where *AccessSystem_install_dir* is directory in which the Access System components are installed and *componentDirectory* is the directory for the component you are modifying: `configureAAAServer`, `configureWebGate`, `configureAccessGate`, or `genCert` (`genCert` is used by Policy Manager)
 For example:

```
cd COREid/WebComponent/access/oblix/tools/genCert
```
 - b. Execute one of the commands in Table 44, depending on which component you are modifying.

Table 8–9 Access System Request Certificate Commands

Operating System	Commands
UNIX	Access Server: <pre>start_configureAAAServer reconfig AccessServer_install_dir/access -R</pre> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: <pre>start_configureAccessGate -i AccessGate_install_dir/access -t AccessGate -R</pre> where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed. WebGate: <pre>start_configureWebGate -i WebGate_install_dir/access -t WebGate -R</pre> where <i>WebGate_install_dir</i> is the directory in which WebGate is installed. Policy Manager: Run the <code>genCert</code> utility to specify the new mode. The <code>genCert</code> utility is located in the directory <i>PolicyManager_install_dir</i>\access\oblix\tools\gencert where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.

Table 8–9 (Cont.) Access System Request Certificate Commands

Operating System	Commands
Windows	Access Server: configureAAAServer.exe reconfig <i>AccessServer_install_dir</i>\access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: configureAccessGate.exe -i <i>AccessGate_install_dir</i>\access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed. WebGate: configureWebGate.exe -i <i>WebGate_install_dir</i>\access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed. Policy Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>PolicyManager_install_dir</i>\access\oblix\tools\gencert where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.

- c. When you are prompted for a mode, select Cert and press Enter.
- d. Indicate that you are requesting a certificate.
- e. Answer the prompts for information, including the following:
 - A two-letter country code (the default is US)
 - A state or province name
 - Your city or other locality
 - An organization name (your company, for example)
 - An organizational unit name (your department, for example)
 - A common name (must be your host machine name)
 - An email contact address
- f. Press Enter.
A message is displayed stating that your certificate request is in the file *AccessServer_install_dir*/access/oblix/config/aaa_req.pem.
The setup_aaa utility actually creates two files in this directory:
aaa_key.pem, which contains your private key, and aaa_req.pem.
- g. Submit the aaa_req.pem file to the Certificate Authority to get your request signed.
3. Save the approved certificate to a file which the installer can access.
4. Save the CA chain in base64 code format to a .pem file that the installer can access.
5. After you receive the certificate from your CA, install the signed certificate.

To install the signed certificate for Cert mode

- 1. Open a Command Prompt window and change to the *AccessSystem_install_dir/access/oblix/tools/componentDirectory*

where *AccessSystem_install_dir* is the directory in which Access System is installed and *componentDirectory* is the directory for the component you are modifying: *configureAAAServer*, *configureWebGate*, *configureAccessGate*, or *genCert* (*genCert* is the utility used by Policy Manager).

For example:

`cd COREid/access/oblix/tools/configureAAAServer`
- 2. Execute one of the commands in

Table 8–10 Access System Install Certificate Commands

Operating System	Commands
UNIX	Access Server: <code>start_configureAAAServer reconfig AccessServer_install_dir/access -R</code> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: <code>start_configureAccessGate -i AccessGate_install_dir/access -t AccessGate -R</code> where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed. WebGate: <code>start_configureWebGate -i WebGate_install_dir/access -t WebGate -R</code> where <i>WebGate_install_dir</i> is the directory in which WebGate is installed. Policy Manager: Run the <i>genCert</i> utility to specify the new mode. The <i>genCert</i> utility is located in the directory <i>PolicyManager_install_dir/access/oblix/tools/genCert</i> where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.

Table 8–10 (Cont.) Access System Install Certificate Commands

Operating System	Commands
Windows	Access Server: configureAAAServer.exe reconfig <i>AccessServer_install_dir</i>\access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: configureAccessGate.exe -i <i>AccessGate_install_dir</i>\access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed. WebGate: configureWebGate.exe -i <i>WebGate_install_dir</i>\access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed. Policy Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>PolicyManager_install_dir</i>\access\oblix\tools\gencert where <i>PolicyManager_install_dir</i> is the directory in which the Policy Manager is installed.

3. When you are prompted to enter Simple or Cert mode, type Cert and press Enter.
4. Indicate that you are installing a certificate.
5. Specify the locations of the key, server certificate, and CA chain files:
 - aaa_key.pem
 - aaa_cert.pem
 - aaa_chain.pem

where aaa is the name you specify for the file (applicable only to Cert and chain files).

WARNING: The Webgate certificate request generates the certificate-request file *aaa_req.pem*. You need to send this certificate request to a root CA that is trusted by the AAA server. The root CA returns the WebGate certificates, which can then be installed either during or after WebGate installation.

If you have installed certificates for an earlier Oracle Access Manager-generated request, use the default value for *aaa_key.pem* when prompted.

Your certificate is installed.

6. Restart the AccessGate or Access Server, as appropriate.

Transport Security Changes for Directory Servers

When you install the Identity Server and the Access Server, you can specify Open or SSL mode between each of these servers and the directory server. To change the transport security mode after installation, you must reconfigure the Identity Server or

the Access Server, as appropriate. During reconfiguration, you can change the security mode between the directory server and the Identity or Access Server.

Note: See the *Oracle Access Manager Installation Guide* for additional information about adding directory certificates after installation.

To change transport security between the Identity Server and directory server

1. From a command line, find the appropriate setup_ois tool for your platform.

On UNIX, for example:

```
IdentityServer_install_dir/identity/oblix/tools/setup
```

2. At the command prompt, run the appropriate executable.

On UNIX, for example:

```
start_setup_ois -i
```

You are guided through the steps required to set up the Identity Server.

3. When you are asked whether you want SSL between the Identity Server and the directory server, select either y (yes) or n (no).

Note: If you select SSL, provide the full path to the location of the CA certificate when asked.

4. Complete the rest of the steps to finish the reconfiguration process.

To change transport security to SSL between the Policy Manager and directory server

1. From a command line, find the appropriate setup_access_manager tool for your platform.

On UNIX, for example:

```
PolicyManager/identity/oblix/tools/setup
```

2. At the command prompt, run the appropriate executable to create the cert8.db file.

On UNIX, for example:

```
start_setup_access_manager -i
```

You are guided through the steps required to set up the Policy Manager.

3. When you are asked, provide the full path of the file containing the Root CA certificate for the directory server.
4. Complete the rest of the steps to finish the reconfiguration process.

To change transport security between the Access Server and the directory server

1. From a command line, navigate to the folder where the configureAAAServer tool is located.

For example:

```
AccessServer_install_dir/access/oblix/tools/configureAAAServer
```

2. At the command line, run the following executable:

```
start_configureAAAServer -i
```

Note: On non-Windows systems, use the start_configureAAAServer tool.

3. Select 1 (Y) to reconfigure the Access Server.

You are guided through the steps required to set up the Access Server. Specify the required information.

4. When you are asked to specify the mode for the directory server, select either Open or SSL.
5. If you select SSL, provide the full path to the location of the CA certificate.
6. Complete the rest of the steps to finish the reconfiguration process.

Changing Transport Security Passwords

When communicating with each other, components authenticate one another using a password-based mechanism.

- **Simple Mode:** In Simple mode, all components in an Identity or Access System must use the same password within the installation. Oracle Access Manager generates certificates that are required by Transport Layer Security (TLS). Any installation can generate valid certificates.
- You can store the password in a local file so that each component can start unattended. Or you may have the component prompt for the password when it starts. Prompting requires a system administrator to start each element manually and type the password.
- **Cert Mode:** Cert mode requires a password for each component's private key file. You can use a different password for each component.

As with Simple mode, you can store the password in a local file so that each component can start unattended, or you may have the component prompt for the password when it starts. Prompting requires a system administrator to start each component manually and type the password.

You can change the password for Cert or Simple transport security mode.

To change the certificate password for the Identity System

1. Open a Command Prompt window and change to the *IdentityServer_install_dir/identity/oblix/tools/setup* directory, where *IdentityServer_install_dir* is the directory in which the Identity Server is installed.

For example:

```
cd COREid/identity/oblix/tools/setup
```

2. Run one of the commands in [Table 8-11](#).

Table 8–11 Identity System Commands for Certificate Password Changes

Operating System	Commands
UNIX	Identity Server: <pre>start_setup_ois -i IdentityServer_install_dir/identity -k</pre> where <i>IdentityServer_install_dir</i> is the directory in which the Identity Server is installed. WebPass: <pre>start_setup_webpass -i WebPass_install_dir/identity -k</pre> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.
Windows	Identity Server: <pre>setup_ois.exe -i IdentityServer_install_dir\identity -k</pre> where <i>IdentityServer_install_dir</i> is the directory in which the Identity Server is installed. WebPass: <pre>setup_webpass.exe -i WebPass_install_dir\identity -k</pre> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.

3. Specify the transport security mode this component is using.
4. Specify the old password.
5. Specify and confirm the new password.
6. Restart the Identity Server.

To change the certificate password for the Access System

1. Open a Command Prompt window and change to the *AccessSystem_install_dir/access/oblix/tools/componentDirectory*

where *AccessSystem_install_dir* is the directory in which the Access System is installed and *componentDirectory* is the directory for the component you are modifying.

For example:

```
cd COREid/access/oblix/tools/configureAccessGate
```
2. Run one of the commands in [Table 8–12](#).

Table 8–12 Access System Commands for Certificate Password Changes

Operating System	Commands
UNIX	Access Server: <pre>start_configureAAAServer chpasswd AccessServer_install_dir/access</pre> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: <pre>start_configureAccessGate -i AccessGate_install_dir/access -t AccessGate -k</pre> where <i>AccessGate_install_dir</i> is the directory in which the Access Server is installed. WebGate: <pre>start_configureWebGate -i WebGate_install_dir/access -t WebGate -k</pre> where <i>WebGate_install_dir</i> is the directory in which the Access Server is installed.
Windows	Access Server: <pre>configureAAAServer.exe chpasswd AccessServer_install_dir\access</pre> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed. AccessGate: <pre>configureAccessGate.exe -i AccessGate_install_dir\access -t AccessGate -k</pre> where <i>AccessGate_install_dir</i> is the directory in which the Access Server is installed. WebGate: <pre>configureWebGate.exe -i WebGate_install_dir\access -t WebGate -k</pre> where <i>WebGate_install_dir</i> is the directory in which the Access Server is installed.

3. Specify the transport security mode this component is using.
4. Specify the old password.
5. Specify and confirm the new password.
6. Restart the Access Server.

Importing Multiple CA Certificates

Oracle Access Manager recognizes one CA certificate for each directory server type for transport security between a component and the directory server for user data, configuration data, or policy data.

If your implementation has separate directory servers for user data, configuration data, or policy data, you can have separate CA certificates for each. Thus you can have up to three CA certificates in your implementation; one for the user directory, one for the configuration directory, and one for the policy directory.

WARNING: If your installation uses replicated or multiple directories that have established SSL using certificates from different certificate authorities, you need to import the various certificates manually into the `cert8.db` file. The `cert8.db` file is encrypted and stored in a proprietary Mozilla format.

For more information about adding directory server CA certificates, see the *Oracle Access Manager Installation Guide*.

Changing Access Server Security Password

You can change the Access Server transport security mode from the command line. For Simple mode, the AccessGate or WebGate and the Access Server must have the same password to allow them to communicate with each other.

To change the transport security mode password

1. Run the following executable:

```
configureAAAServer chpasswd AccessServer_install_dir
```

where *AccessServer_install_dir* is the directory in which the Access Server is installed.

2. Specify the following when prompted:
 - The transport security mode in which the Access Server is configured.
 - The old password
 - The new password
3. Restart the Access Server.

See "[About Transport Security Modes](#)" on page 8-1 for more information.

Reporting

This chapter provides an overview of reporting features, the information each feature presents, the types of output available, and possible uses for these reports. This chapter covers the following topics:

- [About Reporting](#)
- [Summary of Reporting Features](#)

About Reporting

Oracle Access Manager can collect and present a wide range of information related to the following:

- Users and resources in your Oracle Access Manager directory
- Activities on the Access and Identity Systems
- The operation, administration, and maintenance of your system

To help distinguish among the many report-related features built into Oracle Access Manager, this chapter reserves certain terms to describe specific functional areas, as explained in the following table:

Table 9–1 *Reserved Terms Used for Reporting*

Feature	Description
Monitoring	Refers exclusively to the SNMP data collected so that you can monitor the health and performance of the network components that host your system. For a complete discussion of SNMP Monitoring, see " SNMP Monitoring " on page 12-1.
Logging	Refers exclusively to program execution data collected so that you can diagnose the health of the components that make up your system, troubleshoot execution errors, and debug custom AccessGates and other plug-ins. For a complete discussion of logging, see " Logging " on page 10-1.

Table 9–1 (Cont.) Reserved Terms Used for Reporting

Feature	Description
Auditing	Refers to two types of data: - Dynamic audit data is collected from Access Servers and Identity Servers. It encompasses Oracle Access Manager system events such as resource requests, password changes, and account revocation. - Static audit data is collected from the directory server. It encompasses policy and profile information. For a general discussion of static and dynamic reports, see "Report Types" on page 9-2. For a complete discussion of auditing, see "Auditing" on page 11-1.
Diagnostics	The Access Server and Identity Server provide diagnostic tools to help you work with an Oracle Technical Support representative to troubleshoot problems. See "Capturing Diagnostic Information" on page F-13 for details. You can also collect information about parameter settings and states for Access Servers, Identity Servers, and their connections to the Oracle Access Manager directory components. See Table 11–1 on page 11-3 for details.
Access Testing	Refers exclusively to the on-screen display that provides a quick way of determining whether a given user has access to a given resource at a given time. For more on access testing, see Table 11–1 on page 11-3.
Filtered Queries	Refers to the advanced searches of the directory conducted through various Oracle Access Manager applications to generate lists of users or resources that share certain combinations of profile or policy attributes. For more on advanced filtered queries, see: Table 11–1 on page 11-3.
Audit Reports	Refers exclusively to data that is collected from the Oracle Access Manager servers and directory server, stored in the audit database, then extracted, compiled, and formatted by preconfigured Crystal Reports presentation templates. For a complete discussion of Audit Reports, see "About Audit Reports" on page 11-15 and "Setting up Audit Reports" on page 11-44.

Report Types

The information collected and reported by the various reporting features falls into two broad categories:

- Static reports:** Generally compiled from settings stored on Oracle Access Manager components or third-party related components. For example, policy and profile information stored on the Oracle Access Manager directory server is classified as static audit data. Connection settings (and states) fall into the Diagnostic category. Certain Audit Reports use static (stored) policy and profile information to compile a list of resources that are available to specified users during specified times.
- Dynamic Reports:** Focus on events and changes in state at various levels throughout the Oracle Access Manager system. For example, the logging feature can record each function call (and outcome) originating from a given component. This low-level trace capability can be useful to developers. At the other end of the spectrum, the dynamic audit feature can reveal system intrusion threats by reporting patterns of failed authentication attempts on specific servers during a specific interval.

Data Sources

The reporting features can gather data from a variety of sources, the most important of which are covered in [Table 9–2](#)

Table 9–2 Primary Data Sources for the Reporting Features

Data Source	Description
Oracle Access Manager directory	Stores several types of static information, including the following: ■ User, group, and organization profile settings ■ Policy settings for protecting resources ■ Connections settings such as those used to connect with Oracle Access Manager components or the various databases used by Oracle Access Manager ■ Certain security settings ■ Schema used to organize the LDAP directory at the heart of the Oracle Access Manager system
Component configuration files	Many key settings reside in configuration files stored within the directory structure of the Oracle Access Manager component they affect. This can range from the path to a database driver to the size of the buffer used for queuing log output.
System configuration files	These settings for the machines that host the various Oracle Access Manager components can be environment variables that make components visible to each other, or they can be protocol settings that enable components to communicate at the same level. Generally, Oracle Access Manager does not report such system-level configurations directly, but it can sometimes report corresponding settings that must match the settings established at the host system level.
Access Servers	In addition to providing configuration information about the settings they maintain to interact with other components, Access Servers can report Access System events such as authorization requests and their outcomes. This information is useful for determining who has gained (or tried to gain) access to what during a certain interval.
Identity Servers	Identity Servers also store certain settings that govern how they interact with other components. Additionally, they report Identity System events such as who attempted to submit credentials at what time, and whether that authentication attempt succeeded.
Other components	Components such as the Policy Manager can report changes to policies and certain other activities and settings.

Data Output

Generally, the various types of reports can send data to one or more of the following destinations:

- The Oracle Access Manager graphical user interface
- A plain text file on the machine hosting the component that is sending the data
- A system file on the machine hosting the component that is sending the data
- A central database

Note: When data is sent to the audit database, it is generally filtered, compiled, and presented using special Crystal Reports templates that generate Audit Reports.

When a report is sent to the graphical user interface, it is likely to be somewhat less extensive than the equivalent type sent to a file or database. For instance, the on-screen Access Tester tool cannot report on the kind of complex user and resource groups that are available through the User Access Privilege tool, which sends output to a plain-text file or the audit database.

Output Configuration

Generally, you can format report output in one or both of the following ways:

- Through the Oracle Access Manager graphical user interface
- By manually editing a plain-text configuration file.

In a limited number of cases and to a limited extent, you can configure report output through a third-party GUI. For example, you can edit the templates used to generate the Audit Reports through the Crystal Reports interface.

Data Uses

Reports can prove useful to a variety of people, including the following:

- Administrators for Oracle Access Manager
- Network administrators
- Security administrators
- Compliance administrators
- Custom AccessGate and plug-in developers

Summary of Reporting Features

Table 9–3 provides an overview the reporting features, the information they present, and potential uses to which these features can be applied.

Table 9–3 Overview of Reporting Features

Feature	Type	Output	Source	Data	Potential uses
Monitoring	Dynamic	File	SNMP monitor	Network component states and events	Monitoring and troubleshooting the network hosting your Oracle Access Manager system
Logging	Dynamic	File	Oracle Access Manager components	Program execution (states and events)	Diagnosing component health and debugging custom AccessGate and plug-in code

Table 9–3 (Cont.) Overview of Reporting Features

Feature	Type	Output	Source	Data	Potential uses
Auditing	Dynamic	File, DB	Oracle Access Manager servers	System events	Tracking usage patterns, system performance, component loading, and security compliance
Auditing	Static	File, DB	directory server	Profile and policy attributes	Identifying users and resources that fit specified patterns
Diagnostics	Static or dynamic	GUI or file	directory server, Oracle Access Manager servers	Directory component, server, and connection settings and states; all program and thread calls	Verifying server and directory server settings, states, and connection details; taking stack traces
Access Tests	Static	GUI	directory server	Profile and policy attributes	Quick determination of who has access to what at a given time.
Filtered Queries	Static	GUI, file	directory server	Profile and policy attributes	Reporting on complex combinations of shared profile and policy attributes
Audit Reports (from Crystal Report templates by way of the audit database)					
Global Access	Static	GUI, file, hardcopy	directory server by way of audit db	Profile and policy attributes	Advanced reports on user and resource access privileges
Authentication	Dynamic	GUI, file, hardcopy	component servers by way of audit db	Authentication events	Statistics on authentication events
Authorization	Dynamic	GUI, file, hardcopy	component servers by way of audit db	Authorization events	Statistics on authorization events
Activity	Dynamic	GUI, file, hardcopy	component servers by way of audit db	Access and Identity System events	Statistics on and lists of various Oracle Access Manager events
ID history	Dynamic	GUI, file, hardcopy	component servers by way of audit db	Profile attributes and changes to attributes	Statistics on and lists of identity profile changes

An Oracle Access Manager component instance can write information about its processes and states to a log file. The logs can be configured to provide information at various levels of granularity. For example, you can record errors, errors plus state information, or errors, states, and other information to the level of a debug trace.

You can configure different logging levels for different functional areas of a component instance. For example, you can capture debug data for LDAP activity while recording only error-level data for all other component activity. You can also record the time taken for each request that a component processes, and you can send different levels of log data to different destinations. For example, you can send error information to a file and all other log data to the system log.

This chapter discusses following topics:

- [About Logging, Log Levels, and Log Output](#)
- [About Log Configuration File Paths and Contents](#)
- [About Directing Log Output to a File or the System File](#)
- [Structure of the Log Configuration File](#)
- [About Activating and Suppressing Logging Levels](#)
- [Log Configuration Parameters](#)
- [Configuring Logs in the Identity System Console](#)
- [Configuring Different Threshold Levels for Different Types of Data](#)
- [Logging the Amount of Time to Process Requests](#)

Note: For an overview of other reporting features, including system event auditing, diagnostics, and SNMP monitoring, see "[Reporting](#)" on page 9-1.

About Logging, Log Levels, and Log Output

The logging feature enables you to analyze system performance and health, and to troubleshoot issues.

You can configure logging for individual instances of the following components:

- Access Servers
- Identity Servers
- Policy Managers

- WebPasses
- WebGates
- Custom AccessGates
- Custom plug-ins

You configure logging by editing a configuration file that is stored with the component. For Identity Servers only, you can also set some parameters for logging in the Identity System Console.

You can request logging at various levels. The highest level is Fatal and the lowest level is Trace. See ["About Log Levels"](#) on page 10-2 for details.

In the log configuration file, a parameter known as a log writer determines the destination for log output. See ["About Directing Log Output to a File or the System File"](#) on page 10-8 for details. You create a complete definition for your log output by identifying a log writer and a log level. This complete definition is known as a log-handler. See ["The Second Compound List and Log Handlers"](#) on page 10-11 for details.

For analysis of performance issues, the logs can capture data on the amount of time it takes to perform specific operations. See ["Logging the Amount of Time to Process Requests"](#) on page 10-25 for details.

The rest of this section discusses the following topics:

- [About Log Levels](#)
- [About Capturing Stack Trace Data in the Log Files](#)
- [About Log Output](#)

About Log Levels

A logging level determines the amount of data that is written to the log data file. Each logging level is cumulative, that is, each level contains all the data generated by the higher levels. For example, Error logs contain all the data generated by the Fatal logs, plus the events that are specific to the Error category.

The default log level is Warning.

[Table 10–4](#) describes the levels.

Table 10–1 Logging Levels

Level	Number of Events Reported	Description
LOGLEVEL_ FATAL	> 60	Records critical errors. Generally, these events can cause the component to exit. In the event of a system failure, Fatal-level messages are always flushed to the log file.
LOGLEVEL_ ERROR	> 960	Records events that may require corrective action, for example, a component is unavailable. Error logs can also be generated for transient or self-correcting problems, for example, failure to connect to another component.
LOGLEVEL_ WARNING	> 1200	Records issues that may lead to an error or require corrective action in the future.

Table 10–1 (Cont.) Logging Levels

Level	Number of Events Reported	Description
LOGLEVEL_INFO	> 400	Records completed actions or the current state of a component, for example, the component is initializing.
LOGLEVEL_DEBUG1	> 400	Records debugging information. Typically, the information at this level is only meaningful to a developer.
LOGLEVEL_DEBUG2	> 100	Records advanced debugging information. This level augments the Debug1 log level. Typically, the information at this level is only meaningful to a developer.
LOGLEVEL_DEBUG3	> 900	Records a large amount of debugging information or data pertaining to an expensive section of the code. This level is useful for debugging a tight loop or a performance-sensitive function. Typically, the information at this log level is only meaningful to a developer. These logs can contain sensitive information.
LOGLEVEL_TRACE	> 900 Oracle Access Manager API > 150 third-party API	This log level is used to trace code path execution or to capture performance metrics. This information is captured at the entry and exit points for each component function. Typically, the information at this log level is only meaningful to a developer. These logs can contain sensitive information.
LOGLEVEL_ALL	> 5000	This level includes all the events and states from all other levels.

You can collect log data from non-adjacent levels and send different levels of log data to different destinations. For example, you can send the Fatal logs to the system log, and write Error logs to a file. See ["The Second Compound List and Log Handlers"](#) on page 10-11 for details.

You configure a global cutoff, or threshold, for logging on the LOG_THRESHOLD_LEVEL parameter in the log configuration file. By default, if a configured level for a log-handler exceeds the cutoff, the log data is not collected. Note that logs can fail to be written despite the configured level because the LOG_THRESHOLD_LEVEL parameter takes precedence over the level configured in the log-handler. Only the MODULE_CONFIG section of the log configuration file overrides the global threshold. See ["The Simple List, Logging Threshold, and Identity System Console Synchronization"](#) on page 10-10 for details.

You specify function- or module-specific overrides for the global logging threshold on the MODULE_CONFIG parameter. See ["Configuring Different Threshold Levels for Different Types of Data"](#) on page 10-19 for details.

Note: The Trace and Debug3 level logs can contain sensitive information.

About Capturing Stack Trace Data in the Log Files

If you have enabled logging for a component instance and the instance crashes, Oracle Access Manager writes stack trace information to the log files. You can also generate stack traces on demand.

See ["Capturing Diagnostic Information"](#) on page F-13 for details.

About Log Output

Each line of the log output file follows a particular structure. A line starts with a date and time stamp, followed by the thread that is processing the request, the name of the function or module being logged, and the log level.

The following is a snapshot of the leftmost columns of the log output file:

```
2007/06/01@00:50:56.859000    5932  2672  DB_RUNTIME    DEBUG3
2007/06/01@00:50:56.859000    5932  2672  DB_RUNTIME    TRACE
2007/06/01@00:50:56.859000    5932  2672  LDAP          DEBUG1
2007/06/01@00:50:56.859000    5932  2672  LDAP          TRACE
2007/06/01@00:50:56.859000    5932  2672  LDAP          TRACE
```

The two columns to the right of the log level are internal code references, and can be ignored. The following is an example of these columns:

```
0x00000205    ldap_connection_mgr.cpp:212
```

To the right of the internal code reference columns, you see the log message that is associated with this log level, for example, "Function called" or "Function returned," followed by the name of the function, as illustrated in the following example:

```
"Function called"    _CallName^ldap_init
```

The log message and function name can be followed by additional information, for example, the duration of the process, the address space where the function is running, or state information, as illustrated in the following examples:

```
"Connection health check result"    Server^dlsun4072    Port^389    Server Priority^1
Connection available^true
```

```
"Function entered"    _TraceName^ConnectionWatcherThread::CheckPrimaries
```

```
"Function exited"    _TraceName^ConnectionWatcherThread::CheckPrimaries
TraceDuration^0.000028
```

```
"Connection Pool Status in ValidateConnections()"    "NumLivePrimaryConnections^1
Maximum Connections^1    UpConnections^1    Failover Threshold^1    Max Session
Time^0    SleepFor^60
```

About Log Configuration File Paths and Contents

You configure parameters that control log output in XML-based log files that you edit with a plain text editor. Changes that you make to these files are effective immediately.

The rest of this section discusses the following topics:

- [Log Configuration File Paths and Names](#)
- [Log Configuration File Contents](#)

Log Configuration File Paths and Names

Each Oracle Access Manager component has a log configuration file where you define what type of data is recorded in the log output. A log configuration file is distinct from the log output file. For details on log output files, see ["About Log Output"](#) on page 10-4.

Log configuration files reside in the following location:

Component_install_dir\identity | access\oblix\config

Where *Component_install_dir* is the directory where you installed the component. Do not change the paths.

If you install more than one instance of a component, for example, if you install multiple Identity Servers, a log configuration file is installed for each instance.

[Table 10-2](#) lists the names of the log configuration files. Do not change the names.

Table 10-2 Log Configuration File Names for Components

Component	Log Configuration File Name
Access Server	oblog_config.xml
Identity Server	oblog_config.xml
Policy Manager	oblog_config_am.xml
WebGate	oblog_config_wg.xml
WebPass	oblog_config_wp.xml
Access Manager SDK (custom AccessGate)	oblog_config.xml

Important: Do not change the default path or name for the configuration file.

Log Configuration File Contents

The log configuration file controls items such as the following:

- What is logged for that component
- Where the data is sent
- In certain cases, the size of the write buffer used for the log
- Log file rotation intervals

The configuration file contains XML statements that you can edit in a text editor.

For Identity Servers only, you can also modify configuration parameters in the log file through the Identity System console, provided that the AUTOSYNC parameter in the configuration file is set to the default value, True. See ["Configuring Logs in the Identity System Console"](#) on page 10-17 for details.

When Changes to the File Take Effect

A watcher thread picks up changes to the log configuration file every 60 seconds and ensures that changes take effect. It is unnecessary to restart the server.

For Identity Servers, if you modify the logging settings using the Identity System Console, the changes are written to the log configuration file if the AutoSync

parameter in the file is set to `True`. If this parameter is set to `False`, the old configuration file are reinstated after the server is restarted.

About Comments in the Log File

Each default log configuration file contains comments that are intended to assist with editing the file. Comments can span one or multiple lines. Comments look similar to the following:

```
<!-- A sample comment -->
```

[Example 10–1](#) shows a typical log configuration file with comments. [Example 10–6](#) shows an example of a log file without comments.

Example 10–1 The Default Log Configuration File with Comments

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--===== -->
<!--===== -->
<!--NetPoint Logging Configuration File -->
<!-->
<!--Changes to this file to take effect upon saving the file. -->
<!-->
<!-->
<!--===== -->
<!--===== -->
<!--Set the Log Threshold -->
<!-->
<!--The log Threshold determines the amount of information to log. -->
<!--Selecting a lower level of logging includes the information -->
<!--logged at the higher levels. For example, LOGLEVEL_ERROR -->
<!--includes the information collected at LOGLEVEL_FATAL. -->
<!-->
<!--Choices are: -->
<!--LOGLEVEL_FATAL - serious error, possibly a program halt. -->
<!--LOGLEVEL_ERROR - a transient or self-correcting problem. -->
<!--LOGLEVEL_WARNING - a problem that does not cause an error. -->
<!--LOGLEVEL_INFO - reports the current state of the component. -->
<!--LOGLEVEL_DEBUG1 - basic debugging information. -->
<!--LOGLEVEL_DEBUG2 - advanced debugging information. -->
<!--LOGLEVEL_DEBUG3 - logs performance-sensitive code. -->
<!--LOGLEVEL_TRACE - used when you need to trace the code path -->
<!--execution or capture metrics. Includes all previous levels. -->
<!-->
<!--If you do not specify a threshold, the default is WARNING. -->
<!-->
<!--In addition to specifying a threshold, you need to specify -->
<!--if changes that you make to the logging configuration in -->
<!--the GUI overwrite the settings in this file. The -->
<!--AutoSync parameter accomplishes this. This parameter takes a -->
<!--value of True or False. If set to True, changes made in the -->
<!--GUI overwrite changes in this config file. If False, changes -->
<!--made in the GUI are only in effect until the server is -->
<!--stopped or restarted. The default is True. -->
<!-->
<!-->
<CompoundList xmlns="http://www.oblix.com" ListName="logframework.xml.staging">
  <SimpleList>
    <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
    <NameValPair ParamName="AUTOSYNC" Value="True" />
```

```

        </SimpleList>
<!-->
<!-->
<!--===== -->
<!--===== -->
<!--Configure the Log Level -->
<!-->
<!-->
<!--To configure a log level, you specify a name for the -->
<!--configuration (for instance, MyErrorLog1) and -->
<!--the log level that you are configuring. You can create -->
<!--more than one configuration per log level if you want -->
<!--to output to more than one destination. You can output to -->
<!--the system log or to a file, as specified on -->
<!--the LOG_WRITER parameter. The value for the LOG_WRITER -->
<!--parameter may only be SysLogWriter, FileLogWriter or -->
<!--MPFileLogWriter. The MPFileLogWriter is a multi-process safe -->
<!--FileLogWriter. It should be used to log in webcomponents i.e -->
<!--WebGate, Policy Manager and WebPass loaded on multiprocess -->
<!--web servers like Apache and IPlanet(Unix) -->
<!-->
<!--If you do not specify an output destination, the default is -->
<!--SysLogWriter. -->
<!-->
<!--If outputting to a file, you also specify a file name and -->
<!--other parameters. Default parameter values are: -->
<!--FILE_NAME: <installdir>/oblix/log/oblog.log -->
<!--BUFFER_SIZE: 32767 (number of bytes) -->
<!--MAX_ROTATION_SIZE: 5242880 (bytes, equivalent to 5MB) -->
<!--MAX_ROTATION_TIME: 86400 (seconds, equivalent to one day) -->
<!-->
<!--Configuring the log level does not ensure that the data is -->
<!--actually collected. Data collection for a log is -->
<!--determined by the LOG_THRESHOLD_LEVEL parameter, above, -->
<!--and the LOG_STATUS parameter in the log configuration. -->
<!-->
<!--If you do not provide a LOG_STATUS, the default for -->
<!--LOGLEVEL_FATAL, LOGLEVEL_ERROR, and LOGLEVEL_WARNING, -->
<!--is On. -->
<!-->
<!--This file contains several sample configurations that are -->
<!--enclosed in comments. To use them, remove the comments. -->
<!-->
        <CompoundList xmlns="http://www.oblix.com" ListName="LOG_CONFIG">
<!--Write all FATAL logs to the system logger. -->
        <ValNameList xmlns="http://www.oblix.com" ListName="LogFatal2Sys">
                <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL" />
                <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
                <NameValPair ParamName="LOG_STATUS" Value="On" />
        </ValNameList>
<!--Write all ERROR logs to the system logger. -->
        <ValNameList xmlns="http://www.oblix.com" ListName="LogError2Sys">
                <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
                <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
                <NameValPair ParamName="LOG_STATUS" Value="On" />
        </ValNameList>
<!--Write all WARNING logs to the system logger. -->
        <ValNameList xmlns="http://www.oblix.com" ListName="LogWarning2Sys">
                <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_WARNING" />
                <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />

```

```

        <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
<!--Write all logs to the Oblix log file. -->
    <ValNameList xmlns="http://www.oblix.com" ListName="LogAll2File">
        <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ALL" />
        <NameValPair ParamName="LOG_WRITER" Value="FileLogWriter" />
        <NameValPair ParamName="FILE_NAME" Value="oblog.log" />
    <!--Buffer up to 64 KB (expressed in bytes) of log entries before flushing to the
    file. -->
        <NameValPair ParamName="BUFFER_SIZE" Value="65535" />
    <!--Rotate the log file once it exceeds 50 MB (expressed in bytes). -->
        <NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" />
    <!--Rotate the log file after 24 hours (expressed in seconds). -->
        <NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" />
        <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>

<!-- List of values that can be specified in the module config -->
<!-- -->
<!-- On - Uses loglevel set in the loglevel threshold -->
<!-- Off - No information is logged -->
<!-- LOGLEVEL_FATAL - serious error, possibly a program halt. -->
<!-- LOGLEVEL_ERROR - a transient or self-correcting problem. -->
<!-- LOGLEVEL_WARNING - a problem that does not cause an error. -->
<!-- LOGLEVEL_INFO - reports the current state of the component. -->
<!-- LOGLEVEL_DEBUG1 - basic debugging information. -->
<!-- LOGLEVEL_DEBUG2 - advanced debugging information. -->
<!-- LOGLEVEL_DEBUG3 - logs performance-sensitive code. -->
<!-- LOGLEVEL_TRACE - used when you need to trace the code path -->
<!-- execution or capture metrics. Includes all previous levels. -->
<!-- -->
<!-- List of modules that can be specified in the module config -->
<!-- -->
<!-- ALL_MODULES - Applies to all log modules -->
<!-- Specific module name - Applies to specific module -->
<!-- -->
<!-- -->
<!-- <ValNameList -->
<!-- xmlns="http://www.oblix.com" -->
<!-- ListName="MODULE_CONFIG"> -->
<!-- <NameValPair -->
<!-- ParamName="CONNECTIVITY" -->
<!-- Value="LOGLEVEL_TRACE"></NameValPair> -->
<!-- </ValNameList> -->

</CompoundList>

```

Note: If you modify logging settings in the Identity System Console, the log configuration file for the associated component instance is saved without comments. See ["Configuring Logs in the Identity System Console"](#) on page 10-17 for details.

About Directing Log Output to a File or the System File

To send log output to a destination, you configure a log writer. A log writer can send log output to one, none, or both of the following:

- A log file.

This 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 Trace data to a file. Or, you can send Fatal data to both the system log and a file.

You define log writers in the log configuration file using the `LOG_WRITER` parameter in a log-handler definition. See ["The Second Compound List and Log Handlers"](#) on page 10-11 for details.

The log writers are described in [Table 10-3](#).

Table 10-3 Log Writers

Writer	Description
SysLogWriter	Sends data to the system log file for the computer that hosts the component being logged. Typically, the system log file contains event information from multiple applications and the host operating system. For Windows, this is the application log file located at My Computer, Manage, Event Viewer, Application. For Unix platforms, the name and location of the system log file can vary according to the computer and the preferences of the system administrator. Consult the administrator of the computer for the file location. The default log configuration file sends Fatal, Error, and Warning messages to the system log file.
FileLogWriter	This writer is recommended when you want to save log data for an Access Server, Identity Server, or other single-process application on a disk file. The FileLogWriter opens the log file and holds it open for disk writes until the approximate file size limit or file rotation interval has been reached. Oracle does not recommend this log writer for situations where more than one process needs to write to the same log file. For these situations, use the MPFileLogWriter.
MPFileLogWriter	This writer resembles the FileLogWriter, except that it opens and closes the log file each time it writes data to the file. This enables multiple processes to write to the file in turn. However, this practice can slow performance substantially. Oracle recommends using MPFileLogWriter only when FileLogWriter fails to record logging data from some of the processes associated with a multi-process application, for example, an AccessGate installed on a multi-process Web server (such as Apache) or the Linux or Solaris versions of the iPlanet Web server.

Structure of the Log Configuration File

The log configuration file conforms to a standard format. You can edit parameters and add or subtract sections known as log-handler definitions, but do not change the underlying format of the log configuration file.

See [Example 10-1](#) or [Example 10-6](#) for a listing of the default log configuration file.

The rest of this section discusses the following topics:

- [The Log Configuration File Header](#)

- [The Initial Compound List](#)
- [The Simple List, Logging Threshold, and Identity System Console Synchronization](#)
- [The Second Compound List and Log Handlers](#)
- [The List for Per-Module Logging](#)
- [About XML Element Order](#)

The Log Configuration File Header

At the beginning of the log configuration file there is an XML file header:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

The header serves the following purposes:

- The header declares the relevant XML version, which is always 1.0.
- It also declares the encoding format, which is always ISO-8559-1.

The Initial Compound List

The header is followed by an initial compound list that is delimited as follows:

```
<CompoundList xmlns="http://www.oblix.com" ListName="logframework.xml.staging">
. . .
</CompoundList>
```

The first compound list is structured as follows:

- The compound list start-tag shows the relevant XML name space for the log configuration file in the `xmlns` parameter.
- The compound list start-tag also provides a name for the compound list in the `ListName` parameter.
- The compound list end-tag occurs near the end of the file.

This compound list delimits all log configuration information.

The Simple List, Logging Threshold, and Identity System Console Synchronization

After the start-tag for the first compound list, a simple list sets the global defaults for logging, as follows:

```
<SimpleList>
. . .
</SimpleList>
```

Between the start and end tags of the simple list, you configure the following:

- The `LOG_LEVEL_THRESHOLD` parameter sets the default logging threshold, as shown in the following example:

```
<NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
```

The global threshold allows logs of a particular level and more general levels to be collected, and prevents lower-level logs from being collected. This threshold can be overridden by a per-module threshold. See ["Configuring Different Threshold Levels for Different Types of Data"](#) on page 10-19 for details.

- The AUTOSYNC parameter determines if logging configuration that is performed in the Identity System Console is written to the log configuration file:

```
<NameValPair ParamName="AUTOSYNC" Value="True" />
```

The following is a complete example of the opening compound list and the simple list that contains the global settings:

```
<CompoundList xmlns="http://www.oblix.com" ListName="logframework.xml.staging">
  <SimpleList>
    <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
    <NameValPair ParamName="AUTOSYNC" Value="True" />
  </SimpleList>
```

The Second Compound List and Log Handlers

After the simple list, and within the start and end tags for the initial compound list, you specify an additional compound list. This compound list contains the log-handler definitions. The start and end tags for this list are as follows:

```
<CompoundList xmlns="http://www.oblix.com" ListName="LOG_CONFIG">
  . . .
</CompoundList>
```

This compound list tag is configured as follows:

- In the start tag for the compound list, the `xmlns` parameter indicates the relevant XML name space.
- Also in the start tag, you specify the name of the list on the `ListName` parameter.

Typically, the name of this list is `LOG_CONFIG`.

Between the start and end tags for the compound list for the log-handler, you specify one or more `ValNameList` elements. Each `ValNameList` element contains the definition for a log-handler. Each instance of this element begins and ends as follows:

```
<ValNameList xmlns="http://www.oblix.com" ListName="Unique_Name">
  . . .
</ValNameList>
```

The `ValNameList` elements are configured as follows:

- The opening tag sets the relevant XML name space on the `xmlns` parameter.
- The opening tag also sets a name for the log-handler on the `ListName` parameter.

Within the opening and closing `ValNameList` tags, you configure the log-handler. A log-handler definition contains three mandatory `NameValPair` elements:

- The first mandatory `NameValPair` element defines the logging level for the log-handler.

This element contains the statement `ParamName="LOG_LEVEL"`, whose value is a reserved name in [Table 10-1](#), as follows:

```
<NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL" />
```

- The second mandatory `NameValPair` element defines the destination for log output.

This element contains a statement `ParamName="LOG_WRITER"`, whose value is a reserved name in [Table 10-3](#), as follows:

```
<NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
```

- The third mandatory NameValPair element toggles this log-handler on and off.

This element contains a statement ParamName="LOG_STATUS", with a value of On or Off, as follows:

```
<NameValPair ParamName="LOG_STATUS" Value="On" />
```

Finally, within the opening and closing ValNameList tags, if you specify FileLogWriter or MPFileLogWriter as the log writer, you can add none, some, or all of the following. See [Table 10-6](#) for details:

- A destination file name, as follows:

```
<NameValPair ParamName="FILE_NAME" Value="oblog.log" />
```

- A buffer size, as follows:

```
<NameValPair ParamName="BUFFER_SIZE" Value="65535" />
```

- A file size that determines when a new log file is generated, as follows:

```
<NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" />
```

- A time in minutes that determines the interval at which a new log file is generated, as follows:

```
<NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" />
```

The List for Per-Module Logging

After the end tag for the compound list that delimits the log-handlers, and before the end tag for the initial compound list, you can add per-module logging parameters. See ["Configuring Different Threshold Levels for Different Types of Data"](#) on page 10-19 for details.

About XML Element Order

When using XML, you can specify parallel elements in a list in any order as long as the elements remain intact and within the tags that originally bracketed them. For example, the lists in [Example 10-2](#) and [Example 10-3](#) are equivalent:

Example 10-2 Name/Value List

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
  <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
  <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
```

Example 10-3 Name/Value List

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
  <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
  <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
```

Similarly, within a given tag, the attributes (except for the tag name, which must always be the first element within the tag brackets) can be reordered, as long as they

remain intact and within the tag elements that originally bracketed them. The opening tags for a name-value list in [Example 10–4](#) and [Example 10–5](#) are equivalent:

Example 10–4 Opening tag for a Name/Value List

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
```

Example 10–5 Opening tag for a Name/Value List

```
<ValNameList ListName="LogError2Sys" xmlns="http://www.example.com">
```

About Activating and Suppressing Logging Levels

Several factors determine if logging is active for a particular log-handler. [Table 10–4](#) lists these factors:

Table 10–4 Factors that Determine Whether Logging Is Active

Factor	Importance	Description
LOG_THRESHOLD_LEVEL	Primary	This parameter sets a cutoff for logging. Any log level that is more detailed than the threshold is suppressed. See Table 10–1 for valid log levels. You override this parameter for a subset of items that can be logged using the MODULE_CONFIG parameter. See "Configuring Different Threshold Levels for Different Types of Data" on page 10-19 for details. For Identity Servers only, you can also configure the threshold in the Identity System Console. See "Configuring Logs in the Identity System Console" on page 10-17 for details.
MODULE_CONFIG	Primary	This sets a per-module override for the global logging threshold. See "Configuring Different Threshold Levels for Different Types of Data" on page 10-19 for details.
LOG_STATUS	Secondary	This parameter toggles logging on or off, as long as it is not overridden by the logging threshold or a module-specific override.
AUTOSYNC	Secondary	When this parameter is set to <code>True</code>, changes that you make to logging settings in the Identity System Console take effect immediately without a server restart and the changes are saved to the configuration file. When <code>AUTOSYNC</code> is <code>False</code>, the changes that you make in the Identity System Console take effect immediately but are not saved to the configuration file and are discarded after the server is restarted.
The physical position of a log handler	Secondary	See "About Log Handler Precedence" on page 10-13.

About Log Handler Precedence

You can configure up to three log-handler definitions for a single log level in a log configuration file. Three different log handlers are required if you want to send output for a particular log level to each of the three log writers described in [Table 10–3](#).

If you specify different LOG_STATUS settings in these log handlers, the setting in the log-handler definition closest to the physical end of the log configuration file sets the status for the other log-handler definitions of the same log level. For example, you can set LOG_STATUS to Off for the first two log handlers for the Error log level, but if LOG_STATUS is On for the third and final log handler in the configuration file, logging still occurs for all three handlers .

The LOG_STATUS settings are moot if that level is more fine-grained than the current LOG_THRESHOLD_LEVEL. In this case, logging cannot be activated at this level unless the threshold is overridden by a module-specific threshold. See ["Configuring Different Threshold Levels for Different Types of Data"](#) on page 10-19 for details.

Log Configuration Parameters

At minimum, each log-handler definition contains five parameters listed in [Table 10-5](#).

Table 10-5 Mandatory Log Configuration File Parameters

Parameter	Comment
xmlns	This parameter is specified in the opening ValNameList tag. It specifies the relevant XML namespace for the current list and is identical for all log-handler definitions in a given logging configuration file. Example: <code>http://www.example.com</code>
ListName	This parameter is specified in the opening ValNameList tag. Where possible, use the default names. When creating a new log-handler definition, select a memorable name that you will not confuse with other log handlers. Examples: <code>WarningsAndAboveToSyslog</code> sends Fatal, Error, and Warning messages to the system log file. <code>WarningsOnlyToFileLog128KBuffer</code> sends messages from just the Warning level to a 128KB buffer, and hence to a disk file. <code>TraceOnlyToMPRotateDaily</code> sends messages from just the Trace level to the multi-process file writer, which opens and closes the file each time it writes to disk. This file is replaced with a fresh (empty) file every day, regardless of the size of the file at the time of replacement.
LOG_LEVEL	This specifies a log level. See Table 10-1 for details. The default logging configuration file activates logging for three levels: Fatal, Error, and Warning.
LOG_WRITER	This specifies the destination for log output for this log-handler. See Table 10-3 for details. The default log configuration file sends output to both the system log and the log data file for the component doing the logging.
LOG_STATUS	This parameter turns the log handler on or off.

If you specify `FileLogWriter` or `MPFileLogWriter` as the value for the LOG_WRITER parameter, the four parameters in [Table 10-6](#) are relevant.

Table 10–6 Log Data File Configuration Parameters

Parameter	Description	Default
FILE_ NAME	Mandatory. Used only for the FileLogWriter or MPFileLogWriter. It is the name and location of the file where log data is written. You can prepend an absolute path to the file name to store it somewhere other than the default location, which is: <i>Component_install_dir\oblix\logs</i> Where <i>Component_install_dir</i> is the root installation directory for the component whose system events you are logging. When you create more than one log-handler definition that sends output to FileLogWriter or MPFileLogWriter, provide unique file names so that multiple handlers do not write to the same file. This caution does not apply to log handlers accessing the SysLogWriter.	oblog.log
BUFFER_SIZE	Optional. This is the size of the buffer, in bytes, for logged data as it is being written to the log file. If you set the buffer value to 0 or a negative number, the default value is used. If you want to write to the log file immediately, without buffering, set the value to a small number, for example, 5. Oracle recommends that you set a small buffer size in situations where there are system failures.	65535 (64KB)
MAX_ ROTATION_ SIZE	Optional. When the log file reaches this size (in bytes), a time stamp is appended to the file name, for example "oblog.log" becomes "oblog.log1081303126." New data is written to the file with the original name.	52428800 (512KB)
MAX_ ROTATION_ TIME	Optional. A time interval, in seconds, when the log file is renamed, whether or not it has reached the maximum rotation size. If the rotation time determines when the file is rotated, the numbers appended to the log files differ by the number of seconds in the rotation interval. For example, "oblog.log.1081389526" and "oblog.log.1081303126" differ by 84,600, which is the number of seconds in 24 hours. This is the rotation interval set in the log configuration file.	86400 (1 day, in seconds)

Settings in the Default Log Configuration File

As installed with each component, the log configuration file activates only the highest three levels (Fatal, Error, and Warning) and directs all log output to the system log.

On Windows, you can view the system log for the computer that hosts the component you are logging by navigating to My Computer, Manage, Event Viewer, Application. System event entries for the components being logged are interspersed among the system events for the operating system and applications other than Oracle Access Manager.

For Solaris and Linux environments, the location of the system log is recorded in a system configuration file whose particulars can vary from machine to machine. For the name and location of this system file or the system log, consult the owner of the computer that hosts the component whose system log you wish to examine.

[Example 10–6](#) shows the default log configuration file with comments removed to expose the file structure.

Example 10–6 A Default Log Configuration File Without Embedded Comments

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<CompoundList xmlns="http://www.example.com ListName="logframework.xml.staging">
  <SimpleList>
    <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
    <NameValPair ParamName="AUTOSYNC" Value="True" />
  </SimpleList>
  <CompoundList xmlns="http://www.example.com" ListName="LOG_CONFIG">
    <ValNameList xmlns="http://www.example.com" ListName="LogFatal2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
    <ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
    <ValNameList xmlns="http://www.example.com" ListName="LogWarning2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_WARNING" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
    <ValNameList xmlns="http://www.example.com" ListName="LogAll2File">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ALL" />
      <NameValPair ParamName="LOG_WRITER" Value="FileLogWriter" />
      <NameValPair ParamName="FILE_NAME" Value="oblog.log" />
      <NameValPair ParamName="BUFFER_SIZE" Value="65535" />
      <NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" />
      <NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
  </CompoundList>
</CompoundList>
```

Description of the Settings in the Default Log Configuration File

The default configuration file sends Fatal, Error, and Warning messages to both the system log and to a log data file named oblog.log.

The simple list near the top of the file sets the following parameters:

- It sets the LOG_THRESHOLD_LEVEL to Warning.
The threshold suppresses logging for levels that are more fine-grained than Warning. You can override this threshold. See ["Configuring Different Threshold Levels for Different Types of Data"](#) on page 10-19 for details.
- The simple list also sets the AUTOSYNC parameter to True.
If set to True, this setting enables you to save values that you set in the Identity System console to this configuration file so that they persist after you restart the Identity Server. This parameter applies only to Identity Servers.

The nested compound list contains four log-handler definitions:

- The first, named `LogFatal2Sys`, sets the logging level to `Fatal` and sets `LOG_STATUS` to `On`.

The threshold level is `Warning`, which is more fine-grained than `Fatal`, so this definition is in effect. The log output is written to the system log, as specified by the `LOG_WRITER` parameter.

- The `LogError2Sys` log-handler definition sends `Error` level messages to the system log.

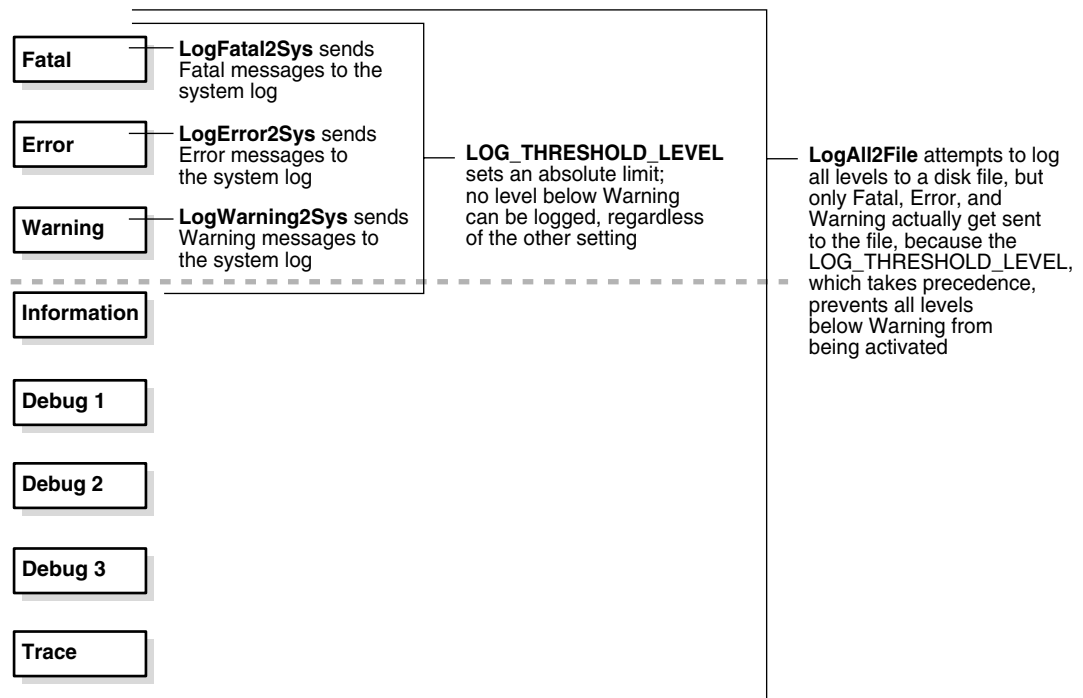
`Error` is located prior to the current threshold level (`Warning`), so this definition is in effect.

- The `LogWarning2Sys` definition sends `Warning` level output to the system log. Like the two previous log-handler definitions, it is not overridden by the current `LOG_THRESHOLD_LEVEL` parameter.

- `LogAll2File`, the final log-handler definition, appears to send output from all log levels to a disk file named `oblog.log`.

The `LOG_THRESHOLD_LEVEL` parameter is set to `Warning`, so only the output from the `Fatal`, `Error`, and `Warning` levels are recorded in this log data file. Since output from `LogAll2File` goes to the `FileLogWriter`, the parameters governing file name, buffer size, rotation size, and rotation interval all take effect.

Figure 10–1 Log-Level Activation in the Default Log Configuration File



Configuring Logs in the Identity System Console

For Identity Servers only, you can modify certain log settings through the Identity System Console. Alternatively, you can edit the log configuration file manually.

If you modify logging settings in the Identity System Console, the log configuration file for the associated component instance is saved without comments. To view the

original comments, open the following read-only duplicate of the original configuration file:

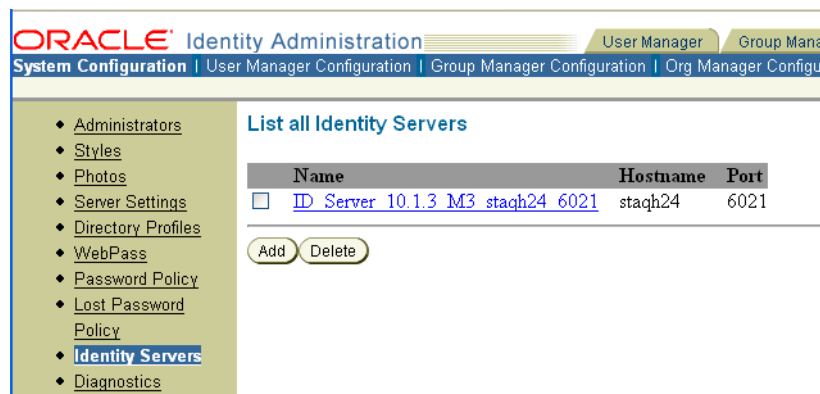
`Identity_Server_install_dir/oblix/config/oblog_config_original.xml`

Where `IdentityServer_install_dir` is the root installation folder for your Identity Server.

To view or modify log-handler definitions

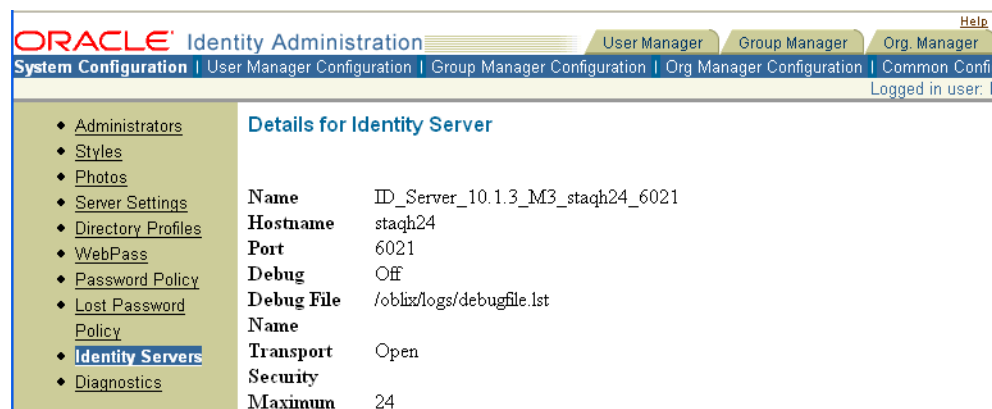
1. From the Identity System landing page, click the Identity System Console link.
If you are already logged in, click the Identity System Console tab.
2. From the Identity System Console, click the System Configuration sub-tab, then click the Identity Servers link in the left navigation pane.

The List of All Identity Servers page appears.



3. Click the link for the Identity Server whose activity you want to log.

The Details for Identity Server page appears with a list of log-handler definitions at the bottom of the page.



4. Examine the Log Threshold setting above the Log Handler Definitions table. This represents the current `LOG_THRESHOLD_LEVEL`.

To change this setting, click Modify at the bottom of the page and proceed to "To modify the log threshold from the Identity System Console" on page 10-19.

Otherwise, continue to the next step.

5. In the table of log-handler definitions, click the link for the log handler you wish to examine or change.

The Modify the Log Handler Definition page appears. From this page, you can specify values as described in [Table 10-5](#). If you specify File in the Output To field, you must complete the fields described in [Table 10-6](#).

You can change the defaults for the log file name, log file maximum size, log file rotation interval, and log buffer maximum size, as listed in [Table 10-6](#).

6. Click Save.

To modify the log threshold from the Identity System Console

1. From the Identity System Console, click the System Configuration sub-tab, then click Identity Servers in the left navigation pane.
2. Click the name of the Identity Server whose settings you want to examine.
3. Click Modify at the bottom of the Details of the Identity Server page.
4. Use the list to set the Log Threshold Level to the value you want.
5. Click Save.

The change takes effect immediately. If AUTOSYNC is True in the log configuration file, the change is written to the log configuration file so that the change persists after you restart the server.

To add or delete a log-handler definition from the Identity System Console

1. From the Identity System Console, click the System Configuration sub-tab, then click the Identity Server link in the left navigation pane.
2. Click the name of the Identity Server to which you wish to add a log-handler definition.
3. Click Modify at the bottom of the page.

The Modify Identity Server page appears.

4. Under Log Handlers Definition, complete the appropriate action:
 - To delete a log output configuration, check the box next to the appropriate link, then click Delete.
 - To add a log writer, click Add.

If you click Add, the Add a New Log Writer page appears.

5. Supply a name and a log level for the new log writer.
6. Verify that the log level is the same as or higher than the current log threshold level, as described in ["To view or modify log-handler definitions"](#) on page 10-18.

If the new log level is lower than the current threshold level, set the threshold level to the new log level or lower, as detailed in ["To modify the log threshold from the Identity System Console"](#) on page 10-19.

7. If you choose to output to a file rather than the system log, you must supply a file name and path, as described in [Table 10-6](#).
8. Click Save.

Configuring Different Threshold Levels for Different Types of Data

When diagnosing a problem, you may not want detailed logs for every operation that a component performs. For example, to diagnose slow response times for requests that

an Identity Server submits to its directory, you would want detailed information on LDAP operations and fewer details about other types of operations.

As of release 10.1.4.2, you can configure per-module or per-function threshold levels in the log configuration file, so that Oracle Access Manager generates detailed logs for some components while generating concise logs, or no logs, for others.

You configure per-module logging thresholds in a `MODULE_CONFIG` section in the `oblog_config.xml` file. The `MODULE_CONFIG` section overrides the global default that you specify on the `LOG_THRESHOLD_LEVEL` in the simple list section of this file.

The rest of this section discusses the following topics:

- [About the `MODULE\_CONFIG` Section](#)
- [Configuring a Log Level Threshold for a Function or Module](#)

About the `MODULE_CONFIG` Section

As described in ["Structure of the Log Configuration File"](#) on page 10-9, in the log configuration file you configure a global logging threshold. The following is an example of the global `LOG_THRESHOLD_LEVEL` setting:

```
<SimpleList>
  <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
  . . .
</SimpleList>
```

In addition to the global threshold, the configuration file can contain a `ValNameList` that defines function- or module-specific log thresholds. The name of this list is always `MODULE_CONFIG`. Only one instance of this list is permitted in the log configuration file, and the information in the list applies to all log writers defined in the file. As of release 10.1.4.2, the default log configuration file contains a commented sample of the `MODULE_CONFIG` list.

Each item in the `MODULE_CONFIG` list sets a logging level for a module, as shown in the following example:

```
<ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">
  <NameValPair ParamName="LDAP" Value="LOGLEVEL_TRACE"></NameValPair>
  <NameValPair ParamName="DB_RUNTIME" Value="LOGLEVEL_TRACE"></NameValPair>
</ValNameList>
```

The elements in this section are as follows:

- The `ValNameList` tag delimits the list of per-module logging thresholds.
- One `NameValPair` tag delimits each specific per-module logging threshold.
- The `ParamName` parameter sets the name of a module or function.
See [Table 10-7](#) for a list of valid values.
- The `Value` parameter sets the logging threshold for the module that you specify as a value for the `ParamName` parameter.

[Table 10-1](#) lists the permissible values for the `Value` parameter. In addition to these values, you can specify the value `ON` to enable logging for the module and a value of `OFF` to disable logging for the specific module.

Location of the Per-Module Logging Section in the Log Configuration File

You add the per-module logging threshold section near the end of the log configuration file, after the closing tag for the compound list for the log-handlers and before the closing tag for the first compound list in the file.

This section contains an example of the per-module logging section. See ["To configure a module-specific log threshold"](#) on page 10-24 for details.

List of Modules That Can Be Logged

[Table 10–7](#) describes the a partial list of the values that you can specify for the ParamName parameter in the MODULE_CONFIG list.

Table 10–7 ParamName Values You Can Configure for Per-Module Logging Threshold

ParamName Value	Logging Threshold That This Parameter Sets
AAA_ACTIONS	Sets a logging threshold for triggered actions that are configured as part of a policy in the Access System. <pre><ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG"> <NameValPair Paramname="AAA_ACTIONS" Value="OFF"> </NameValPair></pre> See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
AAA_AMENGINE	Sets a logging threshold for activity performed by the Access Manager engine. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
AAA_ISRESRCOPPROT	Sets a logging threshold for all Access System activities related to determining if a resource operation is protected. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
ACCESS_CLIENT	Sets a logging threshold for operations performed by an access client, that is, an AccessGate or WebGate. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
ACCESS_GATE	Sets a logging threshold for operations performed by an AccessGate. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
ACCESS_MGR	Sets a logging threshold for operations performed by the Policy Manager. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
ACCESS_SDK	Sets a logging threshold for operations performed by the Access Manager SDK interface. See the <i>Oracle Access Manager Developer Guide</i> for details.
ACCESS_SERVER	Sets a logging threshold for operations performed in the Access Server. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
AM_SDK	Sets a logging threshold for the Access Manager Manager SDK. See the <i>Oracle Access Manager Developer Guide</i> for details.

Table 10–7 (Cont.) ParamName Values You Can Configure for Per-Module Logging

ParamName Value	Logging Threshold That This Parameter Sets
AUDIT	Sets a logging threshold for auditing. See "Auditing" on page 11-1 for details.
AUTHENTICATION	Sets a logging threshold for user authentication operations. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
AUTHN_MGMT	Sets a logging threshold for authentication scheme management. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
AUTHN_PLUGIN	Sets a logging threshold for operations performed by an authentication plug-in. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
AUTHORIZATION	Sets a logging threshold for user authorization operations. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
AUTHZ_MGMT	Sets a logging threshold for authorization scheme management. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
AUTHZ_PLUGIN	Sets a logging threshold for authorization plug-in operations. See the <i>Oracle Access Manager Access System Administration Guide</i> for details.
CACHE	Sets a logging threshold for cache management and operations on the caches.
CONN_MGMT	Sets a logging threshold for connection management.
CONN_RUNTIME	Sets a logging threshold for connection runtime.
CONNECTIVITY	Sets a logging threshold for client-sever connectivity and messaging.
DB_CONFIGURATION	Sets a logging threshold for the data store interface layer configuration.
DB_RUNTIME	Sets a logging threshold for the data store interface layer runtime.
DIAGNOSTIC_FRAMEWORK	Sets a logging threshold for the diagnostic framework.
GROUPDB	Sets the threshold for logging accesses of Group Manager data in the directory.
GROUP_MGR	Sets the threshold for logging Group Manager operations.
HTTP_REQ	Sets the threshold for logging HTTP request processing.
IDXML	Sets the threshold for logging IDXML operations. See the <i>Oracle Access Manager Developer Guide</i> for details.
LDAP	Sets a logging threshold for LDAP SDK, for example: <pre><ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG"> <NameValPair Paramname="LDAP" Value="LOGLEVEL_TRACE"> </NameValPair></pre>

Table 10–7 (Cont.) ParamName Values You Can Configure for Per-Module Logging

ParamName Value	Logging Threshold That This Parameter Sets
LPM	Sets the threshold for logging Lost Password Management. See "Lost Password Management" on page 7-57 for details.
NET	Sets a logging threshold for network APIs.
OBMYGROUPS	Sets a logging threshold for ObMyGroups processing. This refers to searches of groups where the person who initiated the search is a member.
OIS_CLIENT	Sets a logging threshold for the Identity client.
OIS_ENGINE	Sets a logging threshold for the Identity Server engine.
OIS_MGMT	Sets a logging threshold for Identity System management.
OIS_SERVER	Sets a logging threshold for operations performed by an Identity Server.
ORG_MGR	Sets a logging threshold for operations performed in the Organization Manager.
POLICY_MGMT	Sets a logging threshold for policy and policy domain management.
PPP	Sets a logging threshold for Identity Event Plug-in API operations. See the <i>Oracle Access Manager Developer Guide</i> for details.
QUERY_BUILDER	Sets a logging threshold for Query Builder operations.
REQUEST_TIMINGS	Sets a logging threshold for the time consumed by all calls. See "Logging the Amount of Time to Process Requests" on page 10-25 for details.
SECURITY	Sets a logging threshold for the security and encryption library.
SELECTOR	Sets a logging threshold for Selector operations.
SERVER	Sets a logging threshold for server infrastructure.
SNMP	Sets a logging threshold for SNMP activity. See "SNMP Monitoring" on page 12-1 for details.
SSOTOKEN	Single sign-on token management.
USER_ADMIN	Sets a logging threshold for administrative operations in the User Manager.
UIDINFO_CACHE	Sets a logging threshold for the Uidinfo cache.
USER_MGR	Sets a logging threshold for operations performed in the User Manager.
USER_SVR	Sets a logging threshold for IDXML calls to the user service center (the User Manager).
USERDB	Sets a logging threshold for searches of user information in the directory.
UTILS	Sets a logging threshold for utility classes.
WEB	Sets a logging threshold for the Web server plug-in interface.
WORKFLOW_EXEC	Sets a logging threshold for workflow execution. See "Chaining Identity Functions Into Workflows" on page 5-1 for details.

Table 10–7 (Cont.) ParamName Values You Can Configure for Per-Module Logging

ParamName Value	Logging Threshold That This Parameter Sets
WORKFLOW_MGMT	Sets a logging threshold for workflow configuration and management. See "Chaining Identity Functions Into Workflows" on page 5-1 for details.
XML	Sets a logging threshold for the XML Infrastructure.

Configuring a Log Level Threshold for a Function or Module

The following procedure describes how to configure a function- or module-specific log level threshold.

Note: The following example shows a log configuration file with extended comments regarding per-module logging thresholds. If you have installed the 10.1.4.2 patch release, these comments do not appear in this file.

To configure a module-specific log threshold

1. Open the log configuration file in the following location:

Component_install_dir\identity\access\oblix\config

Where *Component_install_dir* is the directory where you installed the component for which you want to configure logging.

2. If a ValNameList section with a ListName of MODULE_CONFIG does not already exist in this file, create one that is similar to the following:

```
<ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">
</ValNameList>
```

Place this list after the end tag for the compound list that contains the log handler definitions. If there are comments immediately after this end tag, place the list after the comments.

3. Between the opening and closing tags of the new ValNameList element, configure one or more NameValPair elements.

This element contains a ParamName parameter and a Value parameter. See [Table 10–7](#) for the modules that you can supply on the ParamName parameter. See [Table 10–1](#) for values, or you can specify a value of On or Off. The following is an example:

```
<NameValPair ParamName="LDAP" Value="LOGLEVEL_TRACE"></NameValPair>
```

You can specify multiple ValNamePair elements within the ValNameList.

A complete per-module logging threshold section is illustrated in **bold** in the following example:

```
<!-- ===== -->
<!-- Configure the Log Level -->
. . .
<CompoundList xmlns="http://www.oblix.com" ListName="LOG_CONFIG">

  <!-- Write all FATAL logs to the system logger. -->
  <ValNameList xmlns="http://www.oblix.com" ListName="LogFatal2Sys">
```

```

        <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL">
    </NameValPair>
    <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter">
    </NameValPair>
    <NameValPair ParamName="LOG_STATUS" Value="On">
    </NameValPair>
</ValNameList>
. . .
</CompoundList>
<!-- List of values that can be specified in the module config -->
<!--
<!-- On - Uses loglevel set in the loglevel threshold -->
<!-- Off - No information is logged -->
<!-- LOGLEVEL_FATAL - serious error, possibly a program halt. -->
<!-- LOGLEVEL_ERROR - a transient or self-correcting problem. -->
<!-- LOGLEVEL_WARNING - a problem that does not cause an error. -->
<!-- LOGLEVEL_INFO - reports the current state of the component. -->
<!-- LOGLEVEL_DEBUG1 - basic debugging information. -->
<!-- LOGLEVEL_DEBUG2 - advanced debugging information. -->
<!-- LOGLEVEL_DEBUG3 - logs performance-sensitive code. -->
<!-- LOGLEVEL_TRACE - used when you need to trace the code path -->
<!-- execution or capture metrics. Includes all previous levels. -->
<!--
<!-- List of modules that can be specified in the module config -->
<!--
<!-- ALL_MODULES - Applies to all log modules -->
<!-- Specific module name - Applies to specific module -->
<!--
<!--
<!-- <ValNameList -->
<!-- xmlns="http://www.oblix.com" -->
<!-- ListName="MODULE_CONFIG"> -->
<!-- <NameValPair -->
<!-- ParamName="CONNECTIVITY" -->
<!-- Value="LOGLEVEL_TRACE"></NameValPair> -->
<!-- </ValNameList> -->

<ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">
    <NameValPair ParamName="LDAP" Value="LOGLEVEL_TRACE"></NameValPair>
    <NameValPair ParamName="DB_RUNTIME" Value="LOGLEVEL_TRACE">
    </NameValPair>
</ValNameList>

</CompoundList>

```

Logging the Amount of Time to Process Requests

For capacity planning and performance tuning purposes, you can log the time it takes to process calls to external components. For example, when performing capacity planning, you may want to know what calls to the directory server are taking the most time. After identifying these calls, you can ask your directory administrator to set up appropriate indexes. Or, you can analyze the client IP addresses to determine what components are handling the most calls. The logs can indicate if an Identity Server is receiving calls from multiple high-traffic WebPasses. If the calls from the IP address for internal users is much higher than the calls from the IP address for external users, you can add more Identity Servers for internal users.

When reviewing logs of request processing times, you can also determine what servers are implicated when more time is taken than expected. These logs capture the IP

address of the Web component that makes the request. Additionally, these logs show the size of various request queues, which can also indicate where there are performance bottlenecks.

The rest of this section discusses the following topics:

- [About the Calls Captured in the Request Timing Logs](#)
- [About Interpreting Request Times in the Logs](#)
- [Configuring the Log Files to Capture Time Data for External Requests](#)

About the Calls Captured in the Request Timing Logs

[Table 10–8](#) lists the calls that are captured in the request timing logs:

Table 10–8 List of Calls to External Components That Are Written to the Logs

Component	Calls That Are Logged
Identity Server and Access Server	■ LdapAddCalls ■ LdapDeleteCalls ■ LdapSearchCalls ■ LdapModifyCalls ■ LdapBindCalls ■ LdapGetDnCalls ■ LdapOtherCalls ■ AuditDB calls
Identity Server-specific	■ Identity Server-to-Identity Server cache flush calls ■ Identity Server-to-Access Server calls ■ Calls to external plug-ins (Identity, or PPP, plug-ins)
Access Server-specific	■ Authentication plug-in calls ■ Authorization plug-in calls

About Interpreting Request Times in the Logs

Each request timing log consists of four lines of output, as shown in [Table 10–9](#). Items of particular interest in this table include the following:

- In the third line, if you notice that a particular WebPass or WebGate is the requesting party most of the time, this can give you insight into where you may need to redistribute the load.
- The fourth line shows the time spent on different types of calls and the frequency of the call type.

The information on this line can indicate what types of requests are taking the longest.

[Table 10–9](#) illustrates an example of a REQUEST_TIMINGS entry in the LOGLEVEL_INFO log. Note that "OIS" is a frequently-used abbreviation for the Identity Server and "PPP" is a frequently-used abbreviation for an Identity System plug-in.

Table 10–9 Annotated Example of a REQUEST_TIMINGS Call in the Log Output

Line	Contents and Explanation
First line	The first line indicates the program that has initiated the request, as illustrated in the following example: <pre>2007/06/01@01:13:50.125000 5932 4892 REQUEST_TIMINGS INFO 0x0000093A obmessagechannel.cpp:410 "Request Received" _seqno^38 _opcode^12 _program^workflowSaveCreateProfile</pre>
Second line	The second line indicates that the request was taken from the request queue for processing. This line prints a log message and shows the wait time, which is the amount of time the request spent in the request queue before being processed. If you follow the MessageThreadID in the log output, you can view all of the operations that the thread processed for a particular request. After the thread has finished processing the request, it is assigned to a new request. <pre>2007/06/01@01:13:50.125000 5932 3980 REQUEST_TIMINGS INFO 0x0000093B servicethread.cpp:109 "Request started executing" Waiting time for the Request^0.000769 MessageThreadID^4892 _seqno^38 _opcode^12</pre>
Third line	The third line shows that the request is being processed, and it provides the IP address of the WebPass or WebGate that issued the request. This line prints a log message, and may display the IP address of the requesting client, as shown in the following example: <pre>2007/06/01@01:13:50.125000 5932 3980 REQUEST_TIMINGS INFO 0x0000093C data\module_main.cpp:614 "Request is being executed" _programName^workflowSaveCreateProfile _ ClientIP^111.11.11.111</pre>
Fourth line	The fourth line contains the frequency and time consumed by LDAP calls, audit calls, plug-in calls, and flush calls, as well as the queue size, as shown in the following example: <pre>2007/06/01@01:13:52.812000 5932 3980 REQUEST_TIMINGS INFO 0x0000093D servicethread.cpp:143 "Processing time for this request" queue size^0 MessageThreadID^4892 _seqno^38 _opcode^12 Total_Dura^2.687061 LdapAllCalls_Freq^55 LdapAllCalls_Dura^0.014618 LdapAddCalls_Freq^9 LdapAddCalls_ Dura^0.001843 LdapDeleteCalls_Freq^0 LdapDeleteCalls_ Dura^0.000000 LdapSearchCalls_Freq^28 LdapSearchCalls_ Dura^0.010169 LdapModifyCalls_Freq^4 LdapModifyCalls_ Dura^0.001000 LdapBindCalls_Freq^1 LdapBindCalls_Dura^0.000155 LdapGetDNCalls_Freq^13 LdapGetDNCalls_Dura^0.001452 LdapOtherCalls_Freq^0 LdapOtherCalls_Dura^0.000000 AuditDBCall_Freq^0 AuditDBCall_Dura^0.000000 PPPPluginCall_ Freq^0 PPPPluginCall_Dura^0.000000 OIS2OISflush_Freq^0 OIS2OISflush_Dura^0.000000 OIS2AAACALL_Freq^0 OIS2AAACALL_ Dura^0.000000</pre>

Configuring the Log Files to Capture Time Data for External Requests

You capture logs of request processing times on a per-module basis. To do this, you add the REQUEST_TIMINGS parameter to the MODULE_CONFIG section of the log configuration file. The following procedure describes how to capture only request processing times in the logs.

Note: The following example shows a log configuration file with extended comments regarding per-module logging thresholds. If you have installed the 10.1.4.2 patch release, these comments do not appear in this file.

To log the time taken to process external requests

1. Open the following file:

installdir\config\oblog_config.xml

Where *installdir* is the directory where you installed the component, for example, the Identity Server.

2. If a `ValNameList` section with a `ListName` of `MODULE_CONFIG` does not already exist in this file, create one that is similar to the following:

```
<ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">
</ValNameList>
```

Place this list after the end tag for the compound list that contains the log handler definitions. If there are comments immediately after this end tag, place the list after the comments.

3. To restrict the information written to the logs to only the time consumed by different types of calls, in the `MODULE_CONFIG` section, add a listing that contains the `REQUEST_TIMINGS` parameter, as illustrated in **bold** in the following example.

See ["Configuring Different Threshold Levels for Different Types of Data"](#) on page 10-19 for details on the `MODULE_CONFIG` section.

```
<!-- ===== -->
<!-- Configure the Log Level -->
. . .
<CompoundList xmlns="http://www.oblix.com" ListName="LOG_CONFIG">

<!-- Write all FATAL logs to the system logger. -->
<ValNameList xmlns="http://www.oblix.com" ListName="LogFatal2Sys">
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL">
  </NameValPair>
  <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter">
  </NameValPair>
  <NameValPair ParamName="LOG_STATUS" Value="On">
  </NameValPair>
</ValNameList>
. . .
</CompoundList>
<!-- List of values that can be specified in the module config -->
<!-- -->
<!-- On - Uses loglevel set in the loglevel threshold -->
<!-- Off - No information is logged -->
<!-- LOGLEVEL_FATAL - serious error, possibly a program halt. -->
<!-- LOGLEVEL_ERROR - a transient or self-correcting problem. -->
<!-- LOGLEVEL_WARNING - a problem that does not cause an error. -->
<!-- LOGLEVEL_INFO - reports the current state of the component. -->
<!-- LOGLEVEL_DEBUG1 - basic debugging information. -->
<!-- LOGLEVEL_DEBUG2 - advanced debugging information. -->
<!-- LOGLEVEL_DEBUG3 - logs performance-sensitive code. -->
<!-- LOGLEVEL_TRACE - used when you need to trace the code path -->
<!-- execution or capture metrics. Includes all previous levels. -->
```



```

<!-- -->
<!-- List of modules that can be specified in the module config -->
<!-- -->
<!-- ALL_MODULES - Applies to all log modules -->
<!-- Specific module name - Applies to specific module -->
<!-- -->
<!-- -->
<!-- <ValNameList -->
<!-- xmlns="http://www.oblix.com" -->
<!-- ListName="MODULE_CONFIG"> -->
<!-- <NameValPair -->
<!-- ParamName="CONNECTIVITY" -->
<!-- Value="LOGLEVEL_TRACE"></NameValPair> -->
<!-- </ValNameList> -->
<!-- Write only REQUEST_TIMINGS logs when when using LOGLEVEL_INFO. -->
<ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">
  <NameValPair ParamName="REQUEST_TIMINGS" Value="LOGLEVEL_INFO">
</ValNameList>

</CompoundList>

```


This chapter focuses on the auditing features and how to configure these using the Identity System console. The following topics are provided:

- [About Auditing](#)
- [Audit Output Considerations](#)
- [Controlling Audit Output](#)
- [Auditing Requirements](#)
- [Audit-to-Database Architecture](#)
- [Setting Up File-Based Auditing](#)
- [Setting Up Database Auditing](#)
- [Setting up Audit Reports](#)

Note: For details about installing audit-to-database components, see the *Oracle Access Manager Installation Guide*.

About Auditing

The auditing feature collects and presents data pertaining to policy and profile settings, system events, and usage patterns. Oracle Access Manager can generate two types of audit reports:

- **Static:** These reports are derived from policy and profile information that is stored on the Oracle Access Manager directory server.
See "[Static Audit Reports](#)" on page 11-3 for details.
- **Dynamic:** These reports are derived from Access System and Identity System events that are collected from the servers in your system.

At the most detailed level, dynamic audit reports reveal when a system event was triggered and who triggered it. At a higher level, these reports can reveal component load levels, resource request patterns, system intrusion attempts, and overall system performance. See "[Static Audit Reports](#)" on page 11-3 for details.

In addition to auditing, Oracle Access Manager supports logging, SNMP monitoring, and other reporting features. See [Chapter 9, "Reporting"](#) on page 9-1 for details.

Audit Output Considerations

You can record all dynamic audit reports and some static audit reports to disk file, to a relational database, or both. Some static reports can also be displayed in limited form through the graphical user interface.

Audit Security Considerations

Database auditing provides the following advantages over file based auditing in the area of security:

- All audit information is stored in a central database that can be protected by any security methods that your database supports.

The audit-to-file option records data to a plain-text file on each server that collects audit data. Such files are not protected by database-level security.

WARNING: To take full advantage of database security, make sure you turn off the audit-to-file feature for all Access and Identity Servers in your system. You should also store the password to the default audit database user account in the RDBMS profile on the directory server rather than in the ODBC.ini file (if used) on each server host.

- Data can be sent to an audit database using the transport security methods supported by ODBC or OCI, as applicable to your database.
- Using the audit database, Crystal Reports can generate security-related statistics.
For instance, you can track the number of resource requests that were refused during a given interval or compile a list of users who are locked out of the system.
- Auditing-to-database can assist in compliance reporting for regulatory acts such as Sarbanes-Oxley, Gramm-Leach-Bliley, and HIPAA (the Health Information Privacy and Accountability Act of 1996).

Audit Performance Considerations

Auditing, whether to database or file, can slow the performance of your Oracle Access Manager System. You can control the impact of auditing as follows:

- Turn on auditing only for selected servers.
See ["To enable and configure auditing for each Identity Server"](#) on page 11-36 and ["To enable and configure auditing for each Access Server"](#) on page 11-41 for details.
- Turn on auditing only for selected profile attributes, events, and Identity System applications.
See ["To specify global Identity System events and profile attributes for audit"](#) on page 11-38 and ["To specify User, Group, or Organization Manager events for audit"](#) on page 11-39 for details.
- Increase the database audit retry interval so that whenever the connection to the database is broken, the server does not initiate thrashing by resending a failed write attempt before the connection is restored.

You control this interval with the DBAuditRetryInterval parameter in the globalparams.xml file in the following directory:

Component_install_dir\apps\common\bin

where *Component_install_dir* is the installation directory of the server whose audit behavior you want to control. This parameter takes as a value the number of seconds to wait before initiating another attempt to write data to the audit database.

- For file-based auditing only, increase the size of the audit buffer.

This measure reduces the number of times the audit feature accesses your hard disk. See ["To enable and configure auditing for each Identity Server"](#) on page 11-36 and ["To enable and configure auditing for each Access Server"](#) on page 11-41 for details.

- For file-based auditing only, lengthen the interval between buffer flushes.

This reduces the number of times the auditing feature writes to disk, but it also increases the potential amount data you lose in a system failure.

WARNING: Only fatal errors are flushed to file if a server fails. All other audit items in the buffer at the moment of the failure are lost. Therefore, by increasing the buffer size or lengthening the interval between buffer flushes, you increase the potential volume of audit data lost in the event of a system failure.

Static Audit Reports

Static audit reports are generated from policy and profile information stored on the Oracle Access Manager directory server. You can generate five types of static reports:

Table 11–1 Static Audit Report Types

Report Type	Description
User Access Privilege Report	A global list of resources that users or groups of users can access at a specified point in time. They are also referred to as filtered profile queries. See "To create and manage user access privilege reports" on page 11-43 for details.
Resource Access Privilege Report	A global list of users who are authorized to access a specified resource or group of resources at a specified point in time. They are also referred to as filtered policy queries. See the procedure "To create and manage user access privilege reports" on page 11-43 for details.
Access Test	A limited, on-screen display that verifies whether a specified user or group of users can access a specified resource at a specified point in time. You cannot test for access to randomly defined groups of resources the way you can with the preceding two types of filtered queries.
Access System Diagnostic Report	An on-screen table of status information for some or all of the Access Servers in your system. This includes details about the directory components to which the Access Servers are connected. See Table 11–2 on page 11-5 for details.
Identity System Diagnostic Report	An on-screen table of status information for some or all of the Identity Servers in your system. This includes details about the directory components to which the Identity Server(s) are connected. See Table 11–2 on page 11-5 for details.

Dynamic Audit Reports

To be able to send data to the audit database, you must install and configure one of the following databases on a host within your domain:

- Microsoft SQL Server for environments where the Oracle Access Manager servers all run on Windows.
- Oracle Database for environments where the computer hosting the Oracle Access Manager server contains either an Oracle database server or an Oracle database client that is configured to talk to an Oracle database server.

The Oracle database client that resides on the computer hosting the Oracle Access Manager server can run on a different platform from the Oracle database server. For example, the Oracle Access Manager server and Oracle database client can run on a supported Unix host and the Oracle database server can be on a Windows host.

In addition, you can install and configure Crystal Reports presentation software on a Windows machine in your Oracle Access Manager domain. See ["To install Crystal Reports"](#) on page 11-45 for details.

Controlling Audit Output

You can control the type and amount of audit data collected by each server. For example, you can configure the Master Audit Rule on an Access Server to record authentication failures, but not authentication successes. See ["To modify audit output formatting for the Access System"](#) on page 11-42 for details. Or, you can configure the Application Auditing Policy on an Identity Server to record the time and date of each user login, but not the time of logout or session expiration. See ["To modify audit output formatting for the Identity System"](#) on page 11-37 for details.

If you send data to the audit database, you can display the collected information in Crystal Reports templates that have been preconfigured to present audit data. The generated audit reports fall into the following categories:

- Global View Access
- Authentication
- Authorization
- Activity
- Identity Management

See ["About Audit Reports"](#) on page 11-15 for details.

About Audit Options

You set all audit options through configuration pages in Oracle Access Manager, as detailed in [Table 11-2](#):

Table 11–2 Where to Set Audit Options

Audit-Related Functionality	Location in GUI	Scope
Enable file-based auditing and database auditing, and modify audit file attributes on an individual Identity Server.	Identity System Console, System Configuration, Identity Server, <i>ServerName</i> , Modify where <i>ServerName</i> specifies the Identity Server you want to modify	Per server
Modify the default formatting used for file-based and database auditing, including date format, date separator, message format, escape character, record separator, and field separator. To enable database auditing, you must replace the default message format string. See "To modify audit output formatting for the Identity System" on page 11-37. If you modify any other attributes, you may have to reconfigure your Crystal Report templates and repository settings.	Identity System Console, Common Configuration, Master Audit Policy, Modify	Global for file-based and database auditing in the Identity System
Specify the Identity System events to be audited. This includes successes and failures for login and logout, and password management.	Identity System Console, Common Configuration, Global Auditing Policies, Modify	Global for file-based and database auditing and for all applications in the Identity System
Create or modify RDBMS profiles and associated database instances. (These are necessary only for database auditing.)	Identity System Console, System Configuration, Directory Profiles, Configure RDBMS Profiles, Modify or Access System Console, System Configuration, Server Settings, Configure RDBMS Profiles, Modify	Global for database auditing only
Specify the Identity Servers to be included in the on-screen diagnostics display. Note: To ensure that the Diagnostics page displays the current status of a given Server, exercise the connection to that server by attempting a login or a user search before accessing the Diagnostics display.	Identity System Console, System Configuration, Diagnostics	Global (for the Identity System only) or for a server

Table 11–2 (Cont.) Where to Set Audit Options

Audit-Related Functionality	Location in GUI	Scope
Activate the collection of audit success and audit failure data for the following events: Search, View Profile, Modify Profile, View Location, Modify Location, Substitute Right, Workflow, Configuration, Deactivated User, Reactivated User, Created User, Deleted User, and Workflow Duration.	Identity System Console, User Manager Configuration, Audit Policies, Modify	Global (for User Manager reports only)
Activate the collection of success and failure data for the following events: Search, View Profile, Modify Profile, View My Group, View Group Member, Expand Group, Subscribe Group, Workflow, Configuration, and Workflow Duration.	Identity System Console, Group Manager Configuration, Audit Policies, Modify	Global (for Group Manager reports only)
Activate the collection success and failure data for the following events: Search, View Profile, Modify Profile, Containment Profile, Container Limit, View Location, Modify Location, Workflow, Configuration, and Workflow Duration.	Identity System Console, Org. Manager Configuration, Audit Policies	Global (for Organization Manager reports only)
Enable file-based and or database auditing and modify audit file attributes on an individual Access Server.	Access System Console, Access System Configuration, Access Server Configuration, <i>ServerName</i> , Modify where <i>ServerName</i> specifies the Access Server you want to modify.	Per server
Create or modify RDBMS profiles and associated database instances. (These are necessary only for database auditing.)	Access System Console, System Configuration, Server Settings, Configure RDBMS Profiles, Create (or Modify) or Identity System Console, System Configuration, Directory Profiles, Configure RDBMS Profiles	Global (for both file-based and database auditing)

Table 11-2 (Cont.) Where to Set Audit Options

Audit-Related Functionality	Location in GUI	Scope
Create or modify a master audit rule, which covers the following: audit events (success and failure of authentications and authorizations), audit event mapping, date format, escape character, audit record format, and cache formatting. To enable database auditing, you must replace the default audit record format string. See "To modify audit output formatting for the Access System" on page 11-42 for details. If you modify any other attributes, you may have to reconfigure your Crystal Report templates and repository settings.	Access System Console, Access System Configuration, Common Information Configuration, Master Audit Rule, Modify	Global (for both file-based and database auditing within the Access System)
You can specify the Access Server(s) to be included in the on-screen diagnostics display. Note: To ensure that the Diagnostics page displays the current status of a given Oracle Access Manager Server, exercise the connection to that server by attempting a login or a user search before accessing the Diagnostics display.	Access System Console, System Management, Diagnostics	Global (for the Access system only) or for a server
Create, modify and manage Global User Access Privilege Reports.	Access System Console, System Management, Manage Reports, Add or Modify	For a server

Auditing Requirements

Displaying audit reports on-screen or sending audit output to disk files does not require the installation of special components.

Auditing to a database is restricted to certain Oracle Access Manager system configurations and requires the installation of special components, as detailed in the following sections.

The rest of this section discusses the following topics:

- [Requirements for Auditing](#)
- [Requirements for Auditing to a Database](#)

Requirements for Auditing

The following are requirements when configuring auditing:

- For auditing with Oracle Access Manager 10.1.4.0.1, verify that you can reach the target database using the database client's SQLplus.

- For auditing with Oracle Access Manager 10.1.4.0.1, ensure that the ORACLE_HOME environment variable is set appropriately.
- For auditing with Oracle Access Manager 10.1.4.0.1, apply the most recent hot fixes and patches for the release before activating file auditing and database auditing at the same time.
- For Oracle Access Manager 10.1.4.0.1 and later versions, you have a full range of operating system choices if Oracle OCI is used for database auditing.

Requirements for Auditing to a Database

Database auditing requires special components, as outlined in the next section. Installation details are available in the *Oracle Access Manager Installation Guide*.

Special Components for Database Auditing

To enable auditing to a database, you must install various components listed in [Table 11-3](#):

Table 11-3 Special Components Needed for Database Auditing

Platform	Installation Notes
Oracle Access Manager 10.1.4.0.1 on Windows and supported Unix platforms with the Oracle Call Interface (OCI)	The following are supported options: ■ Host 1: Oracle Access Manager Identity Server or Access Server and an Oracle Database server 10.1.0.5. ■ Host 1: Oracle Access Manager Identity Server or Access Server and an Oracle Database client 10.1.0.5. Use the full client, not the instant database client. The client version must be 10.1.0.5. Host 2: Oracle Database server 9.2.0.7 or 10.1.0.5 or 10.2.0.2. ■ Host 1: Oracle Access Manager Identity Server or Access Server and Oracle Application Server 10.1.0.5. Host 2: Oracle Database server 9.2.0.7 or 10.1.0.5 or 10.2.0.2. To obtain the latest platform support information, see "Updates to Supported Versions and Platforms" on page 11-9.
Oracle Access Manager 10.1.4.2 on Windows and supported Unix platforms with the Oracle Call Interface (OCI)	The following are supported options: ■ Database client: No Database client is needed on the Oracle Access Manager server host. The Oracle Instant Client binaries are bundled with the Oracle Access Manager servers. ■ Database server: The Database server version must be 9.2.0.7 or 10.1.0.5 or 10.2.0.2. Notes: To obtain the latest platform support information, see "Updates to Supported Versions and Platforms" on page 11-9.

Table 11–3 (Cont.) Special Components Needed for Database Auditing

Platform	Installation Notes
Oracle Access Manager 10.1.4.0.1 or 10.1.4.2 on Windows with ODBC	Requirements on a Windows platform with ODBC for the Oracle Database server: ■ The computer hosting the Oracle Access Manager Identity Server or Access Server must have the ODBC driver for Oracle. ■ The Database server must be version 9.2.0.7 or 10.1.0.5 or 10.2.0.2. Requirements on a Windows platform with ODBC for MS SQL Server: ■ The Oracle Access Manager Identity Server or Access Server must have the ODBC driver for SQL Server. The Database server must have Microsoft SQL Server 2000, Standard, Enterprise, or Developer. If you are using Oracle Access Manager 10.1.4.2, Microsoft SQL Server 2005 is also supported. Notes: When using SQL Server, all servers that are connected to the audit database run on Windows hosts. If you use SQL Server, you do not have to install additional database drivers. To obtain the latest platform support information, see "Updates to Supported Versions and Platforms" on page 11-9.
Crystal Reports	You install Crystal Reports plus a required patch on a Windows machine that can access the ODBC database. See "Setting up Audit Reports" on page 11-44 for details. The Crystal Reports host must run Windows The following Crystal Reports packages have been tested: Crystal Reports 9.22a, Advanced Edition, patch = CR90DBEXWIN_EN_200403

Updates to Supported Versions and Platforms

To see the latest supported versions and platforms for this integration refer to Metalink, as follows.

To view information on Metalink

1. Go to the following URL:
<http://metalink.oracle.com>
2. Click the Certify tab.
3. Click View Certifications by Product.
4. Select the Application Server option and click Submit.
5. Choose Oracle Application Server and click Submit.

Audit-to-Database Architecture

[Figure 11–1](#) and [Figure 11–2](#) show a distributed environment with Oracle Access Manager servers on one or more host machines, the OCI or ODBC-compatible database server on another host, and the Crystal Reports application on yet another host.

In a simpler deployment, you can install your entire Oracle Access Manager system and all database auditing components on one Windows computer. In a single-host scenario, if you install Oracle database with ODBC, you need only one table of ODBC data-source definitions (one ODBC.ini file) on your host.

Figure 11–1 and Figure 11–2 illustrate the components that you install and configure to enable auditing to database.

Figure 11–1 Audit-to-Database Architecture: SQL Server

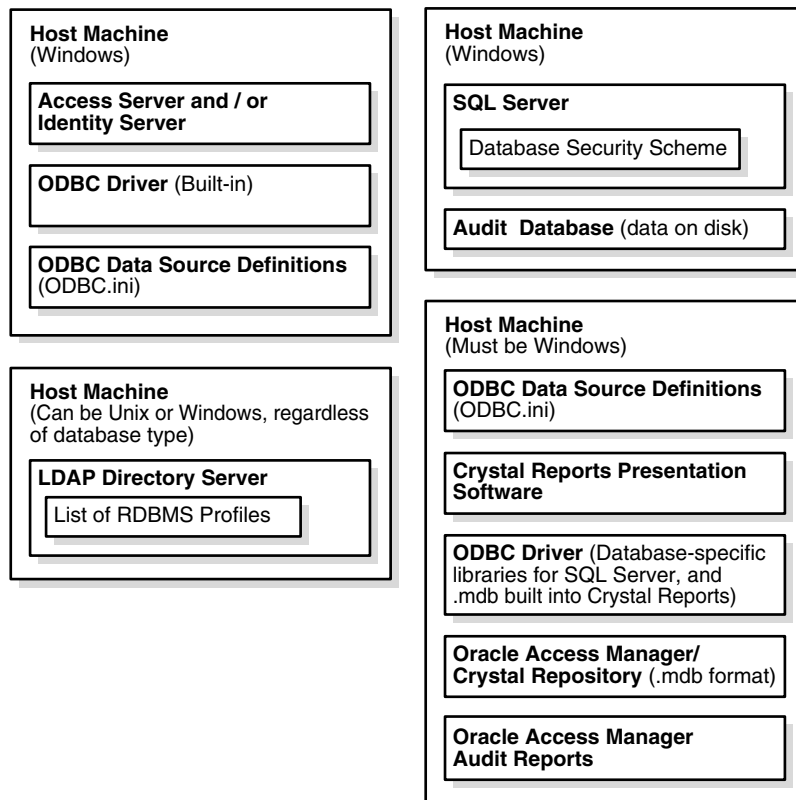


Figure 11–1 shows an audit-to-database architecture on SQL Server that is distributed across four host machines. In this configuration, all Access and Identity Servers must run on Windows, and the database must run on Windows. The LDAP Server does not have to be Windows-based.

Figure 11–2 illustrates the following architecture for auditing using the Oracle Database:

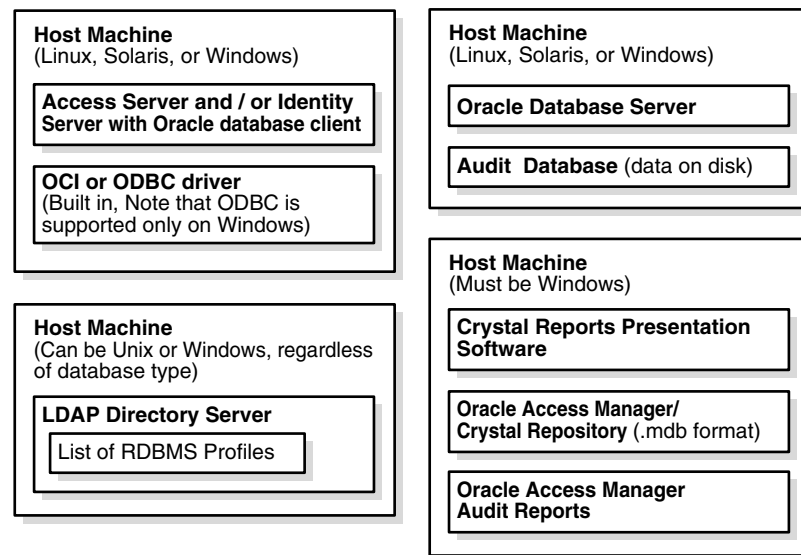
- The Identity or Access Server can reside on Windows or on a supported Unix platforms for auditing.
On Windows, ODBC and OCI connection types are supported for the audit database.
On a Unix platform, only an OCI connection type is supported.
- On a computer that hosts a version 10.1.4.0.1 Identity or Access Server, if you use an OCI connection type, the host must contain an ORACLE_HOME version 10.1.0.5 installation area.

The 10.1.0.5 ORACLE_HOME can be an Oracle Database client installation or an Oracle Database server installation.

- On a computer that hosts a version 10.1.4.2 Identity or Access Server, if you use an OCI connection type, the host does not require an ORACLE_HOME installation area.
- If you use an ODBC connection type on Windows, the computers that host the Identity and Access Servers do not require an ORACLE_HOME of exactly version 10.1.0.5.

The Oracle Database server can reside on a different computer, and it can be version 9.2.0.7, 10.1.0.5, or 10.2.0.2.

Figure 11–2 Audit-to-Database Architecture: Oracle Database



About OCI Settings

The following sections describe configuration requirements if you are auditing to an Oracle database that uses an OCI connection type:

- [OCI Configuration for Oracle Access Manager 10.1.4.0.1](#)
- [OCI Configuration for Oracle Access Manager 10.1.4.2](#)

OCI Configuration for Oracle Access Manager 10.1.4.0.1

For Oracle Access Manager 10.1.4.0.1, if you are auditing to an Oracle database that uses an OCI connection type, the host computers for the Identity Servers and the Access Servers must contain one of the following:

- Oracle Database server 10.1.0.5
- Oracle Database client 10.1.0.5

If any host for an Identity Server or Access Server does not already have an Oracle Database client or server, you must install one.

OCI Configuration for Oracle Access Manager 10.1.4.2

For Oracle Access Manager 10.1.4.2, the OCI Instant Client binaries are installed on the Identity and Access Servers. As a result, you are not required to have an Oracle home directory that contains the Oracle Database server or client.

For Oracle Access Manager 10.1.4.2, you must set the TNS_ADMIN environment variable to the directory that contains the following file:

Oracle_database_server/NETWORK/ADMIN/tnsnames.ora

If the audit database server resides on a different host computer than the Identity and Access Servers, copy the file tnsnames.ora to a directory on the Identity or Access Server host and set the TNS_ADMIN environment variable to the new directory where tnsnames.ora resides.

Finally, for the OCI Instant Client 10.1.0.5 on a supported Unix platform, the configuration file tnsnames.ora has a limitation. All entries in this file must fit on one line. A complete entry is similar to the following:

```
OCIDB2=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host)(PORT=1521))(CONNECT_
DATA=SERVICE_ NAME=OCIDB2)))
```

If an entry extends over more than one line, Oracle Access Manager servers may fail when resolving the service name that is used to connect to the Oracle Database server.

About ODBC Data Source Definitions

ODBC data-source definitions encapsulate all the information necessary for a client application such as an Oracle Access Manager server or Crystal Reports to connect with ODBC 3.0-compatible databases formatted for SQL Server or Microsoft Access (.MCPO).

ODBC data-source definitions are stored in a file named ODBC.ini on each Windows computer that hosts an application connected to the audit database. Only one such list of ODBC data-source definitions should exist on a given machine, and that file should be shared by all the applications that connect to the audit database.

- Users generally add or modify data source definitions through a Windows administration GUI.
Because this GUI hides many configuration details, users may never become aware that ODBC.ini exists, much less learn its location.
- If you are auditing to an Oracle database that uses an ODBC connection type, users may also need to set the environment variable ORACLE_HOME. This is the case if the host machine for the Oracle Access Manager component being audited contains an Oracle database client that is configured to talk to an Oracle database server on a different machine. If the Oracle Database server resides on the same machine as the component being audited, users do not have to set ORACLE_HOME.

The following table lists the most important attributes in a data source definition:

Table 11–4 Key Attributes in an ODBC Data-Source Definition

Attribute	Description
DSN (Data Source Name)	Identifies a unique data source definition to all the clients that access a given data source. (The term DSN is often used incorrectly to denote an entire ODBC data source definition.) A DSN must be unique within your Oracle Access Manager environment. Furthermore, all the ODBC.ini files and RDBMS profiles referencing a particular DSN must contain identical information related to that DSN, including login name, password, database, and so on.
User	Identifies the database user account authorized to access and modify the ODBC data source. When an Oracle Access Manager server or the Crystal Reports application needs to access the data source, it uses this account to supply credentials to the database security scheme. For SQL Server, the default user account is "sa," which stands for system administrator.
Password	This is the password associated with the account specified by User Name. You specify this password in the default user account for the audit database and again in either the RDBMS profile or the ODBC data source definition in the ODBC.ini file on each Oracle Access Manager server connected to the audit database. If you specify a password in both the ODBC data source definition and the RDBMS profile, you should know that the former stores the password string on each Oracle Access Manager host in unencrypted form in ODBC.ini, which is a plain text file, while the latter stores the string in encrypted form on just the Oracle Access Manager LDAP directory server.
Database	This is the name of the target data source, which, in the case of the audit-to-database feature, is one of the following: ■ The name of the database containing the Oracle Access Manager audit data ■ A Microsoft Access database (.mdb file) for the Crystal Repository containing .gif image files and SQL-compatible queries used by the Crystal Report templates preconfigured to present audit information
Server	This the name of the machine on which the RDBMS server (SQL Server) resides.
Port	This is the port on which the RDBMS server listens for incoming requests.
Driver	The fully qualified path to the ODBC driver libraries on the local machine.
Description	Details to help you identify the data source definition.

About ODBC Drivers

An ODBC driver library is specific to the type of database server to which you are connecting and the platform on which the driver is installed.

Each ODBC driver provides libraries that facilitate connection to the audit database.

An ODBC driver must exist on each machine hosting an Oracle Access Manager server that connects to the audit database. When both an Access Server and a Identity Server reside on the same machine, only a single ODBC driver is required for that host.

About the Windows ODBC Driver

By default, Windows installs the ODBC driver for SQL Server in the \Windows\System32 directory. It is accessible through the ODBC Data Source Administrator, which you launch by navigating to Start, Programs, Administrative Tools, Data Sources (ODBC).

The About tab in the ODBC Data Source Administrator displays the driver version number. If, for any reason, the installed version is lower than 3.5, or the driver is damaged or missing, you can download a replacement from the following Web site:

<http://www.microsoft.com/odbc>

The self-installing file is named `odbc35in.exe`.

About RDBMS Profiles for Database Auditing

An RDBMS profile is a definition for an audit database where all Identity and Access servers send audit data. RDBMS profiles can be defined for primary and secondary database instances for use in the event of failover.

RDBMS profiles reside on the Oracle Access Manager directory server, where they are accessed by all the Access and Identity Servers that are connected to that directory server. You configure RDBMS profiles in the Access System Console or the Identity System Console. See ["To create an RDBMS profile"](#) on page 11-33 for details.

Generally, reporting (static reports) and auditing (dynamic reports) share a single RDBMS profile. All the Access and Identity Servers that use a particular feature (such as reporting and auditing) must use the same RDBMS profile.

LDAP database profiles are server- and operation-specific. They can be shared by Access and Identity Servers, but they do not need to be. Two or more Access or Identity Servers can each use a different LDAP database profile even though each LDAP database profile is set up for the same LDAP server and operation.

About Profiles For Databases That Use an ODBC Connection Type

Each RDBMS profile can be configured for an ODBC or an OCI connection type. The RDBMS profile contains a database instance definition that configures the connection between an Access or Identity Server and the audit database. For an ODBC connection type, the database instance includes the DSN (Data Source Name) for the ODBC data source definition that is used to connect to the audit database. It also includes a copy of the attributes listed in [Table 11-4](#).

The same DSN appears in the ODBC.ini file of every machine that hosts an Access or Identity Server that is connected to the audit database. Details associated with the DSN stored in the RDBMS profile server must match exactly the details associated with every instance of that DSN in the ODBC.ini files throughout your Oracle Access Manager system.

If the associated attributes fail to match, the values for USER and PASSWORD recorded in the RDBMS profile take precedence over the corresponding values stored in ODBC.ini. On the other hand, the values for DATABASE and other attributes stored in ODBC.ini take precedence over the corresponding values in the RDBMS profile. The values in one location are never overwritten by the values stored in the other location.

About Profiles For Databases That Use an OCI Connection Type

Each RDBMS profile can be configured for an ODBC or an OCI connection type. The RDBMS profile contains a database instance definition that configures the connection between an Access or Identity Server and the audit database. For an OCI connection

type, you specify a Global Database Name (GDN) in the database instance definition. The database instance also includes a copy of the attributes listed in [Table 11-4](#).

About the Audit Database

The audit database collects data from all the Access Servers and Identity Servers in your system. Oracle Access Manager supports the following:

- ODBC 3.0-compliant databases on SQL Server, which runs on the Windows platform
- ODBC 3.0- and OCI-compliant Oracle Database running on Windows and UNIX-based computers

About the Crystal Repository

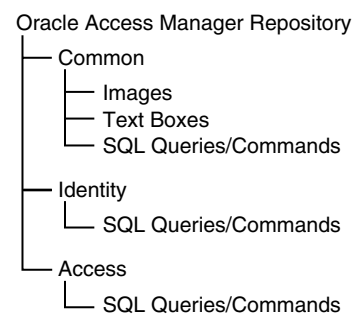
Within the context of the auditing to a database, the Oracle Repository and the Crystal Repository are synonymous because you link them through the orMap.ini file. See the procedure ["To edit orMap.ini"](#) on page 11-46.

This repository is a Microsoft Access format (.mdb) database that contains the following resources:

- .gif files used in Audit Reports
- SQL queries and commands used in Audit Reports
- Custom functions
- Templates that give Audit Reports a consistent look and feel
- Sample reports

[Figure 11-3](#) shows the organization of common, identity, and access resources in the repository.

Figure 11-3 Organization of resources in the repository



About Audit Reports

[Table 11-4](#) describes the audit reports:

Table 11-5 Content Types in the Audit Reports

Audit Data Type	Audit Report Type	Description
Authentication Statistics	Authentication/Dynamic	The number of authentication successes and failures that occurred on a given server or across the Oracle Access Manager system during a given interval.

Table 11–5 (Cont.) Content Types in the Audit Reports

Audit Data Type	Audit Report Type	Description
Authorization Statistics	Authorization/Dynamic	The number of authorization successes and failures that occurred on a given server or across the Oracle Access Manager system during a given interval.
Access Failures by User	Authorization/Activity/Dynamic	The number of authorization requests from a given user that failed during a given interval.
Access Failures by Resource	Authorization/Activity/Dynamic	The number of authorization requests for a given resource that failed during a given interval.
Access Privileges	Filtered Query/Static	Two types of access privilege reports are supported: ■ All the users allowed to access a list containing one or more resources. ■ All the resources accessible by a list containing one or more users. When this information is recorded to a file or database, it is referred to as a User Access Privilege Report or advanced Filtered Profile Query. See the procedure "To create and manage user access privilege reports" on page 11-43 for details. When simpler queries are displayed through the GUI, they are referred to as Access Tester output. This type of audit information is static. It is derived from policy information that is stored on the directory server rather than collected on a historical, event-by-event basis from an Access Server or Identity Server.
User Profile History	Identity Management/Dynamic	Changes to password, policy, profile, and so on for all users.
Group History	Identity Management/Dynamic	A list of groups that a user has been added to or removed from in a given interval.
Revoked Users	Identity Management/Dynamic	A list of users who have been locked out of the system.
Deactivated Users	Identity Management/Dynamic	A list of users whose access accounts have been deactivated. Lists of reactivated users can also be generated.
Password Changes	Identity Management/Dynamic	The number of passwords that have been changed throughout the system during a given interval.
User Status Changes	Identity Management/Dynamic	The groups to which a given user or users has been added within a given interval.
Identity History	Identity Management/Dynamic	Changes to password, policy, profile, and so on for one or more individual users.
Workflow Execution Time	Identity Management/Dynamic	The average and maximum length of time it has taken to complete a workflow during a given period.

Setting Up File-Based Auditing

You turn file-based auditing on and off as well as change the name and location of the audit file generated by an individual Access Server or Identity Server through the Oracle Access Manager GUI. By default, the audit flag is off for all Oracle Access Manager Servers. The following two procedures detail the steps for activating and configuring file-based auditing for Identity Servers and Access Servers, respectively.

Note that you must activate the auditing flag for each Oracle Access Manager server individually.

You can also modify the defaults for the following three categories of audit settings:

- **Common:** See ["To specify global Identity System events and profile attributes for audit"](#) on page 11-38 for details.
- **Oracle Access Manager server-specific:** See the procedures ["To enable and configure auditing for each Identity Server"](#) on page 11-36 and ["To enable and configure auditing for each Access Server"](#) on page 11-41 for details.
- **Identity System application-specific:** See the procedure ["To specify User, Group, or Organization Manager events for audit"](#) on page 11-39 for details.

The categories in the preceding list apply to both file-based and database auditing.

To configure file-based auditing for an Identity Server

1. From the Identity System landing page, click Identity System Console.
If you are already logged in to the Identity System, click the Identity System Console tab.
2. Click the System Configuration sub-tab, then click Identity Servers in the left navigation pane.
3. From the list of identity servers, select the link for the server that you want to modify.
4. Review the Audit to File settings.

ORACLE Identity Administration		User Manager	Group Manager	Org. Mana
System Configuration User Manager Configuration Group Manager Configuration Org Manager Configuration Common				
Logged in u				
- Administrators - Styles - Photos - Server Settings - Directory Profiles - WebPass - Password Policy - Lost Password Policy Identity Servers - Diagnostics	Modify Identity Server Name ID_Server_10.1.3_M3_staqh24_6021 Hostname* staqh24 Port* 6021 Debug* ☒ Off ☐ On Debug File Name* /oblix/logs/debugfile.lst Transport Security* ☒ Open ☐ Simple ☐ Cert Maximum Session Time (hours)* 24 Number of Threads* 20 Audit to Database Flag (auditing on/off) ☒ Off ☐ On Audit to File Flag (auditing on/off) ☒ Off ☐ On Audit File Name Audit File Maximum Size (bytes) 100000 Audit File Rotation Interval (seconds) 7200 Audit Buffer Maximum Size (bytes) 25000			

Table 11-6 describes the audit-to-file configuration parameters.

Table 11–6 Audit-to-File Configuration Parameters

Parameter	Description	Default
Audit to File Flag	The radio buttons turn the audit to file feature to On or Off	Off
Audit File Name	You can specify the absolute path and name of the audit file for the Access or Identity Server you are auditing. You may find it convenient to specify something similar to the following: <i>Component_install_dir</i>\oblix\log\auditfile.lst where <i>Component_install_dir</i> is the root installation directory for the associated Access or Identity Server.	[blank]
Audit File Maximum Size	The approximate size, in bytes, at which the existing audit file is closed and renamed to the following <i>AuditFileName.lst</i> <i>TimeStamp</i> where <i>AuditFileName</i> is the name of the audit file, and <i>TimeStamp</i> is a numerical representation, in seconds since midnight, January 1, 1971, of the moment when the file was created. By default, <i>AuditFileName</i> is <i>AuditFile</i>. Simultaneously, a new audit file named <i>AuditFileName</i> is created and opened for input.	100000
Audit File Rotation Interval	How often, in seconds, the audit file is renamed and a new one created to replace it. Time-based rotation occurs regardless of the current size of the audit file. See the previous row in this table for details.	7200
Audit Buffer Maximum Size	The amount of audit data, in bytes, that can be accumulated in a buffer before the entire buffer is written to disk.	[blank]
Audit Buffer Flush Interval	The number of seconds after which the content of the audit buffer is written to the audit file regardless of the amount of data in the buffer.	7200

To configure file-based auditing for an Access Server

- From the landing page for the Access System, click the link for the Access System Console.

If you are already logged in to the Access System and are working in the Policy Manager, click the Access System Console link at the top of the page.
- Click the Access System Configuration tab, then click Access Server Configuration in the left navigation pane.
- From the list in the Access Server Configuration page, select the Access Server you want to modify.
- In the Details for Access Server page, examine the audit file settings.

If you wish to change any of them, click the Modify button at the bottom of the page.
- In the Modify Access Server page, modify the Audit File parameters.

ORACLE Access Administration Access Manager Help

System Configuration System Management Access System Cor

Logged in user: M

- Access Server Clusters
- AccessGate Configuration
- Add New Access Gate
- Access Server Configuration**
- Authentication Management
- Authorization Management
- User Access Configuration
- Common Information Configuration
- Host Identifiers

Modify Access Server

Name: dummy_Access_Server

Hostname*: dummy

Port*: 65001

Debug*: ☒ Off ☐ On

Debug File Name*:

Transport Security*: ☒ Open ☐ Simple ☐ Cert

Maximum Client Session Time (hours)*: 24

Number of Threads*: 60

Access Management Service*: ☒ Off ☐ On

Audit to Database (on/off)*: ☒ Off ☐ On

Audit to File (on/off)*: ☒ Off ☐ On

Audit File Name:

Setting Up Database Auditing

The following are high-level tasks for setting up database auditing:

Task overview: Enabling database auditing

1. Set up and verify your Oracle Access Manager environment.
See ["Setting Up Your System for Database Auditing"](#) on page 11-19 for details.
2. Install and configure your RDBMS application (SQL Server or the Oracle database), then create and configure the Oracle Access Manager audit database.
See ["Setting Up Your System for Database Auditing"](#) on page 11-19 for details.
3. Configure Oracle Access Manager for database auditing.
For an OCI connection type, you create an RDBMS profile. For an ODBC connection type, this involves enabling your Oracle Access Manager servers to connect to the audit database by creating ODBC data source definitions and an RDBMS profile. You also need to configure and verify both your Identity and Access systems for auditing.
See ["Configuring Auditing"](#) on page 11-35 for details.
4. Install and configure Crystal Reports, then verify that the Oracle Access Manager audit templates can display audit database information.
See ["Setting up Audit Reports"](#) on page 11-44 for details.

Setting Up Your System for Database Auditing

The Oracle Access Manager audit database is an ODBC 3.0 compliant database running on SQL Server or the Oracle Database, or an OCI-compliant Oracle Database.

Task overview: Preparing for the audit database

1. If you are installing SQL Server, read ["About installing SQL Server \(Windows\)"](#) on page 11-20.

2. Create and configure the Oracle Access Manager audit database on the database server.

See the procedures ["SQL Server on Windows: To create the audit database"](#) on page 11-21, ["Oracle Database on Windows: To create the audit database"](#) on page 11-21, or ["Oracle Database on a supported Unix platform: To create the audit database"](#) on page 11-22.

When creating the Oracle database, specify the Unicode character set (AL32UTF8).

The SQL Server installation uses the Unicode character set UCS-2 by default.

Select UTF-8 as the national character set.

3. Upload the auditing and reporting schema to the auditing database.

See ["Task overview: Uploading the audit schema"](#) on page 11-23.

4. Create an ODBC data source definition (System DSN) on each Oracle Access Manager server that will send data to the audit database.

See the procedures ["To create an ODBC data source definition \(Windows\)"](#) on page 11-30.

5. Create an RDBMS profile on the Oracle Access Manager LDAP directory server so that each Access or Identity Server that is connected to the directory server can recognize the ODBC data source definition on its host machine.

See ["To create an RDBMS profile"](#) on page 11-33.

6. Restart all your Oracle Access Manager servers.

See ["To make the RDBMS profile visible \(Windows\)"](#) on page 11-35 or ["To make the RDBMS profile visible \(supported Unix platforms\)"](#) on page 11-35.

About installing SQL Server (Windows)

You can use the Standard, Enterprise, or Developer Edition of SQL Server 2000.

If you plan to implement other Oracle Access Manager features that use SQL Server (for example, the SharePoint Portal Server integration), the auditing feature can share a single SQL Server installation with the these other features, provided that SQL installation meets the minimum requirements dictated by each feature.

Follow the instructions supplied by Microsoft to install SQL Server. The installation wizard prompts you to specify setup options. In most cases, you should accept the defaults as you progress through the wizard pages, but first check the following table and enter any settings that differ from the defaults:

Table 11-7 Special Settings for SQL Server Installation

Wizard Page Setting	What to Specify
autorun.exe opening screen	SQL Server 2000 Components, Install Database Server
Installation target	"Local Computer"
Installation option	"Create a new instance of SQL Server"
Type of installation	"Server and Client Tools"
Instance name	"Default"
Type of setup	"Typical"
Services accounts	"Use the same account for each service. Auto Start SQL User Service"

Table 11-7 (Cont.) Special Settings for SQL Server Installation

Wizard Page Setting	What to Specify
Service settings	"Use Local System account" The default login name, which is also referred to as the Login ID or User Name, is "sa," and the password can be blank if the box labeled "blank password" is checked. The password can be whatever you wish if "blank password" is not checked. In any case, record the login name and associated password so that you can duplicate them exactly when you create your RDBMS profile and the ODBC data source definitions on each Oracle Access Manager server host.
Authentication mode	"Mixed Mode"

After you have installed SQL Server, proceed to ["SQL Server on Windows: To create the audit database"](#) on page 11-21.

Creating the Audit Database

The procedure for creating the Oracle Access Manager audit database differs depending on whether you are using SQL Server or the Oracle Database.

SQL Server on Windows: To create the audit database

1. On the machine hosting SQL Server, navigate to:

`My Computer, Manage, Services and Applications, Microsoft SQL Servers, hostname`

where *hostname* is the Windows Services name for the machine hosting SQL Server.

2. In the left pane of the Computer Management window, right-click Databases in the branch beneath the host name of the machine on which SQL Server is installed, then click New Database.
3. Select a descriptive name for the database, then click OK.
An icon representing the new database appears in the right hand pane of the Computer Management window.
4. Proceed to ["Uploading the Audit Schema"](#) on page 11-23.

Oracle Database on Windows: To create the audit database

1. Start the Oracle Database server.
2. Start the Database Configuration Assistant by clicking Start, Programs, Oracle - OraDb10g_home1, Configuration and Migration tools, Database configuration assistant.
3. When the wizard prompts you for a Global Database Name, record the name and use it in the database instance definition of the RDBMS profile in Oracle Access Manager.
4. In the Initialization Parameters screen, choose AL32UTF8 as the database character set, and choose UTF8 as the national character set.
5. Proceed to ["Uploading the Audit Schema"](#).

Oracle Database on a supported Unix platform: To create the audit database

1. Start the Oracle Database server located in the following directory:
`ORACLE_HOME/bin/dbca`
2. Start the Database Configuration Assistant.
3. When the wizard prompts you for a Global Database Name, record the name and use it in the database instance definition of the RDBMS profile in Oracle Access Manager.
4. In the Initialization Parameters screen, choose AL32UTF8 as the database character set, and choose UTF8 as the national character set.
5. Tune the audit schema.
See ["Tuning the Audit Schema"](#) on page 11-22 for details.
6. Upload the audit schema.
See ["Uploading the Audit Schema"](#) on page 11-23 for details.

Tuning the Audit Schema

If you perform auditing using the Oracle database with OCI, you must use the default schema. You cannot use the information in this section.

For MS SQL Server and the Oracle database using an ODBC connection type, you can optionally tune the audit schema before or after uploading it. In this section, tuning refers to increasing or decreasing the maximum length of audit record fields of type ncharvar in MS SQL Server or ncharvar2 in the Oracle database.

The audit schema path for MS SQL Server is as follows:

OracleAccessManager_Install_dir\oblix\reports\crystal\audit.sql

By default, the values of all nvarchar fields for the Identity and Access Server audit records in MS SQL Server are truncated to the first 170 characters. For MS SQL Server 2000, the total size of a record cannot exceed a limit of 8060 bytes. Each nvarchar occupies 2 bytes of memory. The maximum size of each nvarchar field with default schema is 340 bytes. The maximum size of the default audit record is 7836 bytes, equivalent to $(340 * 23) + 8 + (2 * 2) + 4$.

The audit schema path for the Oracle database is as follows:

OracleAccessManager_Install_dir\oblix\reports\crystal\audit_oracle.sql

By default, the values of all nvarchar2 fields for the Identity and Access Server audit records in the Oracle database are truncated to the first 255 characters.

You can increase or decrease the column lengths of the following fields:

- adminDN
- applicationInfo
- applicationName
- exceptionString
- objects
- reason
- sourceUserDN

- targetUserDN
- url
- userProfileAttrs
- webGateContextInfo

For example, in an Oracle database using an ODBC connection type, if the default maximum length of the exceptionString field is insufficient, you can increase it, for example, to a maximum of 400 characters.

You can modify the audit schema before uploading it, or if you have previously uploaded the audit schema and it contains audit records, you can modify the existing schema.

To modify the default length of fields

1. Open the following file in a text editor:

`component_installdir/apps/common/bin/globalparams.xml`

Where `component_installdir` is the installation directory for the Access or Identity Server.

2. In the `globalparams.xml` file for both the Access Server and the Identity Server, set the value of the `DBAuditTruncateDataToCollLength` parameter to "true".

See the parameter reference appendix in the *Oracle Access Manager Customization Guide* for details.

3. If the schema has been uploaded to the database, you can also use the alter command to modify the length for an nvarchar field in the database.

The following is an example for the MS SQL Server database:

```
alter table oblix_audit_events alter column exceptionString nvarchar(400);
```

The following is an example for the Oracle Server database:

```
alter table oblix_audit_events modify exceptionString nvarchar2(400);
```

4. If schema has not yet been uploaded to the database server, change the column lengths in the database before uploading it, using the instructions for the database.

Uploading the Audit Schema

The audit schema enables you to import audit data from the Oracle Access Manager servers and export that data to the Crystal Repository, where it is presented in Oracle Access Manager audit reports.

Task overview: Uploading the audit schema

1. Copy the Oracle Access Manager audit schema and supporting resources from an Oracle Access Manager server host to the Oracle Access Manager audit database host.

The copy procedure differs depending whether you are performing a Windows-to-Windows or Unix-to-Unix transfer. See:

["To copy the audit and reporting schema to the audit database host"](#) for details.

2. Upload the audit and reporting schema to your audit database and verify that the upload was successful, which differs depending on whether you are using SQL Server or the Oracle Database. See:

- [SQL Server on Windows: To upload the audit schema](#)
- [SQL Server on Windows: To verify the audit schema](#)
- [SQL Server on Windows: To upload and verify the access reporting schema](#)
- [Oracle Database on Windows or a UNIX-based Platform: To upload and verify the audit schema](#)
- [Oracle Database on Windows or a UNIX-based Platform: To upload and verify the access reporting schema](#)

To copy the audit and reporting schema to the audit database host

1. On any machine hosting a Oracle Access Manager server, locate the directory containing the Oracle Access Manager audit schema by navigating to:

Component_Install_dir\oblix\reports\crystal

where *Component_Install_dir* is the root installation directory of your Oracle Access Manager server (IdentityServer_install_dir\identity\, for example).

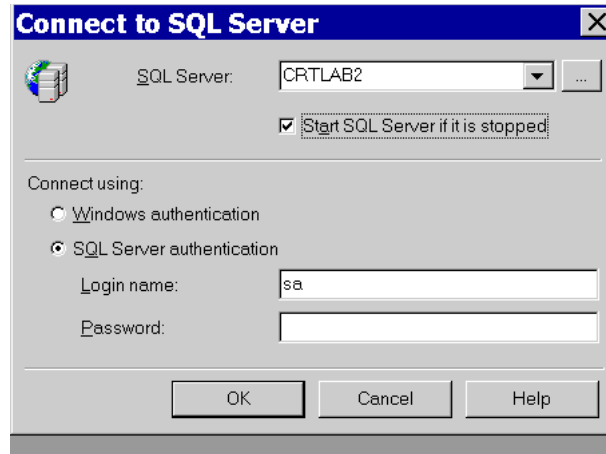
2. Using any of the means available for your particular operating system and network domain, copy the file audit.sql to a directory on the machine hosting your Oracle Access Manager auditing database.

This procedure isn't necessary if you installed your audit database on the same machine as one of your Oracle Access Manager servers.

3. Continue with the following procedures as is appropriate for the database application you are using:
 - [SQL Server on Windows: To upload the audit schema](#)
 - [SQL Server on Windows: To verify the audit schema](#)
 - [SQL Server on Windows: To upload and verify the access reporting schema](#)
 - ["Oracle Database on Windows or a UNIX-based Platform: To upload and verify the audit schema"](#)
 - ["Oracle Database on Windows or a UNIX-based Platform: To upload and verify the access reporting schema"](#)

SQL Server on Windows: To upload the audit schema

1. On the machine hosting SQL Server, navigate to:
Start, Programs, Microsoft SQL Server, Query Analyzer
2. If the "Connect to SQL Server" page is not already displayed in the SQL Query Analyzer window, navigate to:
File, Connect



3. In the Connect to SQL Server page, verify that the Windows Service name of your SQL Server host is displayed in the field labeled SQL Server.
4. Check "Start SQL Server if it is stopped."
5. Set "Connect using" to "SQL Server authentication."
6. Enter the login name and password you selected when installing SQL Server, then click OK.

A Query window will open in the SQL Query Analyzer window.

7. Launch the Oracle Access Manager audit database in the SQL Query Analyzer.

In the SQL Query Analyzer menu, navigate to:

File, Open

8. Navigate to "audit.sql" which is located under the directory you copied from your Oracle Access Manager server to your audit database host in the preceding procedure.

For details, see the procedure ["To copy the audit and reporting schema to the audit database host"](#) on page 11-24. The specific location of audit.sql is:

```
..\reports\crystal\audit.sql
```

9. In the Query window, add the following line to the very beginning of the file audit.sql:

```
use AuditDBName;
```

where *AuditDBName* specifies the Oracle Access Manager audit database you created in the procedure ["SQL Server on Windows: To create the audit database"](#) on page 11-21. In our example, we named the database NPAuditDB.

For all SQL statements, don't forget to place a semi-colon at the end of the line.

```

use NPAuditDB;

drop table oblix_audit_events;

create table oblix_audit_events (
    eventDateAndTime    dateTime not null,
    serverId             varchar(255) not null,
    webServerId          varchar(255),
    eventName            varchar(255) not null,
    workflowInstanceId   varchar(255),
    workflowType         varchar(255),
    workflowName        varchar(255),
    workflowAction      varchar(255),
);

```

10. Press F5 to execute the command. Alternatively, select Query, Execute from the SQL Query Analyzer menu.

The first time you do this, the application will return the following error message:

cannot drop the table 'oblix_audit_events', because it does not exist in the system catalog yet

This is both customary and logical, because the table did not exist when the "use" command was executed. If you save audit.sql and subsequently re-execute this command, the error message will not reappear, because the table now exists.

11. Minimize, but do not close the Query window; you will need to add another line to audit.sql when you verify that the schema have uploaded successfully. Proceed to: ["SQL Server on Windows: To verify the audit schema"](#) on page 11-26.

SQL Server on Windows: To verify the audit schema

1. Perform a "dummy" select from the oblix_audit_events table.

Add the following line immediately beneath the line you added to audit.sql in the procedure ["SQL Server on Windows: To upload the audit schema"](#) on page 11-28:

```
select * from oblix_audit_events;
```

Remember to include a semi-colon at the end of the line.

2. Click F5 to execute the command.

```

use NPAuditDB;
select * from oblix_audit_events;

drop table oblix_audit_events;

create table oblix_audit_events (
    eventDateAndTime    dateTime not null,
    serverId             varchar(255) not null,
    webServerId          varchar(255),
    eventName            varchar(255) not null,
    workflowInstanceId   varchar(255),
    workflowType         varchar(255),
);

```

Column headings such as eventDateAndTime appear in a pane immediately beneath the code pane in the Query window. These indicate that the audit.sql schema uploaded successfully.

3. In the SQL Query Analyzer window, click File, Save to record the changes to your audit.sql, which is now linked to your Oracle Access Manager audit database.
4. Proceed to ["SQL Server on Windows: To upload and verify the access reporting schema"](#) next.

The next procedure is similar to the previous procedures (where you uploaded and verified the audit schema using the oblix_audit_events table. In the following procedure, you copy table definitions and commands for oblix_rpt_as_reports, oblix_rpt_as_resource, and oblix_rpt_as_users from the audit.sql file into the Query Analyzer workspace, and execute them.

SQL Server on Windows: To upload and verify the access reporting schema

1. On the machine hosting SQL Server, perform the following activities as needed to login:
 - Navigate to Start, Programs, Microsoft SQL Server, Query Analyzer
 - If the "Connect to SQL Server" page is not already displayed in the SQL Query Analyzer window, navigate to: File, Connect.
 - In the Connect to SQL Server page, verify that the Windows Service name of your SQL Server host is displayed in the field labeled SQL Server.
 - Check "Start SQL Server if it is stopped."
 - Set "Connect using" to "SQL Server authentication."
 - Enter the login name and password you selected when installing SQL Server, then click OK.

A Query window will open in the SQL Query Analyzer window.

2. Launch the Oracle Access Manager audit database in the SQL Query Analyzer, as follows:
 - In the SQL Query Analyzer menu, navigate to File, Open
 - Navigate to the audit.sql file under the directory you copied from your Oracle Access Manager server to your audit database host earlier. For example:

IdentityServer_install_dir\identity\oblix\reports\crystal\audit.sql

3. In the Query window, add the following line to the very beginning of the file audit.sql:

```
use AuditDBName;
```

where *AuditDBName* specifies the Oracle Access Manager audit database you created in the procedure ["SQL Server on Windows: To create the audit database"](#) on page 11-21. In our example, we named the database NPAuditDB.

For all SQL statements, don't forget to place a semi-colon at the end of the line.

4. Add all drop and create commands for the three Oracle Access Manager tables (oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users) to the Query Analyzer window and execute these at one time, as follows:

- Copy the drop table oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users information (3 drop commands followed by 3 create table commands) together from the audit.sql file and paste these into the Query Analyzer window at once.

Note: Copy all at one time and do not change the order of these commands. There are dependencies between the tables.

- Press F5 to execute the commands at one time (or select Query, Execute from the SQL Query Analyzer menu).
 - Minimize, but do not close the Query window; you will need to add another line to audit.sql when you verify that the schema have uploaded successfully.
5. Verify information for the oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users tables as follows:
 - Perform a "dummy" select from the oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users tables (individually or all at once).
 - Add the following line immediately beneath the line you added from audit.sql in the procedure ["SQL Server on Windows: To upload the audit schema"](#) on page 11-28:


```
select * from oblix_rpt_as_reports;
select * from oblix_rpt_as_resources;
select * from oblix_rpt_as_users;
```
 - Click F5 to execute the command.
 6. Proceed to ["Enabling Access and Identity Servers to Connect to the Audit Database"](#).

The next two procedures are similar to the previous procedures. However, these are specific to the Oracle Database on Windows or a UNIX-based platform.

Oracle Database on Windows or a UNIX-based Platform: To upload and verify the audit schema

1. Start the Oracle Database server and the iSQL*Plus application.
2. Connect to the iSQL *Plus web application of Oracle DB server.

A typical URL for this is the following:

```
http://oracle_DB_host_name:port/isqlplus/
```

Where *Oracle_DB_host_name* is the name of the Oracle Database instance and *port* is the port number that you have chose during the installation of the Oracle Database server.

3. Log in to iSQL *Plus by providing the user name, password and GDN of the database.
4. Copy the schema definition from the following file to the iSQL*Plus Workspace page:


```
Identity_Server_install_dir\oblix\reports\crystal\audit_oracle.sql
```
5. Click the Execute button.
6. To verify the audit schema, enter the following command in iSQL*Plus:

```
desc oblix_audit_events
```

Alternatively, you can enter the `select *` command for `oblix_audit_events`; in iSQL*Plus.

7. Proceed to ["Oracle Database on Windows or a UNIX-based Platform: To upload and verify the access reporting schema"](#) next.

The next procedure is similar to the previous procedures (where you uploaded and verified the audit schema using the `oblix_audit_events` table. In the following procedure, you copy table definitions for `oblix_rpt_as_reports`, `oblix_rpt_as_resource`, and `oblix_rpt_as_users` from the `audit.oracle.sql` file into the iSQL *Plus web application of Oracle DB server, and execute them.

Oracle Database on Windows or a UNIX-based Platform: To upload and verify the access reporting schema

1. Login to the iSQL *Plus, as needed:
 - Start the Oracle Database server and the iSQL*Plus application
 - Connect to the iSQL *Plus web application of Oracle DB server.
 - Log in to iSQL *Plus by providing the user name, password and GDN of the database.
2. Add all drop and create commands for the three Oracle Access Manager tables (`oblix_rpt_as_reports`, `oblix_rpt_as_resources`, and `oblix_rpt_as_users`), and execute as follows:
 - Add all drop and create commands for the three Oracle Access Manager tables (`oblix_rpt_as_reports`, `oblix_rpt_as_resources`, and `oblix_rpt_as_users`) from the following file to the iSQL*Plus Workspace page:


```
Identity_Server_install_dir\oblix\reports\crystal\audit_oracle.sql
```
 - Click the Execute button
3. Verify the schema for all three Oracle Access Manager tables (`oblix_rpt_as_reports`, `oblix_rpt_as_resources`, and `oblix_rpt_as_users`) individually or all at once with the following command in iSQL*Plus:

```
desc oblix_rpt_as_reports;
desc oblix_rpt_as_resources;
desc oblix_rpt_as_users;
```

Alternatively, you can enter the `select *` command for the three Oracle Access Manager tables `oblix_rpt_as_reports`; `oblix_rpt_as_resources`; `oblix_rpt_as_users`; in iSQL*Plus.

4. Proceed to ["Enabling Access and Identity Servers to Connect to the Audit Database"](#) on page 11-29.

Enabling Access and Identity Servers to Connect to the Audit Database

You enable your servers to connect to the audit database by creating a RDBMS profile on the directory server and ODBC data source definitions on each machine hosting a server that connects to the audit database. A single, unique System DSN (System-wide Data Source Name) connects all of these objects.

It is extremely important that every attribute associated with a given DSN in both the RDBMS profile and the ODBC data source definitions on the server hosts match exactly. For details, see ["To create a primary RDBMS instance"](#) on page 11-33.

Task overview: Enabling Oracle Access Manager servers to connect to the audit database

1. All—Set the value of the SQLDBType parameter in globalparams.xml.
2. Windows—create an ODBC data source definition (System DSN) on each Oracle Access Manager Server host.
3. All—Using either the Identity System Console or the Access System Console, create an RDBMS profile on the directory server.

See ["Task overview: Setting up an RDBMS profile"](#) on page 11-32. This includes the following tasks:

- a. Create a primary RDBMS instance as described in ["To create a primary RDBMS instance"](#) on page 11-33.
- b. Create optional secondary RDBMS instances for your RDBMS profile as described in ["Task overview: To create a secondary RDBMS instance"](#) on page 11-34.
- c. Restart all Oracle Access Manager servers as described in ["To make the RDBMS profile visible \(Windows\)"](#) on page 11-35.

To set the SQLDBType parameter

1. Open the following file:

Component_install_dir/identity/apps/common/bin/globalparams.xml

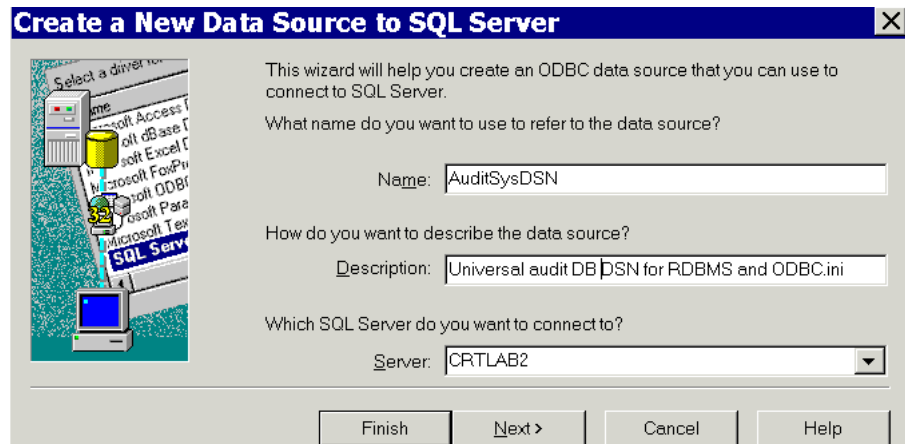
where *component_install_dir* is the location where the Access or Identity Server was installed.

2. Set the value of the SQLDBType parameter in globalparams.xml as follows:
Oracle: Indicates an Oracle Database that uses an ODBC connection type.
Oracle_OCI: Indicates an Oracle Database that uses an OCI connection type.
SQLServer: Indicates a SQL Server database.

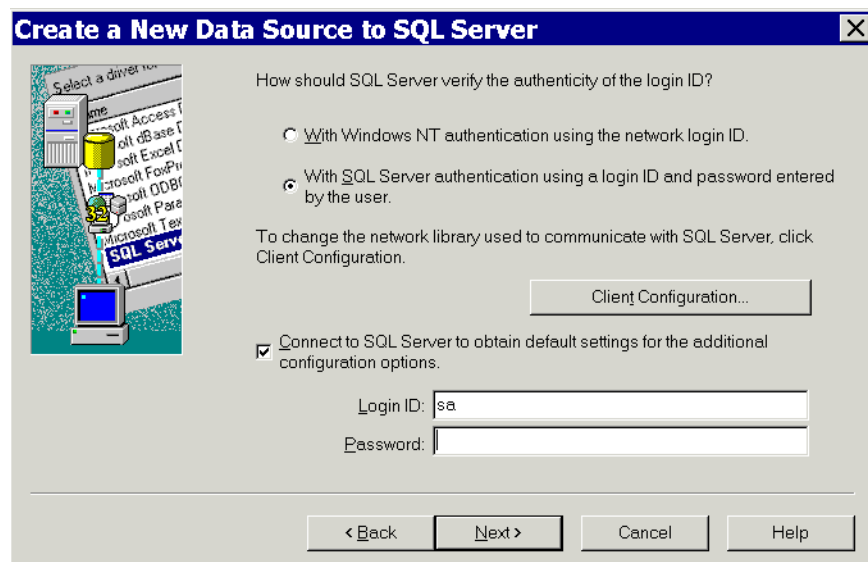
To create an ODBC data source definition (Windows)

1. On a Oracle Access Manager server host you wish to connect to the audit database, navigate to: Start, Settings, Control Panel, Administrative Tools, Data Sources (ODBC).
2. Click the System DSN tab.
3. Click Add.
4. From the list of database drivers, select SQL Server, then click Finish.
5. In the Name field, type a descriptive name.

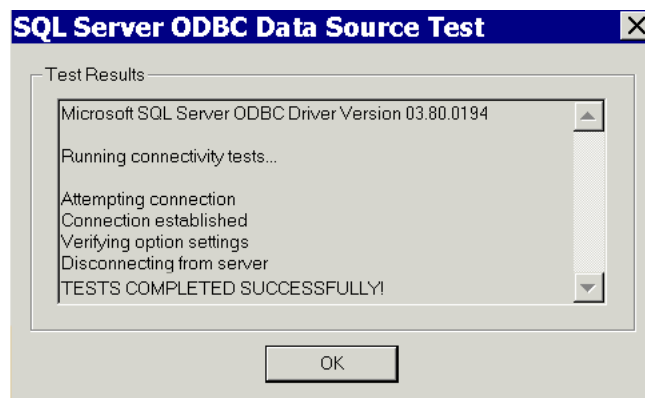
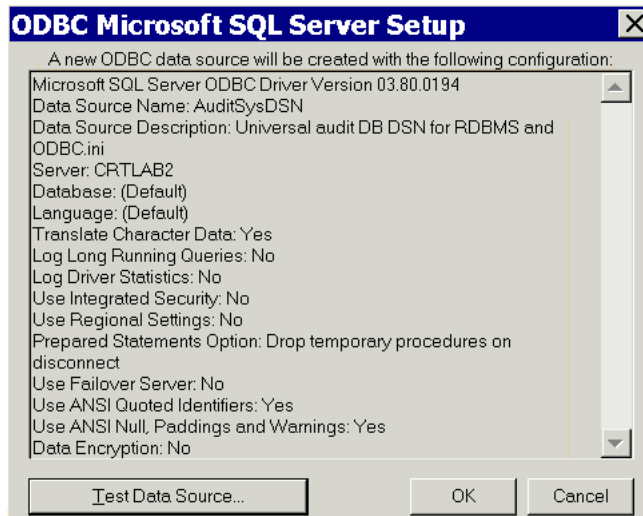
For instance, AuditSysDSN stands for the System DSN for the audit database. Write this name down, because you will have to use this exact character string for the ODBC data source definitions on every other Oracle Access Manager Server host, and for the primary RDBMS instance in your RDBMS profile as well.



6. In the Description field enter notes to help users identify this object.
7. In the Server field, select the Windows Services name of the host on which the Oracle Access Manager audit database is running, then click Next.
8. When the next page appears, select "With SQL server authentication. . ."



9. Verify that "Connect to SQL server to obtain . . ." is selected.
10. Type the Login ID and password you specified when you installed SQL Server.
11. Leaving the default settings on the next two pages untouched, click Next, then click Finish.
12. After a page appears listing the settings for the new ODBC data source definition, click Test Data Source.



13. After a page appears to announce success, click OK three times to dismiss the open pages.
14. Repeat this procedure on every Oracle Access Manager server host you wish to connect to the audit database.

Make sure you use the exact same settings in every case, and for your RDBMS database instances as well. Proceed to ["Task overview: Setting up an RDBMS profile"](#).

Task overview: Setting up an RDBMS profile

1. Create an RDBMS profile.
See ["To create an RDBMS profile"](#) on page 11-33 for details.
2. Create a primary RDBMS instance.
See the procedure ["To create a primary RDBMS instance"](#) on page 11-33 for details.
3. Create (optional) secondary RDBMS instances.
See ["Task overview: To create a secondary RDBMS instance"](#) on page 11-34 for details.
4. Make the RDBMS profile visible.

As appropriate for the database application you are using, See the procedure "[To make the RDBMS profile visible \(Windows\)](#)" on page 11-35 or "[To make the RDBMS profile visible \(supported Unix platforms\)](#)" on page 11-35.

To create an RDBMS profile

1. From the Identity System Console, click System Configuration, then click the link for Directory Profiles in the left navigation pane, then click the Add button in the Configure RDBMS Profiles section of the Configure Profiles page.

Alternatively, from the Access System Console, click System Configuration, then click Server Settings in the left navigation pane, then click the Add button in the Configure RDBMS Profiles section of the page.

The Create RDBMS Profile page is identical for the Identity System Console and the Access System Console.

ORACLE Identity Administration

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration | Help

Logged in user: |

Create RDBMS Profile

Name *

Database Connection Type * ☒ ODBC ☐ OCI

Used By * ☐ Reporting ☐ Auditing ☐ MHS

Name	Server Type
1	

Maximum Active Servers 1

Failover Threshold 1

Sleep For (Seconds) 60

Max. Session Time (Min.) 0

☒ Enable Profile

Note: The fields marked with an asterisk(*) are required fields

2. In the Name field, enter a descriptive name.
For instance AuditDBSysDSN refers to the System DSN created for the audit database. You are creating an RDBMS profile on this page, but this name provides a convenient universal name to identify matching sets of data source definition values in the RDBMS profile and the ODBC.ini files on each Oracle Access Manager server host.
The name of each RDBMS profile on a directory server must be unique.
3. In the Database Connection Type field, select the type of connection that your database uses.
4. In the Used By field, check the Reporting and Auditing options.
5. Verify that the Enable Profile box is selected.
6. Proceed to "[To create a primary RDBMS instance](#)" on page 11-33.

To create a primary RDBMS instance

1. Navigate to the Create RDBMS Profile page, as described in "[To create an RDBMS profile](#)" on page 11-33.

2. From the Create RDBMS Profile page, click the Add button next to the table labeled Database Instances.
3. In the Name field of the Create Database Instance page, enter a descriptive name.
For convenience, you can use the universal name you gave to the RDBMS Profile, such as AuditDBSysDSN.
4. The following field will either be DSN Name or GDN, depending on whether you specified an ODBC or an OCI connection type for the database.
For SQL Server, you can use the same name for the database instance and the RDBMS profile, for example AuditDBSysDSN. For the Oracle database, use the GDN that you specified when configuring the database. See ["Oracle Database on Windows: To create the audit database"](#) on page 11-21 or ["Oracle Database on a supported Unix platform: To create the audit database"](#) on page 11-22 for details.

WARNING: The character string you specify as the DSN for your RDBMS instance must match exactly the DSN you specify for the ODBC data source definition on each Oracle Access Manager server. Furthermore, the values for all other database instance attributes must be empty or match exactly the values for the corresponding attributes in the ODBC data source definitions throughout your Oracle Access Manager system.

5. In the Database field, specify the name of the audit database.
This example uses NBAuditDB.
6. In the User name field, enter the login name you specified when you created the audit database.
7. Enter the password associated with the audit database login name.
8. Leave the other fields at their default settings.
You can change them later, if necessary.
9. Click Save to commit the database instance settings you have entered.
10. When the Modify RDBMS Profile page appears, click Save to commit the RDBMS profile settings you have entered.
11. If you wish to create a secondary RDBMS instance, proceed to the task overview immediately following.
Otherwise, proceed to ["To make the RDBMS profile visible \(Windows\)"](#) on page 11-35 or ["To make the RDBMS profile visible \(supported Unix platforms\)"](#) on page 11-35.

Task overview: To create a secondary RDBMS instance

1. Perform all the steps in ["Creating the Audit Database"](#) on page 11-21.
For convenience, you may want to name the second instance of the audit database something like NPAuditDB_2.
2. Perform all the steps in ["Uploading the Audit Schema"](#) on page 11-23.
3. Perform steps 5 through 11 in ["To create an RDBMS profile"](#) on page 11-33.
For convenience, you may want to specify the name of the RDBMS instance and the DSN name as something like AuditDBSysDSN_2.

4. After the Modify RDBMS Profile page appears, verify that the Server Type for your secondary RDBMS instance is set to secondary.
5. Add the ODBC data source definitions for the secondary RDBMS instance (s) to ODBC.ini on each Oracle Access Manager server host.

As appropriate for the database application you are using, see ["To create an ODBC data source definition \(Windows\)"](#) on page 11-30.

6. As appropriate for the database application you are using, proceed to ["To make the RDBMS profile visible \(Windows\)"](#) on page 11-35 or ["To make the RDBMS profile visible \(supported Unix platforms\)"](#) on page 11-35.

To make the RDBMS profile visible (Windows)

1. On any Oracle Access Manager server host, navigate to My Computer, Manage, Services and Applications, Services.
2. Right-click the icon representing the Oracle Access Manager server on the machine, then select Restart from the dropdown menu.

If you installed both an Access Server and a Identity Server on the same machine, perform this procedure for both servers.

3. Repeat this procedure for all the Oracle Access Manager server hosts you wish to connect tot the audit database.
4. Proceed to ["Configuring Auditing"](#) on page 11-35.

To make the RDBMS profile visible (supported Unix platforms)

1. On a machine hosting a Oracle Access Manager server, run one of the following commands to stop your Oracle Access Manager server.
 - Access Servers: `stop_access_server`
 - Identity Servers: `stop_ois_server`
2. Run one of the following commands to start your Oracle Access Manager server.
 - Access Servers: `start_access_server`
 - Identity Servers: `start_ois_server`
3. Repeat this procedure for all the Oracle Access Manager server hosts you wish to connect to the audit database.
4. Proceed to ["Configuring Auditing"](#) on page 11-35.

Configuring Auditing

You can configure Oracle Access Manager for both file-based and database auditing.

By default, both file-based auditing and database auditing are turned off for all Oracle Access Manager servers. You can manually enable file-based and database auditing for each Oracle Access Manager server in your system.

You can configure audit options on a system-wide, server, event, and application basis. See ["About Audit Options"](#) on page 11-4 for a summary.

The defaults for auditing are optimal for most situations. You need to turn on the type or types of auditing you want on the Oracle Access Manager servers that you want to audit. If you send data to the audit database, you must also replace the default audit data format string on both the Identity and Access systems. See ["To modify audit](#)

[output formatting for the Identity System](#)" on page 11-37 and ["To modify audit output formatting for the Access System"](#) on page 11-42 for details.

Note that the global auditing settings that you configure in the Common Configuration sub-tab of the Identity System Console are different from the application-specific events that you audit from the sub-tabs for User, Group, and Org. Manager Configuration.

Task overview: To configure auditing

1. Turn on either or both file-based and database auditing for each Identity Server, and modify audit file attributes, if you wish.

See ["To enable and configure auditing for each Identity Server"](#) on page 11-36.
2. Configure the audit output formatting for the Identity System.

See ["To modify audit output formatting for the Identity System"](#) on page 11-37 for details.
3. Specify what data for which events will be audited.

This includes the following categories:
 - a. Events common to the User, Group, and Organization Manager applications.

See ["To specify global Identity System events and profile attributes for audit"](#) on page 11-38 for details.
 - b. User, Group, or Organization Manager events.

See ["To specify User, Group, or Organization Manager events for audit"](#) on page 11-39 for details.
4. Verify that all Identity Servers can record data to the audit database.

See ["To verify that all Identity Servers can record data to the audit database \(Windows\)"](#) on page 11-40 for details.
5. Turn on file-based or database auditing for individual Access Servers, and modify audit file attributes, if you wish.

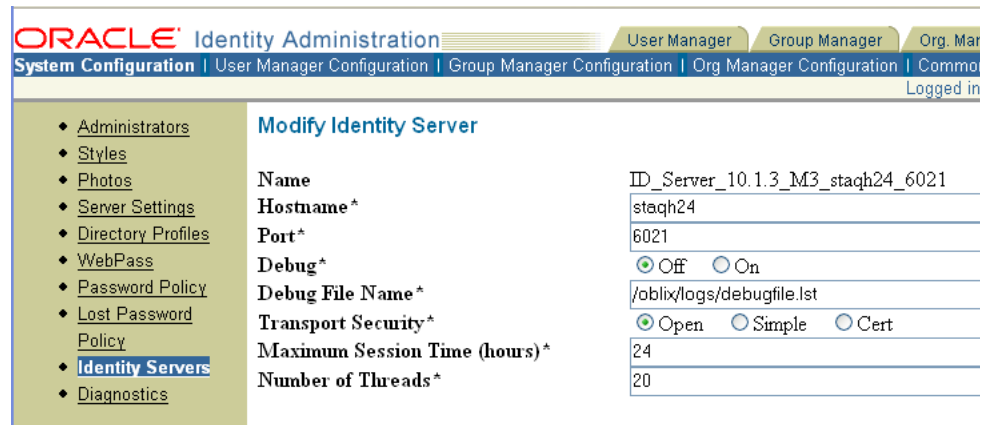
See ["To enable and configure auditing for each Access Server"](#) on page 11-41 for details.
6. Globally modify the audit output formatting for the Access system.

See ["To modify audit output formatting for the Access System"](#) on page 11-42 for details.
7. Create and manage User access privilege reports.

See ["To create and manage user access privilege reports"](#) on page 11-43 for details.

To enable and configure auditing for each Identity Server

1. From the Identity System Console, click the System Configuration sub-tab, then click Identity Servers in the left navigation pane.
2. Click the link for the server that you want to modify, then click Modify.



ORACLE Identity Administration | User Manager | Group Manager | Org. Manager

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | Common Configuration | Logged in

- Administrators
- Styles
- Photos
- Server Settings
- Directory Profiles
- WebPass
- Password Policy
- Lost Password Policy
- Identity Servers**
- Diagnostics

Modify Identity Server

Name	ID_Server_10.1.3_M3_staqh24_6021
Hostname*	staqh24
Port*	6021
Debug*	☒ Off ☐ On
Debug File Name*	/oblix/logs/debugfile.lst
Transport Security*	☒ Open ☐ Simple ☐ Cert
Maximum Session Time (hours)*	24
Number of Threads*	20

- Set the file auditing and database auditing flags according to your preference, and change whichever audit file attributes you prefer.

Click save to put your changes into effect.

- For database auditing, open the globalparams.xml file in the following directory. For example:

`Component_install_dir/identity/apps/common/bin/globalparams.xml`

where `component_install_dir` is the location where the Access or Identity Server was installed.

Set the value of the SQLDBType parameter in globalparams.xml as follows:

Oracle: Indicates an Oracle Database that uses an ODBC connection type.

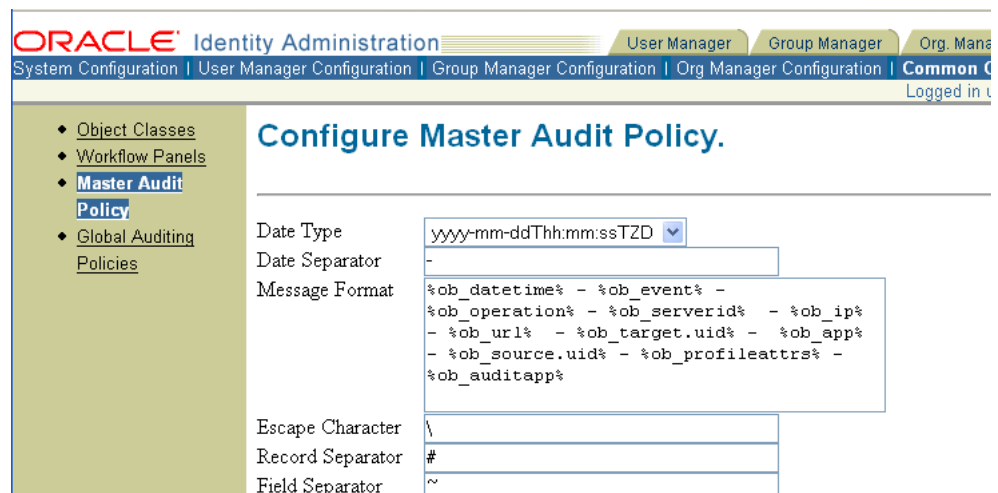
Oracle_OCI: Indicates an Oracle Database that uses an OCI connection type.

SQLServer: Indicates a SQL Server database. This is the default.

- Repeat this for all Identity Servers in your Oracle Access Manager system, then proceed to ["To modify audit output formatting for the Identity System"](#) on page 11-37.

To modify audit output formatting for the Identity System

- From the Identity System Console, click the Common Configuration sub-tab, then click Master Audit Policy in the left navigation pane, then click Modify.



ORACLE Identity Administration | User Manager | Group Manager | Org. Manager

System Configuration | User Manager Configuration | Group Manager Configuration | Org Manager Configuration | **Common Configuration** | Logged in

- Object Classes
- Workflow Panels
- Master Audit Policy**
- Global Auditing Policies

Configure Master Audit Policy.

Date Type	yyyy-mm-ddThh:mm:ssTZD
Date Separator	-
Message Format	%ob_datetime% - %ob_event% - %ob_operation% - %ob_serverid% - %ob_ip% - %ob_url% - %ob_target.uid% - %ob_app% - %ob_source.uid% - %ob_profileattrs% - %ob_auditapp%
Escape Character	\
Record Separator	#
Field Separator	~

2. Click anywhere within the Message Format text box, press Control-A to select everything within the text box, even the contents that are obscured, then press Delete.
3. In the empty text box, insert the following string:

```
%ob_datetime% - %ob_event% - %ob_operation% - %ob_serverid% - %ob_ip% - %ob_url% - %ob_target.uid% - %ob_app% - %ob_source.uid% - %ob_profileattrs% - %ob_auditapp%
```

Do not add a semi-colon or line return to the end of the string.
4. If you prefer, modify the default values in the Date Type, Date Separator, Escape Character, Record Separator, and Field Separator fields.

Note that if you change any of these values, you will need to reconfigure the Crystal report templates used to generate Audit Reports.
5. Click Save.

The new message format string and any other changes you made will display in the Configure Master Audit Policy page.
6. The new message format string applies across the Identity System, so you do not need to repeat the process for the other Identity Servers, but you do need to perform a similar procedure to set the format string for the Access system.

See ["To modify audit output formatting for the Access System"](#) on page 11-42 for details.
7. Proceed to ["To specify global Identity System events and profile attributes for audit"](#) on page 11-38.

To specify global Identity System events and profile attributes for audit

1. From the Identity System Console click the Common Configuration sub-tab, then click Global Auditing Policies in the left navigation pane, then click Modify.

Event name	Application auditing enabled	Audit success	Audit failure
Login	☒	☒	☒
Logout	☒	☒	☒
Password Management	☒	☒	☒

2. Select up to five profile attributes to audit.

Profile attributes (Full Name, Employee Number, Department Number, and the like) are specific to the user performing the action/event being audited (Search or View Profile or Modify Profile, for example). The purpose of profile attributes is to help you identify the user performing the action/event.

WARNING: To avoid exposing a challenge phrase or response attribute, Oracle recommends that you do not select these as profile attributes for auditing. If you add a challenge phrase or response as a profile attribute, it is audited in proprietary encoded format.

3. Modify the default audit flag settings for the common User, Group, and Organization Manager application events you prefer.
4. Click Save to apply these settings to all the Identity Servers in your system.
5. Proceed to ["To specify User, Group, or Organization Manager events for audit"](#) on page 11-39.

To specify User, Group, or Organization Manager events for audit

1. From the Identity System Console, click the User, Group, or Org. Manager Configuration sub-tab, then click Audit Policies in the left navigation pane, then click Modify.

ORACLE Identity Administration User Manager Group Manager

System Configuration | **User Manager Configuration** | Group Manager Configuration | Org Manager Configuration | Lc

◆ Tabs
◆ Reports
◆ **Audit Policies**

Modify Application Auditing Policy

Profile Attributes

Full Name ▾
— ▾
— ▾
— ▾
— ▾
— ▾

Event name	Application auditing enabled	Audit success	Audit failure
Search	☒	☐	☒
View Profile	☒	☐	☒
Modify Profile	☒	☒	☒
View Location	☒	☐	☒

2. Select up to five profile attributes to audit.

Profile attributes (Full Name, Employee Number, Department Number, and the like) are specific to the user performing the action/event being audited (Search or View Profile or Modify Profile, for example). The purpose of profile attributes is to help you identify the user performing the action/event.

WARNING: To avoid exposing a challenge phrase or response attribute, Oracle recommends that you do not select these as profile attributes for auditing. If you add a challenge phrase or response as a profile attribute, it is audited in proprietary encoded format.

3. Modify the default audit flag settings for whichever common User Manager application events you prefer. Identity Server
4. Click Save to apply these settings to all the Identity Servers in your system.

To verify that all Identity Servers can record data to the audit database (Windows)

1. From any page in the Identity System Console for any Identity Server for which you have completed all the audit setup procedures up to this point, click Logout in the upper right corner of the application window.
2. Click OK when asked if you really want to log out.
3. Open the SQL Server Query Analyzer window on the machine hosting your audit base.

You minimized this window when you completed the procedure ["SQL Server on Windows: To verify the audit schema"](#) on page 11-26.

If, for any reason, the window is no longer open, re-launch it by navigating to: Start, Programs, Microsoft SQL Server, Query Analyzer, File, Open, *Login_Credentials*, OK, File, Open, *audit_sql_path*, OK

where *Login_Credentials* is the user name and password you specified when installing SQL Server and *audit_sql_path* is the path to the audit.sql file you copied to the audit database host and subsequently modified in the procedure ["SQL Server on Windows: To verify the audit schema"](#) on page 11-26.

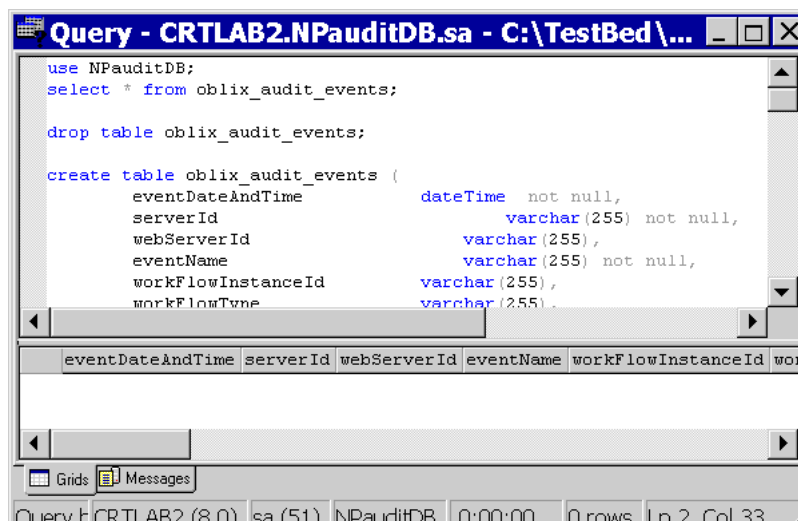
4. Press F5 to execute audit.sql.

You previously saved audit.sql after adding the following lines:

```
use AuditDBName;
select * from oblix_audit_events;
```

where *AuditDBName* specifies the audit database you created in the procedure ["SQL Server on Windows: To create the audit database"](#) on page 11-21.

The column headings for the Oracle Access Manager schema appear at the bottom of the Query window with particulars for the logout under the appropriate columns.



5. Proceed to ["To enable and configure auditing for each Access Server"](#) on page 11-41.

To verify that all Identity Servers can record data to the audit database (Supported Unix platforms)

1. From any page in the Identity System Console of any Identity Server for which you have completed all the audit setup procedures up to this point, click Logout in the upper right corner of the application window.
2. Click OK when asked if you really want to log out.
3. Perform the following in the iSQL *Plus Web application of the Oracle Database Server:

Log in to iSQL *Plus by providing the user name, password, and GDN of the database.

Enter following command in the iSQL *Plus Workspace:

```
select * from oblix_audit_events;
```

The columns headings for the Oracle Access Manager audit schema appear in the iSQL *Plus Workspace page. Information regarding logout appears under the appropriate column headings.

4. Proceed to ["To enable and configure auditing for each Access Server"](#) on page 11-41.

To enable and configure auditing for each Access Server

1. On any Access Server you plan to connect to the audit database, navigate to: Access System Console, Access System Configuration, Access Server Configuration, *ServerName*, Modify

where *ServerName* specifies the Access Server you want to modify.

2. Set the file auditing and database auditing flags according to your preference.
3. Change whichever audit file attributes you prefer, then click Save to commit your changes.

If you change any of the attributes marked with asterisks, you must restart your Access Server to make the changes take effect.

4. Repeat this for all Access Servers in your Oracle Access Manager system, then proceed to ["To modify audit output formatting for the Access System"](#).
5. For database auditing, open the globalparams.xml file in the following directory:

Component_install_dir/apps/common/bin/

where *component_install_dir* is the location where the Access or Identity Server was installed.

Set the value of the SQLDBType parameter in globalparams.xml as follows:

SQLServer: Indicates a SQL Server database. This is the default.

Oracle: Indicates an Oracle Database that uses an ODBC connection type.

Oracle_OCI: Indicates an Oracle Database that uses an OCI connection type.

To modify audit output formatting for the Access System

1. On any Access Server you plan to connect to the audit database, navigate to: Access System Console, Access System Configuration, Common Information Configuration, Master Audit Rule, Add (or Modify).

ORACLE Access Administration Access Manager

System Configuration System Management **Access System** Logged in u

Shared Secret Master Audit Rule Resource Type Definitions Flush Password Policy

Duplicate Actions

Add the Master Audit Rule

Profile Attributes - +

Audit Events

Authentication Success	☐
Authentication Failure	☐
Authorization Success	☐
Authorization Failure	☐

Audit Event Mapping

Authentication Success	AUTHN_SUCCESS
Authentication Failure	AUTHN_FAIL
Authorization Success	AUTHZ_SUCCESS
Authorization Failure	AUTHZ_FAIL

Audit Date Type 12/31/1999 Format ▼

Audit Escape Character \

Audit Record Format

```
%ob_datetime% - %ob_event% -
%ob_operation% - %ob_serverid% - %ob_ip%
- %ob_url% - %ob_userid% -
%ob_time_no_offset% - %ob_resrc_scheme% -
%ob_wgid% - %ob_wgcontext% - %ob_reason%
```

☐ Update Cache

2. Click anywhere within the Audit Record Format text box, press Control-A to select everything within the text box, even the contents that are obscured, then press Delete.
3. In the empty text box, insert exactly what appears in the following string:

```
%ob_datetime% - %ob_event% - %ob_operation% - %ob_serverid% -
%ob_ip% - %ob_url% - %ob_userid% - %ob_time_no_offset% - %ob_resrc_scheme% - %ob_wgid% - %ob_wgcontext% - %ob_reason%
```

 Do not add a semi-colon or line return to the end of the string.
4. In the Profile Attributes box, type the name of a profile attribute you want to audit, then click the plus sign (+) to the right of the text box.
 Repeat this step to add other profile attributes.
5. Select the events you want to audit.

- If you prefer, modify the default event mappings.
 - If you prefer, modify the default values in the Audit Date Type and Audit Escape Character fields. Remain aware that if you do change any of these values, you need to reconfigure the Crystal report templates used to generate Audit Reports.
6. Click Save.
The new message format string and any other changes you made appear in the Master Audit Rule page.
 7. The new message format string applies across the Access System, so you do not need to repeat the process for the other Access Servers, but you do need to perform a similar procedure to replace the format string for the Identity System.
See ["To modify audit output formatting for the Identity System"](#) on page 11-37.
 8. Proceed to ["To create and manage user access privilege reports"](#) on page 11-43.

To create and manage user access privilege reports

1. On any Access Server you plan to connect to the audit database, navigate to Access System Console, System Management, Manage Reports, Add.

ORACLE Access Administration Access Manager

System Configuration **System Management** Access System

Logged in as:

- ◆ Diagnostics
- ◆ **Manage Reports**
- ◆ Manage Sync Records

Add a new Report

User Access Privileges Report

Report Name

Description

Access Server

Results Storage

☒ Store in Database

☐ Store in File

Name of File

URL	Resource Type	Resource Operation
Add		

2. In the Report Name field, type a descriptive name such as "Midnight Access."
3. In the Description field, type a longer explanation of the report content, such as "Who has night shift access to the loading dock shipping manifest URLs."
4. Specify whether to send the information to the audit database or the audit file on the local host. If you specify the audit file, you must provide a file name.
5. In the "From this IP Address field," type the IP of the host for a specific web browser whose access you want to test.
6. In the "Date/Time..." field, select the date, time, and time zone for which you wish to test access.

This can be a point in the future, because the audit feature does not actually report the historical results of a actual access attempt; rather, it consults the policy and profile information stored on the Oracle Access Manager directory server to

calculate whether the specified users currently have permission to access the specified resource at the specified time.

7. Click the Add button near the List of Resources label to add URLs to the list of resources to be tested.

The Add Resource Rule page appears.

8. Type the URL to be tested.
9. Set the Resource type to http or ejb.
10. Check the action(s) you want tested.
11. Click Save to return to the Add a New Report page.
12. Click Add again to add another resource to be tested, or proceed to the next step.
13. You can test access for all users, or you can use the Selector to test access for specific users.
See ["The Selector"](#) on page 1-10 for details on the Selector.
14. When you are done with the Selector and the Add a new Report page reappears, click Save to commit your changes.

Setting up Audit Reports

To make use of the preconfigured Crystal Reports templates supplied with Oracle Access Manager, you must install the Crystal Reports application on a Windows machine within your Oracle Access Manager server domain. (Crystal Reports cannot be installed on Unix machines, but it can make use of information in a database generated by the Oracle Database installed on a Unix machine.)

In addition to installing Crystal Reports 9, you must also install a patch.

The Oracle Access Manager server installation directories are installed with certain templates, sample reports, database schema, and database drivers which are used by the Crystal Reports application. These are distinct from the Crystal Reports software itself. You must copy then from a Oracle Access Manager server install directory to the machine hosting your Crystal Reports software.

Task overview: To set up audit reports

1. Install Crystal Reports 9.22a on a Windows machine that can connect to the machine hosting SQL Server or the Oracle Database.
2. Install the mandatory patch for Crystal Reports 9.

3. Copy the Oracle Access Manager audit report templates, the Crystal Repository, and associated resources to the machine hosting Crystal Reports.
4. Connect Crystal Reports to the Oracle Access Manager audit database by creating an ODBC data source definition and editing orMap.ini.
5. Connect Crystal Reports to the Crystal database by creating an ODBC data source definition and editing orMap.ini.
6. Connect Crystal Reports to the oblix_audit_events table.

To install Crystal Reports

1. Obtain a copy of the Crystal Reports 9.22 installation package from the vendor.
2. Launch setup.exe and follow the prompts.
3. Specify whichever installation directory you prefer.
4. When prompted, enter the product key, which is provided with the purchase of the reporting package.
5. When prompted, specify "typical" for the installation type.
6. Proceed to "[To install the patch for Crystal Reports](#)" on page 11-45.

To install the patch for Crystal Reports

1. Download the Crystal Reports 9 patch from the following Web site:
http://support.businessobjects.com/communityCS/FilesAndUpdates/cr90dbexwin_en.zip.asp
2. Unzip cr90dbexwin_en.zip into a temporary folder on your hard disk, then launch CR90DBEXWIN_EN_200403.EXE.
3. Follow the prompts to complete the patch installation.
4. Proceed to "[To copy the Oracle Access Manager-specific Crystal resources](#)" on page 11-45.

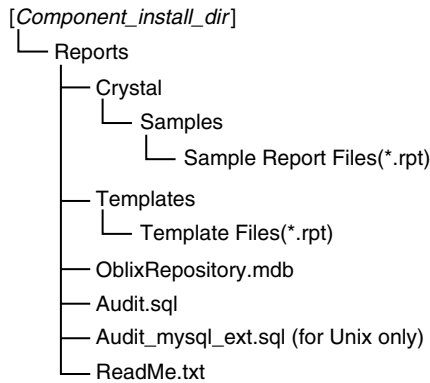
To copy the Oracle Access Manager-specific Crystal resources

1. Using whatever methods you are comfortable with, copy the following resources from a Oracle Access Manager server installation to a directory of your choice on the machine hosting Crystal Reports.

Component_install_dir\oblix\reports

Where *Component_install_dir* is the root installation directory for an Identity Server that is connected to the audit database.

Make sure to copy everything in "..\reports" and its subdirectories. The following diagram shows the resources copied to the Crystal Reports machine.

Figure 11-4 Resources copied to the Crystal Reports machine

2. Proceed to ["To connect Crystal Reports to the audit database"](#) on page 11-46.

To connect Crystal Reports to the audit database

1. Follow the procedure described in ["To create an ODBC data source definition \(Windows\)"](#) on page 11-30 so that Crystal Reports can connect to the audit database.

Make sure that the DSN you specify and all associated details match exactly the values you specified for the RDBMS profile and the ODBC data source definitions you created for the Oracle Access Manager servers that connect to the audit database.

2. Proceed to ["Task overview: To connect Crystal Reports to the Oracle Repository"](#) on page 11-46.

Task overview: To connect Crystal Reports to the Oracle Repository

1. Create an ODBC data source definition to connect Crystal Reports to the Oracle/Crystal Repository (.mdb database).
2. Edit orMap.ini to equate the Oracle Repository with the Crystal Repository.

To create an ODBC data source definition to connect Crystal Reports to the Oracle/Crystal Repository

1. Follow the general procedure described in ["To create an ODBC data source definition \(Windows\)"](#) on page 11-30 so that Crystal Reports can connect to the audit database.

Except where noted in the steps that follow, use the values specified in the original procedure.

2. When prompted for a database driver, select "Microsoft Access driver (.mdb)."
3. For the Name parameter, choose some self-explanatory name such as OracleRepositorySysDSN.
4. Proceed to ["To edit orMap.ini"](#) on page 11-46.

To edit orMap.ini

1. On the machine hosting Crystal Reports, navigate to:

C:\Program Files\Common Files\Crystal Decisions\2.5\bin

2. Open the file orMap.ini in any plain text editor.
3. Replace the line "Crystal Repository=Crystal Repository" with the following:

`Crystal Repository = repository_DSN`

where *repository_DSN* is the System DSN you created for the OracleRepository.mdb file. We have been using OracleRepositorySysDSN in our example.

4. Proceed to ["To connect Crystal Reports to the oblix_audit_events table"](#) on page 11-47.

To connect Crystal Reports to the oblix_audit_events table

1. In Crystal Reports, open the report.
2. Press **F5** to connect the report with the audit database.
3. If you are prompted for new parameter values, provide appropriate parameter values for the report.
4. Select the ODBC data source that is connected to the audit database and click **Next**.
5. Enter the user ID and password, and click **Finish**.
6. From the **Database** menu, select **Set Datasource Location**.
7. In the **Current Data Source** section, select the report name.
8. In the **Replace with** section, select **History**, then select the **ODBC Data source name** used to connect to the audit database, then select the **User Id** used while connecting, then select **Tables**, then **oblix_audit_events**.
9. Click **Update**.
10. To view the report, click **Close**.

SNMP Monitoring

This chapter focuses on network monitoring through the Simple Network Management Protocol (SNMP).

SNMP monitoring is one of several methods of gathering information on your Oracle Access Manager system. Logging, auditing, and other reporting features, are described elsewhere in this guide.

This chapter includes the following topics:

- [Prerequisites](#)
- [About Oracle Access Manager SNMP Monitoring and Agents](#)
- [About the Oracle Access Manager MIB and Objects](#)
- [About Integration with Oracle Enterprise Manager 10g Identity Management](#)
- [Enabling and Disabling SNMP Monitoring](#)
- [Setting Up SNMP Agent and Trap Destinations](#)
- [Changing SNMP Configuration Settings](#)
- [Logging for SNMP](#)
- [SNMP Messages](#)
- [Discrepancies Between Netstat and SNMP Values](#)
- [Configuring the Shutdown Interval](#)

Note: For information about installing SNMP, refer to the *Oracle Access Manager Installation Guide*.

Prerequisites

You need to have a network management station (NMS) installed, and you should be familiar with how to upload and display network statistics gathered from a Management Information Base (MIB). This chapter describes the Oracle Access Manager MIB objects and the Object Identifiers (OIDs) for these objects. However, this chapter does *not* provide information on how to use these OIDs in your NMS to collect statistics. For such information, refer to the documentation for your NMS.

About Oracle Access Manager SNMP Monitoring and Agents

The Simple Network Management Protocol (SNMP) enables you to monitor component activity on the network that hosts your Oracle Access Manager system by

collecting and displaying server-related SNMP data on a network management station. SNMP statistics commonly include data such as:

- The hosts, routers, and servers on your network
- The number of requests being processed on a particular device
- Whether or not a particular device is running
- Whether requests were processed successfully

SNMP data is displayed on a network management station (NMS). The NMS is a workstation running a network management application such as HP OpenView. You configure the NMS to display network statistics in a useful way, for instance, as a graph to show simple network statistics or to show whether the number of requests a device is processing falls within a set of defined limits.

You can capture SNMP statistics for the Identity Server and the Access Server running on any supported platform. Oracle Access Manager supports SNMP polling and trapping. Polling collects information such as:

- The version number of a component
- Configuration status
- Connection status
- Statistics on actions the component has processed

Event traps include information such as:

- Component failure
- Event failure
- Connection status
- Failure to complete actions

Note: Oracle Access Manager supports version 2 of the SNMP protocol.

The SNMP Agent

The Simple Network Management Protocol (SNMP) is an application-layer protocol that enables network devices to exchange information. By using SNMP-transported data (such as successful operations and failure conditions), administrators can monitor network performance and solve problems. The Oracle Access Manager's SNMP Agent enables you to implement SNMP-based data collection for the Identity Server and Access Server. The SNMP Agent enables collection of information such as the number of successful authentications performed by the Access Server and the number of requests processed by the Identity Server.

The SNMP Agent is an optional installable component. The Agent collects information on the host where it is installed, so you must install an Agent on each host where you want to collect SNMP data. If installed, the Agent accesses information about the Identity or Access Server resident on the same server host on which the Agent was installed. The Agent is installed in *SNMP_install_dir*.

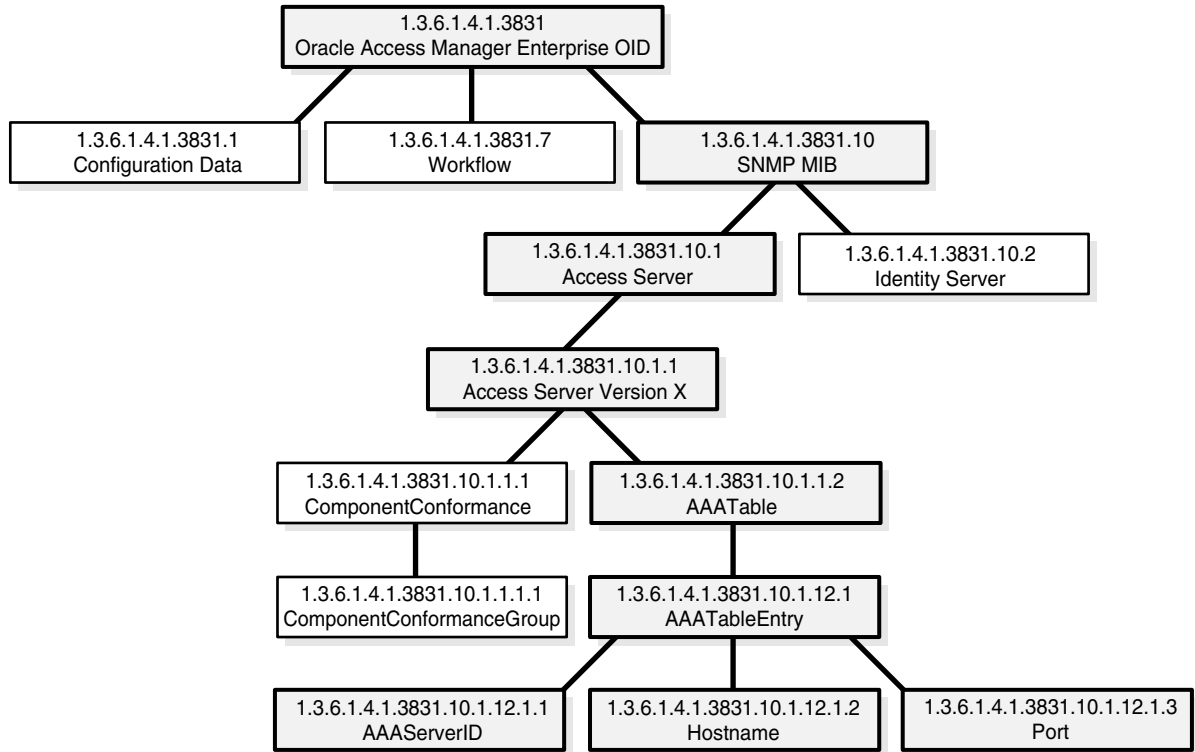
For information on installing the SNMP Agent, see the *Oracle Access Manager Installation Guide*.

About the Oracle Access Manager MIB and Objects

The Management Information Base (MIB) is a specification file that contains variables relevant to the status of different Oracle Access Manager components. The SNMP Agent collects values for fields in the MIB.

Figure 12–1 illustrates the Oracle Access Manager MIB hierarchy.

Figure 12–1 The MIB hierarchy



The Oracle Access Manager MIB can be expressed as a concatenation of branch and object identifiers (OIDs). The label from the MIB root to the top node of the MIB is as follows:

```
iso.org.dod.internet.private.enterprises.oblix.snmp
```

MIB files are located in `SNMP_install_dir/oblix/mibs`. These files conform to SNMP Version 2.

The following discussions describe the MIB objects that are provided with the Oracle Access Manager SNMP component.

Note: Refer to your NMS documentation for information on uploading the MIB files to your NMS.

MIB Index Fields

Each MIB table contains one or more index fields. The index field values help you identify a unique row in the table.

For example, the index fields for `coreidInstanceTable` described in "[Identity Server MIB Objects](#)" on page 12-4 are `coreidHostname` and `coreidPort`. These entries are used as

indexes because they uniquely identify an installation. Suppose that you have two Identity Servers named Identity1 and Identity2, each with a host name of localhost using ports 6023 and 6024, respectively. The indexes for these servers would be localhost.6023 and localhost.6024.

To retrieve the first column value for Identity1, the object identifier you would request from the SNMP Agent would take the following logical form:

```
1.3.6.1.4.1.3831.10.1.1.2.1.1.localhost.6023
```

where 1.3.6.1.4.1.3831.10.1.1.2.1.1 signals that you want the first column of coreidInstanceTable, for the element with an index value of localhost.6023. The index is represented in numeric notation (similar to specifying an OID) which actually contains the length of the string followed by ascii codes for the characters in the string. As a result, this example:

```
1.3.6.1.4.1.3831.10.1.1.2.1.1.localhost.6023
```

is actually represented as follows:

```
1.3.6.1.4.1.3831.10.1.1.2.1.1.9.108.111.99.97.108.104.111.115.
116.6023
```

Note: If you want the entire table to be returned in your SNMP requests, It is not necessary to know the values of the index fields.

Identity Server MIB Objects

Table 12–1 contains the Identity Server objects in the MIB. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.coreid. versionone
```

The name of this table is coreidInstanceTable. Its index fields are coreidHostname and coreidPort. It describes Identity Server instances.

Table 12–1 Identity Server MIB Objects

Managed Object	Syntax	Description
coreidInstanceTable OID: 1.3.6.1.4.1.3831.10.1.1.2	n.a.	Primary table name.
coreidId OID: 1.3.6.1.4.1.3831.10.1.1.2.1.1	SnmpAdminString (size 0-255)	The identifier for the Identity Server instance.
coreidHostname OID: 1.3.6.1.4.1.3831.10.1.1.2.1.2	SnmpAdminString (size 0-255)	The hostname of the machine on which this Identity Server runs. The hostname is an index for this table.
coreidPort OID: 1.3.6.1.4.1.3831.10.1.1.2.1.3	Integer (0-65535)	The port on which the Identity Server listens. The port number is an index for this table.
coreidMode OID: 1.3.6.1.4.1.3831.10.1.1.2.1.4	Integer (0-5)	The transport security mode between the Identity Server and WebPass. 0—Open 1—Simple 2—Cert

Table 12–1 (Cont.) Identity Server MIB Objects

Managed Object	Syntax	Description
coreidStartTime OID: 1.3.6.1.4.1.3831.10.1.1.2.1.5	DateAndTime	The time when the Identity Server was last started.
coreidServiceThreads OID: 1.3.6.1.4.1.3831.10.1.1.2.1.6	Integer (0-65535)	The number of service threads in the Identity Server instance. The number of threads is set in the administration console. The parameter NumberOfServiceThreads in scoreboard_params.lst controls how many slots are allocated (using one for each service thread) to maintain SNMP information for each service thread.
coreidNumOfLanguagesConfigured OID: 1.3.6.1.4.1.3831.10.1.1.2.1.7	Integer (0-65535)	The number of languages installed for this Identity Server instance.
coreidNumOfLogins OID: 1.3.6.1.4.1.3831.10.1.1.2.1.8	counter64	The number of successful logins to the Identity Server instance.
coreidNumOfLoginsFailure OID: 1.3.6.1.4.1.3831.10.1.1.2.1.9	Counter64	The number of failed login attempts to the Identity Server instance.
coreidRequestsProcessed OID: 1.3.6.1.4.1.3831.10.1.1.2.1.10	Counter64	The number of requests processed by the Identity Server instance.
coreidNumOfRequestsSuccess OID: 1.3.6.1.4.1.3831.10.1.1.2.1.11	Counter64	The number of requests successfully handled by this Identity Server instance.
coreidNumOfRequestsFail OID: 1.3.6.1.4.1.3831.10.1.1.2.1.12	Counter64	The number of requests for this Identity Server that produced an error.
coreidTotalServiceTime OID: 1.3.6.1.4.1.3831.10.1.1.2.1.13	Counter64	Total time, in nanoseconds, the Identity Server has taken to serve requests since the last restart?
coreidTotalNumOfCacheFlushRequestSuccess OID: 1.3.6.1.4.1.3831.10.1.1.2.1.14	Counter64	Total number of successful cache flush requests issued by the Identity Server.
coreidTotalNumOfCacheFlushRequestFail OID: 1.3.6.1.4.1.3831.10.1.1.2.1.15	Counter 64	Total number of unsuccessful cache flush requests issued by the Identity Server
coreidNumOfPluginsLoaded OID: 1.3.6.1.4.1.3831.10.1.1.2.1.16	Counter64	The number of plug-ins loaded by the Identity Server instance.
coreidNumOfEmailSentFail OID: 1.3.6.1.4.1.3831.10.1.1.2.1.17	Counter64	The number of failed attempts to send email from this Identity Server instance.

Table 12–1 (Cont.) Identity Server MIB Objects

Managed Object	Syntax	Description
coreidOverflowFlagDirectoryServerSlots OID: 1.3.6.1.4.1.3831.10.1.1.2.1.18	Integer (0-65535)	A flag indicating that the number of configured SNMP information slots for the directory server was insufficient. The variable NumberOfConfiguredDS in scoreboard_params.lst defines the number of slots, using one slot for each directory server. If the value of NumberOfConfiguredDs is less than the actual number of directories that the Identity Server has contacted, the value for coreidOverflowFlagDirectoryServerSlots is set to 1. This flag only indicates an overflow condition. It does not convey how many slots are missing.
coreidOverflowForPPPACTIONSLOTS OID: 1.3.6.1.4.1.3831.10.1.1.2.1.19	Integer (0-65535)	The number of "hooked up" Identity Event API plug-in actions for which a slot could not be allocated.

Table 12–2 contains the MIB objects for capturing information about the Identity Event API plug-in, which enables you to create external events for workflows. More information about this plug-in is provided in the *Oracle Access Manager Developer Guide*. This table has three index fields: coreidHostname, coreidPort, and pppRowIndex. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.coreid.versionone.
pppActionsTable
```

Table 12–2 Identity Event API MIB Objects

Managed Object	Syntax	Description
pppActionsTable	n.a.	Primary table name.
pppRowIndex OID: 1.3.6.1.4.1.3831.10.1.1.3.1.2	Integer (0-65535)	This field is used only for indexing purposes. This value, along with its parent index values, forms a unique identifier for the row.
pppActionName OID: 1.3.6.1.4.1.3831.10.1.1.3.1.2	SnmpAdminString (size 0-255)	The name of the PPP action.
pppFunctionName OID: 1.3.6.1.4.1.3831.10.1.1.3.1.3	SnmpAdminString (size 0-255)	The name of the external function that is executed for the given hook.
pppPluginPath OID: 1.3.6.1.4.1.3831.10.1.1.3.1.4	SnmpAdminString (size 0-255)	The path for the PPP plug-in.
totalCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.5	Counter64	The total number of times the PPP action is executed.
pppOKCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.6	Counter64	The number of times that the return code STATUS_PPP_OK is received for this PPP action.
pppAbortCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.7	Counter64	The number of times that the return code STATUS_PPP_ABORT is received for this PPP action.
pppWorkflowRetryCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.8	Counter64	The number of times that the return code STATUS_PPP_WF_RETRY is received for this PPP action.

Table 12–2 (Cont.) Identity Event API MIB Objects

Managed Object	Syntax	Description
pppWorkflowAsyncCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.9	Counter64	The number of times the return code STATUS_PPP_WF_ASYNC is received for this PPP action.

Table 12–3 contains information about the directory server that communicates with the Identity Server. This table has three index fields: coreidHostname, coreidPort, and coreidDSRowIndex. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.coreid.versionone.coreidDirectoryServerTable
```

Table 12–3 Identity System Directory MIB Objects

Managed Object	Syntax	Description
coreidDirectoryServerTable	n.a.	Primary table name.
coreidDSRowIndex OID: 1.3.6.1.4.1.3831.10.1.1.4.1.1	Integer (0-65535)	This field is used for indexing purposes only. This value, along with its parent index values, forms a unique identifier for the row.
coreidDirectoryServerHost name OID: 1.3.6.1.4.1.3831.10.1.1.4.1.2	SnmpAdminString (size 0 - 255)	The hostname of the directory server.
coreidDirectoryServerPort OID: 1.3.6.1.4.1.3831.10.1.1.4.1.3	Integer (0-65535)	The directory server port.
coreidDirectoryServerMode OID: 1.3.6.1.4.1.3831.10.1.1.4.1.4	Integer (0-65535)	The directory server communication mode: 0—Open 1—SSL
coreidDirectoryServerNoOfLiveConnections OID: 1.3.6.1.4.1.3831.10.1.1.4.1.5	Integer (0-65535)	The number of connections against the directory.

Table 12–4 contains the Identity System objects in the MIB for system events that can be mapped to SNMP traps.

The SNMP Agent supports sending trap messages to multiple NMS systems. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.coreid. versionone
```

For example, the full path to the oblixCoreidServerDown trap is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.coreid.versionone.oblixCoreidServerDown
```

Table 12–4 Identity Server Traps

Managed Object	Fields sent with the trap	Description
oblixCoreidServerDown OID: 1.3.6.1.4.1.3831.10.1.1.0.7001	coreidId coreidHostname coreidPort	A trap generated when the SNMP Agent detects that the Identity Server has done a shutdown with errors. This trap contains the server ID, host name, and port.

Table 12–4 (Cont.) Identity Server Traps

Managed Object	Fields sent with the trap	Description
oblixCoreidServerStart OID: 1.3.6.1.4.1.3831.10.1.1.0.7002	coreidId coreidHostname coreidPort	This trap is generated when the SNMP Agent detects that the Identity Server has been started or restarted. This trap contains the server ID, host name, and port.
oblixCoreidServerFailure OID: 1.3.6.1.4.1.3831.10.1.1.0.7003	coreidId coreidHostname coreidPort	This trap is generated when the SNMP Agent detects that the Identity Server has not done a clean shutdown or has failed. This trap contains the server ID, host name, and port.
oblixCOREidDSFailure OID: 1.3.6.1.4.1.3831.10.1.1.0.7004	coreidId coreidHostname coreidPort coreidDirectoryServer Hostname coreidDirectoryServer Port	This trap is generated when the Identity Server detects that the directory server that it is connected to is down.

Access Server MIB Objects

Table 12–5 describes the Access Server SNMP objects that are available through the MIB. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.versionone
```

Table 12–5 Access Server MIB Objects

Managed Object	Syntax	Description
aaaInstanceTable OID: 1.3.6.1.4.1.3831.10.2.1.2	n.a.	Primary table name.
aaaId OID: 1.3.6.1.4.1.3831.10.2.1.2.1.1	SnmpAdminString (size 0-255)	The identifier for this Access Server instance, as specified in the Access System Console.
aaaHostname OID: 1.3.6.1.4.1.3831.10.2.1.2.1.2	SnmpAdminString (size 0-255)	The name of the machine where the Access Server was installed, as specified in the Access System Console. The host name is an index for this table.
aaaPort OID: 1.3.6.1.4.1.3831.10.2.1.2.1.3	Integer (0-65535)	The port on which the Access Server listens. The port number is an index for this table.
aaaMode OID: 1.3.6.1.4.1.3831.10.2.1.2.1.4	Integer (0-65535)	The transport security mode between the Access Server and other Identity or Access components. 0—Open 1—Simple 2—Cert
aaaNoOfQueues OID: 1.3.6.1.4.1.3831.10.2.1.2.1.5	Integer (0-65535)	The number of service queues for this Access Server instance.
aaaThreadsPerQueue OID: 1.3.6.1.4.1.3831.10.2.1.2.1.6	Integer (0-65535)	The number of threads for each service queue for this Access Server instance.
aaaNoOfListenerThreads OID: 1.3.6.1.4.1.3831.10.2.1.2.1.7	Integer (0-65535)	The number of listener threads spawned. There will be one thread for each WebGate-Access Server connection.

Table 12–5 (Cont.) Access Server MIB Objects

Managed Object	Syntax	Description
aaaNoofConnectionWatcherThreads OID: 1.3.6.1.4.1.3831.10.2.1.2.1.8	Integer (0-65535)	The number of LDAP connection watcher threads.
aaaOverflowFlagDirectoryServerSlots OID: 1.3.6.1.4.1.3831.10.2.1.2.1.9	Integer (0-65535)	A flag indicating whether there are insufficient slots for the number of directories configured for the Access Server. This means that the administrator needs to update the file <i>install_dir/access/oblix/config/obscoreboardparams.xml</i> . 0 - No overflow 1 - Overflow occurred
aaaOverflowForAuthenticationPluginSlots OID: 1.3.6.1.4.1.3831.10.2.1.2.1.10	Integer (0-65535)	The number of authentication plug-ins whose information could not be displayed. The administrator needs to update the <i>install_dir/access/oblix/config/obscoreboardparams.xml</i> file.
aaaOverflowForAuthorizationPluginSlots OID: 1.3.6.1.4.1.3831.10.2.1.2.1.11	Integer (0-65535)	The number of authorization plug-ins whose information could not be displayed. The administrator needs to update the <i>install_dir/access/oblix/config/obscoreboardparams.xml</i> file.
aaaTimeAuditLogWasRotated OID: 1.3.6.1.4.1.3831.10.2.1.2.1.12	DateAndTime	Time when the audit log file was rotated. This setting is determined in the configuration for this Access Server specified in the Access System Console.
aaaStartTime OID: 1.3.6.1.4.1.3831.10.2.1.2.1.13	DateAndTime	The date and time when this Access Server instance was last started.
aaaAuthenticationsSuccess OID: 1.3.6.1.4.1.3831.10.2.1.2.1.14	Counter64	The number of successful authentications by the Access Server instance.
aaaAuthenticationsSuccess OID: 1.3.6.1.4.1.3831.10.2.1.2.1.15	Counter64	The number of successful authentications by this Access Server instance.
aaaAuthenticationsDenied OID: 1.3.6.1.4.1.3831.10.2.1.2.1.16	Counter64	The number of unsuccessful authentications by this Access Server instance.
aaaAuthorizationsSuccess OID: 1.3.6.1.4.1.3831.10.2.1.2.1.17	Counter64	The number of successful authorizations by this Access Server instance.
aaaAuthorizationsDenied OID: 1.3.6.1.4.1.3831.10.2.1.2.1.18	Counter64	The number of unsuccessful authorizations by this Access Server instance.
aaaAuditRequests OID: 1.3.6.1.4.1.3831.10.2.1.2.1.19	Counter64	The number of audit requests made by this Access Server instance.

Table 12–6 is a sub-table of MIB objects that describe the directory server that communicates with the Access Server. This sub-table has index fields of `aaaHostname`, `aaaPort`, and `aaaRowIndex`. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.versionone.aaaDirectoryServerTable
```

Table 12–6 Access System Directory Server MIB Objects

Managed Object	Syntax	Description
aaaDirectoryServerTable OID: 1.3.6.1.4.1.3831.10.2.1.3	n.a.	Primary table name.
aaaDSRowIndex OID: 1.3.6.1.4.1.3831.10.2.1.3.1.1	Integer (0-65535)	An index field. It does not contain any information.
aaaDirectoryServerHostname OID: 1.3.6.1.4.1.3831.10.2.1.3.1.2	SnmpAdminString (size 0-255)	The directory host name.
aaaDirectoryServerPort OID: 1.3.6.1.4.1.3831.10.2.1.3.1.3	Integer (0-65535)	The directory server port.
aaaDirectoryServerMode OID: 1.3.6.1.4.1.3831.10.2.1.3.1.4	Integer (0-65535)	The directory server communication mode with the Access Server: 0—Open 1—SSL
aaaDirectoryServerNoOfLiveConnections OID: 1.3.6.1.4.1.3831.10.2.1.3.1.5	Integer (0-65535)	The number of connections between the Access Server and the directory server.

[Table 12–7](#) is a sub-table of MIB objects for capturing information on authentication plug-ins. This sub-table has index fields of aaaHostname, aaaPort, and authenticationPluginName. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.  
versionone.aaaauthenticationPluginsTable
```

Table 12–7 Authentication Plug-Ins MIB Objects

Managed Object	Syntax	Description
authenticationPluginsTable OID: 1.3.6.1.4.1.3831.10.2.1.4	n.a	Primary table name.
authenticationPluginName OID: 1.3.6.1.4.1.3831.10.2.1.4.1.1	SnmpAdminString (size 0-255)	The name of the plug-in. The authentication plug-in name is an index for this table.
AuthenticationPluginPath OID: 1.3.6.1.4.1.3831.10.2.1.4.1.2	SnmpAdminString (size 0-255)	The path of the authentication plug-in.
AuthenticationPluginStatus OID: 1.3.6.1.4.1.3831.10.2.1.4.1.3	Integer (0-65535)	The status of the plug-in: 0—Not loaded 1—Loaded

[Table 12–8](#), the authorizationPluginsTable has index fields of aaaHostname, aaaPort, and authorizationPluginName. The path to this information is:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.  
versionone.authorizationsPluginsTable
```

Table 12–8 Authorization Plug-Ins MIB Objects

Managed Object	Syntax	Description
authorizationPluginsTable OID: 1.3.1.4.1.3831.10.2.1.5	n.a.	Primary table name.
authorizationPluginName OID: 1.3.6.1.4.1.3831.10.2.1.5.1.1	SnmpAdminString (size 0-255)	The name of this plug-in.
AuthorizationPluginPath OID: 1.3.6.1.4.1.3831.10.2.1.5.1.2	SnmpAdminString (size 0-255)	The path of the authorization plug-in.
AuthorizationPluginStatus OID: 1.3.6.1.4.1.3831.10.2.1.5.1.3	Integer (0-65535)	The status of the plug-in: 0—Not loaded 1—Loaded

[Table 12–9](#) is a sub-table that describes the number of requests in the queue for the Access Server. This table has indexes of aaaHostname, aaaPort, and aaaRequestQueueNumber. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.  
versionone.requestQueueInfoTable
```

Table 12–9 Request Queue MIB Objects

Managed Object	Syntax	Description
requestQueueInfoTable OID: 1.3.6.1.4.1.3831.10.2.1.5	n.a.	Primary table name.
aaaRequestQueueNumber OID: 1.3.6.1.4.1.3831.10.2.1.6.1.1	Integer (0-65535)	Index for the request queue.
aaaRequestQueueSize OID: 1.3.6.1.4.1.3831.10.2.1.6.1.2	Integer (0-65535)	The number of requests in the queue.

[Table 12–10](#) contains objects in the MIB for system events that can be mapped to SNMP traps. The SNMP Agent supports sending trap messages to multiple NMS systems. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.versionone
```

For example, to add the full path to the oblixAAAServerDown trap, you would specify:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.versionone.  
oblixAAAServerDown
```

Table 12–10 Access Server Traps

Managed Object	Fields Sent with the Trap	Description
oblixAAAServerDown OID: 1.3.6.1.4.1.3831.10.2.1.0.7001	aaaId aaaHostname aaaPort	A trap generated when the SNMP Agent detects that the Access Server has done a clean shutdown. This trap captures the Access Server ID, host name, and port.

Table 12–10 (Cont.) Access Server Traps

Managed Object	Fields Sent with the Trap	Description
oblixAAAServerStart OID: 1.3.6.1.4.1.3831.10.2.1.0.7002	aaaId aaaHostname aaaPort	A trap generated whenever the Access Server is restarted. This trap captures the Access Server ID, host name, and port. The trap is generated immediately, so the time of the restart is the time of the trap generation.
oblixAAAServerFailure OID: 1.3.6.1.4.1.3831.10.2.1.0.7003	aaaId aaaHostname aaaPort	A trap generated when the SNMP Agent detects that the Access Server has not done a shutdown with errors or has failed. This trap captures the Access Server ID, host name, and port.
oblixAAADSFailure OID: 1.3.6.1.4.1.3831.10.2.1.0.7004	aaaId aaaHostname aaaPort aaaDirectoryServer Hostname aaaDirectoryServerPort	A trap generated when the Access Server detects that the directory server it is connected to is down.

About Integration with Oracle Enterprise Manager 10g Identity Management

In addition to using SNMP monitoring as described in this chapter, the Oracle Enterprise Manager 10g Identity Management pack provides out-of-box system modeling for Oracle Access Manager.

Oracle Enterprise Manager is the Oracle integrated management solution for managing your computing environment. The Oracle Enterprise Manager 10g Identity Management pack helps improve performance and availability, and reduce the cost and complexity of managing the Oracle Identity and Access Management Suite. Single-step discovery of Oracle Access Manager and other Oracle Identity Management products helps you quickly set up your monitoring environment.

Configuration management capabilities, out-of-box system modeling for Oracle Access Manager, and collection of key performance metrics help accelerate diagnostics. You can track configuration changes for Oracle Access Manager and compare and validate configurations against saved or live or reference configurations.

Configuration management metrics collected for the Oracle Access Manager Identity System include:

- Number of service threads
- Number of languages configured

Configuration management metrics collected for the Oracle Access Manager Access System include:

- Number of service queues
- Number of service threads per service queue
- Number of listener threads
- Number of connection watcher threads
- Size of request queues

With Oracle Enterprise Manager 10g Identity Management pack monitoring and diagnostics you can monitor availability, performance and overall deployment health. This helps you ensure the quality of services, isolate a problem to a particular component, diagnose the cause of a problem, and resolve it with minimal to no down time.

Oracle Enterprise Manager 10g Identity Management pack monitoring and diagnostic metrics for the Oracle Access Manager Identity System include:

- Average Service Time per Request
- Successful/Failed Logins
- Requests processed
- Successful/Failed Requests
- Successful/failed cache flush requests
- Failed sent Emails

Oracle Enterprise Manager 10g Identity Management pack monitoring and diagnostic metrics for the Oracle Access Manager Identity System include:

- Successful authentications
- Failed authentications
- Successful authorizations
- Failed authorizations
- Requests processed

Service Level Management capabilities help you proactively monitor the performance and availability of Identity and Access services using pre-recorded transactions. You can ensure that services are meeting performance expectations and provide visibility to stake holders on Service Level Performance.

A centralized console combined with service and system dashboards provide a top-down enterprise view with drilldown capabilities. Enterprise Manager improves performance & availability and reduces the cost of managing Oracle Access Manager and other Oracle Identity Management products.

For more information, see the *Oracle Enterprise Manager Concepts Guide* and *Oracle Enterprise Manager Advanced Configuration Guide*. Online help is available through Oracle Enterprise Manager.

Enabling and Disabling SNMP Monitoring

You use the Identity and Access Servers configuration pages to enable SNMP and to indicate the TCP/IP port where contact will be established with the SNMP Agent.

Note: Oracle Access Manager does *not* provide a configuration setting for a polling interval to retrieve SNMP statistics. However, most NMS systems provide a polling configuration parameter. This parameter is used by the NMS to periodically poll the Agent to retrieve MIB values.

The following procedure describes how to start and stop the Oracle Access Manager SNMP Agent, and how to start the Agent on another port.

To configure collection of SNMP statistics

1. From the Identity (or Access) System Console, select System Configuration, Identity Server (or Access Server.)
2. Click a link for a particular server.
3. Select the Modify button to display the page where you can turn SNMP monitoring on or off, as follows:
 - **Turn On:** Select the SNMP State On button at the bottom of the page.
 - **Turn Off:** Select the SNMP State Off button at the bottom of the page.
4. In the SNMP Agent Registration Port field, enter the port number to define or change the port on which the SNMP Agent listens.
5. Restart the Identity Server (or Access Server).

Setting Up SNMP Agent and Trap Destinations

You use the following command to setup an SNMP Agent against an SNMP Manager:

```
setup_agent -i
```

The -i option is required.

Following procedures describe and illustrate how to configure the Oracle Access Manager SNMP Agent and trap destinations.

To configure the SNMP Agent and trap destinations

1. Change to the directory containing the SNMP setup_agent command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

where *SNMPDIR* is the directory where you have installed the SNMP Agent.

2. Use the setup_agent command with the following options:

```
-i <install_dir>
```

```
-g Configure General Parameters
```

```
-u <Agent SNMP UDP Port>
```

```
-c <Agent Community String>
```

```
-p <Agent TCP Port>
```

```
-S <Run in silent mode>
```

```
--help Prints help message
```

To add a trap destination in silent mode

1. Change to the directory containing the SNMP setup_agent command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the setup_agent command with the following options:

```
-a
```

```
-m <Manager Station>
```


`-t <Trap port>`

To delete a trap destination in silent mode

1. Change to the directory containing the SNMP setup_agent command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the setup_agent command with the following options:

`-d`

`-m <Manager Station>`

`-t <Trap port>`

To configure general parameters first

1. Change to the directory containing the SNMP setup_agent command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the following setup_agent command:

```
> ./setup_agent -i $SNMPDIR -g -u <UDP Port> -c public -p <TCP Port>
```

This goes to the Manager Station Trap Configuration menu.

To add an SNMP Manager directly after general parameters

1. Change to the directory containing the SNMP setup_agent command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the following setup_agent command:

```
> ./setup_agent -i $SNMPDIR -a -m <Mgr M/c> -t <Mgr Port>
```

To delete an SNMP Manager directly after adding one

1. Change to the directory containing the SNMP setup_agent command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the following setup_agent command:

```
> ./setup_agent -i $SNMPDIR -d -m <Mgr M/c> -t <Mgr Port>
```

You can add any number of Manager Stations. The Agent then sends all the traps to the configured SNMP Managers.

Changing SNMP Configuration Settings

A configuration file named `obscoreboard_params.xml` contains information that defines the collection of SNMP statistics. This file is located in:

`Component_install_dir/identity|access/oblix/config`

where *Component_install_dir* is the directory where the component is installed and *identity|access* represents either the Identity Server or Access Server, respectively.

Identity System File: obscoreboard_params.xml

Access System File: obscoreboard_params.xml

In this file, you can configure threshold levels to determine when various MIB counters are activated.

The following parameters are specified only in the Access Server file obscoreboard_params.xml:

- **NumberOfAuthenticationPlugins:** The maximum number of authentication plug-ins that may be loaded in the Access System. The Access Server maintains information on the number of plug-ins that are loaded. If the actual number of plug-ins loaded by the Access Server exceeds the value specified for NumberOfAuthenticationPlugins, the difference is displayed as the counter aaaOverflowforAuthenticationPluginSlots.
- **NumberOfAuthorizationPlugins:** The maximum number of authorization plug-ins that may be loaded in the Access System. The Access Server maintains information on the number of plug-ins that are loaded. If the actual number of plug-ins loaded by the Access Server exceeds the value specified for NumberOfAuthorizationPlugins, the difference is displayed as the counter aaaOverflowforAuthorizationPluginSlots.

The following parameter is specified only in the Identity Server file obscoreboard_params.xml:

- **NumberOfPPPPluginActions:** The number of Identity Event API plug-in actions that may be connected with this Identity Server. When the Identity Server starts, it reads this value and monitors the actual number of Identity Event API plug-ins. If the number of active plug-ins exceeds the value for NumberOfPPPPluginActions, the difference is indicated by the counter coreidOverflowForPPPACTIONSLOTS.

The following parameters are provided in both scoreboard files:

- **NumberOfServiceThreads:** The value for this parameter is read by the Identity or Access Server at startup. This parameter controls how many slots to allocate (one for each service thread) to maintain SNMP information for each service thread. The server monitors the number of service threads being used. The actual number of service threads is configured through the administration console, from the command line, or as part of a configuration file. This parameter does not control the number of threads to be started by the Identity Server. If the actual number used exceeds this value, there is no SNMP data generated regarding the extra threads.
- **NumberOfConfiguredDS:** The number of directory servers configured for this Identity or Access Server.
- **DsFailureTrapTimeSpan:** The amount of time to wait before sending the next failure trap to the same directory server.
- **NumOfSlotsInEventQueue:** The number of slots to be used in the event queue. This parameter must be updated if traps are not detected. However, the default value of five should be adequate for most installations.
- **SleepTimeInMilliSec:** The interval in milliseconds that the Identity or Access Server uses to check whether the SNMP Agent is up and running.

- **semaphore_filepath:** Information about the semaphore created by the Access Server. Semaphores are used for synchronization between the component (the Identity or Access Server) and the SNMP Agent. This information is used to automatically clean up the semaphore if the component fails.
- **semaphore_id:** The Agent semaphore identifier.

Changing these settings affect the memory map file used for SNMP data collection. On Unix, the memory map file is located in

```
/tmp/netpoint/scoreboard/component/process-id.osb
```

On Windows, this file is located in

```
Component_install_dir/oblix/scoreboard/process-id.osb
```

Logging for SNMP

The SNMP Agent supports logging. Once the SNMP Agent is enabled, it is always set to a certain log level. The SNMP logs can assist with troubleshooting. You can configure what is logged and the type of logs to generate in the Agent configuration file. This file resides in

```
SNMP_install_dir/oblix/config/snmp_agent_config_info.xml
```

where *SNMP_install_dir* is the directory where the SNMP Agent was installed.

The *log_level* parameter in the Agent configuration file may have one of the following values:

- **0:** Debug
- **1:** Information
- **2:** Warning
- **3:** Error
- **4:** No logging (turns logging off)

SNMP Messages

The following are SNMP-related messages.

Message:

```
MErrNoConfigFile {Could not find agent configuration file at location (full path to the agent configuration file)}
```

Description: The installation directory is not correct, or the configuration file is not present. Uninstall and reinstall the SNMP Agent.

Message:

```
MLogAgentStarted {Agent successfully started on port SNMP port number}
```

Description: Status message.

Message:

```
MErrAddressInUse {Agent was not able to bind to port port number, address already in use}
```

Description: The SNMP Agent is unable to bind to its configured TCP registration port. Reconfigure the Agent to use another TCP port, or make the port available by stopping the application using the port.

Note: If you change the Agent TCP registration port, you must also specify the new port when enabling SNMP for the Identity or Access Server using the appropriate System Console.

Message:

Agent was not able to bind to specified port, system lacked sufficient buffer space or queue was full.

Description: The SNMP Agent port is unavailable.

Message:

MErrTLUnsupported {Agent was not able to bind to specified port, address family not supported by protocol family}

Description: The specified port does not support SNMP. Configure a different port.

Message:

MErrRetriveIDs {Error: Unable to determine the uid/gid for which this snmp agent is installed.}

Description: The user who tried to start the SNMP Agent does not have the appropriate permissions. The user should start the SNMP Agent as root or as the user who installed the Agent.

Message:

MErrCouldNotSetIDs {Error: You don't have sufficient access rights to run this snmp agent.}

Description: You need to log in with administrative rights to be able to install the SNMP Agent. If you did not do this, the Agent is unable to run.

Message:

MLogAlreadyRunning {Agent is already running with process id (Process identifier of the agent).}

Description: The user is trying to start the Agent when it is already running.

Message:

MErrRegBindFailed {Error: Unable to bind to configured registration port (configured registration port number).}

Description: The SNMP Agent is unable to bind to the port configured on the Oracle Access Manager server configuration page. Specify a different port, as described in ["Enabling and Disabling SNMP Monitoring"](#) on page 12-13.

Message:

MErrRegListenFailed {Error: Unable to start listening on configured registration port (configured registration port number).}

Description: This message is displayed on Windows if the port is already in use by another application.

Message:

```
MErrReadingMsg {Error: reading message sent by component.}
```

Description: The SNMP Agent and the Oracle Access Manager server talk over a TCP connection. If the Agent encounters a malformed message, it logs an error.

Message:

```
MErrNotRegMsg {Error: Agent expects only registration messages on the registration socket.}
```

Description: The Agent only expects registration messages on the TCP connection from a server that connects to it. If it finds that the message is not a registration message, it logs an error.

Message:

```
MErrMissingMmapFilename {Error: Registration message was missing the component scoreboard file name.}
```

Description: The scoreboard file is where the Identity or Access server stores the statistics that are read by the Agent. This name is communicated by the server to the Agent at registration time. If the registration request is missing the file information, this message is logged.

Message:

```
MErrMappingScoreboard {Error: Unable to memory map the scoreboard file (full path to the scoreboard file) registered by component.}
```

Description: This error can occur due to file permission issues, that is, the Agent cannot read or open the scoreboard file.

Message:

```
MErrUnknownComponent {Error: Unknown component type specified in scoreboard file.}
```

Description: The component type is specified in the registration request. The Agent processes information for the Identity Server and Access Server. If the component type is not either of these, this message is logged.

Message:

```
MErrIndexExists {Error: A component has already registered in table (OID for the table for that component) with index (index that is already in use by some other component).}
```

Description: The same instance of a component tried to register again. Each instance of a component is uniquely identified by a key or index by the same SNMP Agent. If another component instance tries to register using the same key or index, this message is logged.

Message:

```
MErrCreatingAgentSemaphore {Error: Unable to create named semaphore (full path to the agent semaphore file) for agent-component event dispatching.}
```

Description: The Agent and the component create one semaphore that is cleaned up at shutdown. In case of unclean shutdown, the semaphores are deleted on the next server/Agent startup. Probable causes are that the system has run out of semaphores or there are permission issues while creating the semaphore.

Message:

`MErrOnSelect {Error: Select() call returned error code (error code returned for the select() call).}`

Description: This is an error code returned directly from the function. This message is used for troubleshooting purposes.

Message:

`MErrOnPoll {Error: Poll() call returned error code (error code returned for the poll() call).}`

Description: This is an error code returned directly from the function. This message is used for troubleshooting purposes.

Message:

`MErrNotDeregMsg {Error: Agent expected a de-registration message on the socket, instead got a message with code (message code for the message received).}`

Description: The Agent only expects a de-registration message from a component once the component has registered.

Message:

`MErrRemovingComponent {Error: Component with table oid (OID for the table for that component) and index (index which identifies the component in that table) could not be removed.}`

Description: The component has already de-registered, and there has been another request to remove it.

Message:

`MErrMissingEvent {Error: Unable to retrieve event from component with table oid (OID for the table for that component) and index (index which identifies the component in that table).}`

Description: The component sends an event to the Agent, and the Agent converts this to an appropriate trap. The component also signals the Agent that it has dispatched an event. If the Agent is signaled but it does not find an event, this message is logged.

Message:

`MErrMissingTrapData {Error: Missing trap meta-data for component from table oid (OID for the table) and index (index that identifies the component in that table) with event (event identifier supplied by the component).}`

Description: The component did not deliver the complete data for an event.

Message:

`MLogMappedScoreboard {Mapped scoreboard file (full path to the scoreboard file) for a component.}`

Description: This is a status message.

Message:

`MLogComponentRegistered {Component registered with table oid (OID for the table) and index (index that identifies the component).}`

Description: This is a status message.

Message:

`MLogComponentDeregistered {Component with table oid (OID for the table) and index`

(index that identifies the component) de-registered.}

Description: This is a status message.

Message:

MLogComponentFailed {Component with table oid (OID for the table) and index (index that identifies the component) failed.}

Description: This is a status message indicating that the Oracle Access Manager component did not deregister properly. This action is treated as a component failure by the SNMP Agent.

Message:

MLogSentTrap {Sent trap with trap oid (OID for the trap sent) for component with table oid (OID for the component table) and index (index that identifies the component in the table).}

Description: This is a status message.

Message:

MLogSemCleanup {Found left-over semaphore from previous run with key (key for the stale left-over semaphore) and file path (file path for the stale left-over semaphore), successfully cleaned up the semaphore.}

Description: Status message. The Agent and the component create one semaphore that is cleaned up at shutdown. In case of unclean shutdown, the semaphores are deleted on the next server/Agent startup.

Message:

MErrSemCleanup {Found left-over semaphore with key (key for the stale left-over semaphore) and file path (file path for the stale left-over semaphore). Encountered errors while removing it.}

Description: The Agent and the component create one semaphore that is cleaned up at shutdown. In case of unclean shutdown, the semaphores are deleted on the next server/Agent startup. This message would be logged if the Agent encountered errors while cleaning up the semaphores from a previous run. There may be permission issues.

Message:

MSBCreateFailed {Access Server: Could not create scoreboard file (full path for the file) with size file size.}

Description: The probable cause for this message is the system could not create the file due to insufficient space.

Message:

MCreateSemFailed {Access Server: Could not create event queue semaphore with path full path.}

Description: The Agent and the component create one semaphore that is cleaned up at shutdown. In case of unclean shutdown, the semaphores are deleted on the next server/Agent startup. This message is generated when the system has run out of semaphores or there are permission issues when creating the semaphore. Try increasing the semaphore limit on the machine.

Message:

```
MSBDirCreateFailed {Access Server: Could not create scoreboard file file name.}
```

Description: The system could not create the required directory for the scoreboard file, probably due to insufficient permissions.

Discrepancies Between Netstat and SNMP Values

When using the netstat command, the value returned for this command may not always match the information collected for the MIB variables:

```
aaaDirectoryServerNoOfLiveConnections
coreidDirectoryServerNoOfLiveConnections
```

[Table 12–11](#) explains the reason for this discrepancy and the chain of events that takes place

Table 12–11 Netstat Values and Number of Live Connections Displayed

Event	Number Of Live Connections	Netstat Value	Comments
Server startup followed by directory server access.	5	5	
The directory server goes down.	5	0	Oracle Access Manager does not update the counter unless it receives a request.
Oracle Access Manager tries to use a connection for accessing the directory server for servicing a request.	4	0	The directory server access returns an error because the directory server is down. The connection is marked as down and the NumberOfLiveConnections is decreased by one.
Directory server is restarted and Oracle Access Manager tries to reestablish the broken connection.	5	1	When a new connection is formed, the NumberOfLiveConnections is incremented by one. The mismatch between NumberOfLiveConnections and the Netstat value will be seen until all of the remaining four connections are marked as down and new connections are formed. The status for the remaining four connections will not be visible unless they are used.
Oracle Access Manager reestablishes all of the broken connections.	5	1	The netstat value matches NumberOfLiveConnections only after all connections are formed.

Configuring the Shutdown Interval

To ensure that an Identity or Access component can perform a clean shutdown, enough time must be allocated to ensure that all cleanup activities can be completed. For the Identity Server, the Access Server, and the SNMP Agent, the shutdown_time parameter specifies the time allocated for the server to attempt a clean shutdown. This parameter is located in globalparams.xml. The default shutdown time is five seconds.

The globalparams file location is as follows:

For Access Server:

```
AccessServer_install_dir/access/oblix/apps/common/bin/globalparams.xml
```

For Identity Server:

```
Identity_install_dir/identity/oblix/apps/common/bin/globalparams.xml
```


The default shutdown time appears as follows in these files:

```
shutdown_time:5
```

You can change the value to any time, specified in seconds.

Part IV

Appendices

The information here is provided to help you configure Oracle Access Manager with Microsoft Active Directory and implement .NET features. An index is also provided.

Part IV contains the following information:

- [Appendix A, "Deploying with Active Directory"](#)
- [Appendix B, "Configuring for ADSI"](#)
- [Appendix C, "Configuring for Active Directory with LDAP"](#)
- [Appendix D, "Implementing .NET Features"](#)
- [Appendix E, "Oracle Access Manager Parameter Files"](#)
- [Appendix F, "Troubleshooting Oracle Access Manager"](#)

Deploying with Active Directory

After you complete the activities in the *Oracle Access Manager Installation Guide* to install and set up Oracle Access Manager with Active Directory, you can complete activities here to configure these components for daily use and maintenance.

This appendix contains the following topics:

- [Setting Up Directory Profiles and Searchbases](#)
- [Authentication and Authorization with Active Directory](#)
- [Configuring the credential_mapping Plug-In](#)
- [Configuring Single Sign-On for Use with Active Directory](#)
- [About Search Filters](#)
- [About the Length of the SAMAccountName](#)
- [Configuring for .NET Features](#)
- [Troubleshooting](#)
- [Microsoft Resources](#)

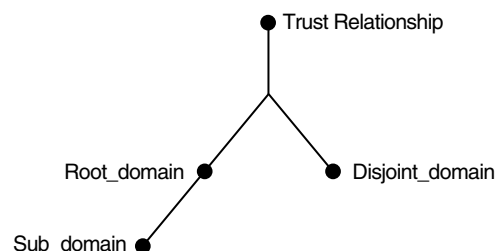
For additional information and procedures, see the *Oracle Access Manager Installation Guide*. See also:

- [Appendix B, "Configuring for ADSI"](#)
- [Appendix C, "Configuring for Active Directory with LDAP"](#)

Setting Up Directory Profiles and Searchbases

The Active Directory forest shown in [Figure A-1](#) includes three domains: Root_domain, Sub_domain, and Disjoint_domain. The configuration in [Figure A-1](#) appears in the discussions that follow.

Figure A-1 Three Domains in a Single Active Directory Forest



When you have finished installation and set up for Active Directory, as described in the *Oracle Access Manager Installation Guide*, you are ready to complete the following tasks.

- [Defining Directory Server Profiles for Remaining Domains](#)
- [Setting Up Disjoint Searchbases](#)
- [Configuring Group-Search Read Operations \(Optional\)](#)

Note: When deploying to multiple forests, all of the forests must use the same schema. This enables you to configure multiple forests in the same manner as other disjoint LDAP domains. See "[Working With Multiple Directory Searchbases](#)" on page 7-34 for details.

Defining Directory Server Profiles for Remaining Domains

A default directory server profile is created automatically each time you install the Identity Server and specify new directory server connection information. The directory server profile contains connection information for one or more directory servers that share the same namespace and operational requirements for read, write, search, and so on. The connection information includes a name, a domain or namespace to which it applies, a directory type, and a set of operations.

After installation, you can use the Identity System Console to modify the directory server profiles as outlined in the following procedure. When you finish the steps in the following procedure, you may set up the Disjoint Searchbases.

For more information, see "[Managing Directory Server Profiles](#)" on page 7-19.

Note: The default directory server profile is created for only your Root_domain. You must set up directory profiles for the remaining domains in your installation: for example, Disjoint_domain and Sub_domain.

To set up additional directory server profiles

1. Navigate to the Identity System Console.
`http://hostname:port/identity/Oblis`
2. Navigate to the directory server profile: Identity System Console, System Admin, System Configuration, Configure Directory Options, link.
3. Add the profile for the Disjoint_domain, if you have one, as described in "[Managing Directory Server Profiles](#)" on page 7-19.
4. Add the profile for the Sub_domain.
5. Rename the Default Directory Profile *if* the default name generated during Identity System setup is not meaningful to you.
6. Set up disjoint searchbases, as described next.

Setting Up Disjoint Searchbases

After the domains are configured, you need to add a disjoint searchbase for the Disjoint_domain and verify that there is no value in the Tab Searchbase field.

Note: Depending on how you configured the Root_domain, you may want to add a disjoint searchbase for the Sub_Domain.

To add a disjoint searchbase for the Disjoint_domain

1. Navigate to the Identity System Console.
`http://hostname:port/identity/oblix`
2. Navigate to and select the Directory Server link: Identity System Console, System Admin, System Configuration, Configure Directory Options, link.
3. Add a disjoint searchbase for the Disjoint_domain and click Save.
4. Navigate to and select the Configure Tab function in the User Manager: Identity System Console, User Manager Configuration, Configure Tab.
5. Select a link on the Configure Tab page.
6. Confirm there is no value in the Tab Searchbase field.
7. Repeat these steps for the Sub_domain, if you have one.

About Deleting a Disjoint Searchbase

If there are searchbase policies for the disjoint searchbase, a user who has this searchbase on this node is able to create a filter with Query Builder whose base is this searchbase. It is advisable to remove all access control policies for this disjoint searchbase before deleting it.

If you remove a disjoint searchbase, you must disable all the database agents that use this searchbase.

Configuring Group-Search Read Operations (Optional)

Active Directory uses incremental retrieval of group members. This means that the Identity System must perform multiple reads to get the complete set of group members.

For Active Directory on Windows 2000, the maximum number of members that can be retrieved with one read is 1000. Unless you change the parameter, the Identity System uses a default value of 1000. For Active Directory on Windows .NET Server 2003 the maximum is 1500. The parameter `maxForRangedMemberRetrieval` located in `globalparams.xml` contains the maximum value that the Identity System uses.

Note: The notation *install_dir* refers to the directory where you installed the named component. For example, *IdentityServer_install_dir* refers to the directory where you installed the Identity Server.

To configure group-search read operations on Windows 2003

1. Locate the `globalparams.xml` file in `\IdentityServer_install_dir\identity\oblix\apps\common\bin\globalparams.xml`.
2. Add the `maxForRangedMemberRetrieval` entry with a value of 1500.

For example:

```
<SimpleList > <NameValPair ParamName="maxForRangedMemberRetrieval"
Value="1500" /> </SimpleList>
```

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

Authentication and Authorization with Active Directory

Two-forest configurations were introduced in the *Oracle Access Manager Installation Guide*. After installation, configuring the Access System for Active Directory can include setting up authentication and authorization in a parent-child domain.

This section contains the following topics:

- [Parent-Child Authentication](#)
- [Parent-Child Authorization](#)
- [ObMyGroups Action Attribute](#)

Parent-Child Authentication

In this case, you need to use the Access System's credential mapping plug-in to authenticate users against both the parent and child domains. This plug-in obtains the user's DN.

For example, suppose you have two domains, `foo.goodwill.oracle.com` and `goodwill.oracle.com`. In the Identity System, you have two directory profiles, one for `foo.goodwill.oracle.com` with a display name of `foo` and another for `goodwill.oracle.com` with a display name of `goodwill`:

```
Foo.goodwill.Oracle.com:
DisplayName=foo
Searchbase = dc=foo,dc=goodwill,dc=Oracle,dc=com
User:Alice
```

```
Goodwill.Oracle.com
DisplayName=goodwill
Searchbase=dc=goodwill,dc=Oracle,dc=com
User:Bob
```

Also suppose you are using a Oracle Access and Identity authentication mechanism that uses the filter in the `credential_mapping` plug-in shown in the previous example. When the user tries to log in to the Access System, a Basic Over LDAP dialog box appears.



The domain is the part of the login ID entered before the "\". From the domain name, the Access System can tell which searchbase to use to identify this user. When Alice logs in, she must specify a domain name of `foo`, and when Bob logs in he must specify a domain name of `goodwill`. Both users will be authenticated.

Note: To access a resource that is protected by a Oracle Access and Identity authentication scheme for an Active Directory forest, the user needs to enter the "*domainname\username*" as the user name in the Authentication dialog box. This *domainname* should be the display name for the DB profile created for the Access Server that is used to perform authentication for this user.

Parent-Child Authorization

You can define separate LDAP rules for each domain. For example, if you want all users who have the title of Manager to access a resource, you need to specify two LDAP rules, one for each domain. An example of these rules:

```
ldap:///dc=goodwill,dc=Oracle,dc=com??sub?(&(title=Manager)
(objectclass=user))
```

```
ldap:///dc=foo,dc=goodwill,dc=oracle,dc=com??sub?(&
(title=Manager)(objectclass=user))
```

Using these rules, managers who are in both foo and goodwill can be authorized.

ObMyGroups Action Attribute

You can use the ObMyGroups action attribute to return in a header variable all of the groups to which a user belongs. Specifying ObMyGroups in the attribute name uses the searchbase configured for the Access System. Access Server does not impose any limit on the number of groups that are returned. The returned groups are only limited by any size limit configured for the directory.

The Access System supports only one searchbase. Therefore, if the user chooses goodwill.Oracle.com as the product searchbase, then ObMyGroups results in a search for groups under goodwill.oracle.com. In this case, the Access System cannot follow referrals, and since the Access System does not have multiple searchbase capability, groups from foo.goodwill.oracle.com cannot be returned.

You can specify ObMyGroups with an LDAP URL. In this case, the searchbase is picked up from the LDAP URL. However, you can only associate one attribute with one header variable, so if you have two domains you need at least two header variables to obtain all of the groups a user belongs to.

For example, suppose you have two domains: dc=goodwill,dc=oracle,dc=com and dc=dilbert,dc=goodwill,dc=oracle,dc=com. To obtain groups from both these searchbases you must define two separate header variables, one for each domain, as shown in [Table A-1](#).

Table A-1 ObMyGroups with LDAP URLs for Two Domains

Type	Name	Return
headervar	HTTP_PARENT_GROUP	"obmygroups:ldap:///dc=goodwill,dc=Oracle,dc=com??sub?(group_type=role)"
headervar	HTTP_CHILD_GROUP	"obmygroups:ldap:///dc=dilbert,dc=goodwill,dc=Oracle,dc=com??sub?(group_type=role)"

- In HTTP_PARENT_GROUP: all the groups in "dc=goodwill,dc=Oracle,dc=com" tree for which the logged-in user is a member and the group_type is role are returned.

- In HTTP_CHILD_GROUP: all the groups in "dc=dilbert,dc=goodwill,dc=Oracle,dc=com" tree for which the logged-in user is a member and the group_type is role are returned.

The following procedures guide as you configure the credential_mapping authentication plug-in for Active Directory and set up SSO, if needed.

Configuring the credential_mapping Plug-In

Each policy domain requires an authentication scheme. Each authentication challenge method is supported by one or more plug-ins. For more information about plug-ins for authentication challenge methods, see the *Oracle Access Manager Access System Administration Guide*.

The credential_mapping plug-in maps the user's user ID to a valid distinguished name (DN) in the directory. You can configure the attribute to which the user ID is mapped. The most common attribute it is mapped to is uid. However, it is possible for a customer to map the user ID to a profile attribute other than uid by changing the obMappingFilter parameter.

The obmappingbase defines the user searchbase. In a single domain, the mapping base must be explicitly defined in the obMappingBase parameter of the credential_mapping plug-in. For example,

```
ou=company,dc=mydomain,dc=Oracle,dc=com).
```

With an Active Directory forest, the user needs to provide the domain plus the user ID during the login to validate the user credential against the specified domain. In this case, the mapping base should be set to obMappingBase="%domain%". The template for defining credential mapping for Oracle Access and Identity in an Active Directory forest should be

```
obmappingbase="%domain%", obmappingfilter=(&(objectclass=user)
(sam accountname=%userid%))", obdomain="domain"
```

The domain information is maintained in the DB profiles in the Identity System Console. Be sure to create a DB profile for each domain. The login name for a multi-domain forest is the display name from the Access Server DB profile.

To configure the credential_mapping plug-in

1. Create a policy domain in the Policy Manager, as usual.

Note: There is currently an upper limit of 350 policy domains when the Access System is deployed with Active Directory.

2. Navigate to the Authentication Management Plug-In page: Access System Console, Access System Configuration, Authentication Management, *link*, Plug-Ins.

where *link* is the name of the authentication scheme you want to alter.

3. Configure the credential_mapping plug-in for Active Directory.

For example:

- For Form Based:

```
obmappingbase="%domain%", obmappingfilter=(&(objectclass=user) (samaccountname=%l
ogin%))
```

- For Oracle Access and Identity:

```
obmappingbase="%domain%",obmappingfilter=(&(objectclass=user)(samaccountname=%u
serid%))", obdomain="domain"
```

Note: To access a resource that is protected by an Oracle Access and Identity authentication scheme for an Active Directory forest, the user needs to enter the *domainname\username* as the user name in the Authentication dialog box. This *domainname* should be the display name for the DB profile created for the Access Server used to perform authentication for this user.

Configuring Single Sign-On for Use with Active Directory

You may want to configure single sign-on for the Identity or Access System, as described in the following procedure. For more information, see about protecting resources with policy domains and configuring single sign-on, see the *Oracle Access Manager Access System Administration Guide*.

To configure single sign-on with the Identity or the Access System

1. Change actions in the policy domain authorization rules that need to pass the header variable ObUniqueId (rather than uid) HTTP_OBLIX_UID.

Note: There is currently an upper limit of 350 policy domains with Active Directory. See "[Troubleshooting](#)" on page A-9 for details.

2. On the Web server, modify the value of the WhichAttrIsLogin parameter to ObUniqueId in the following files:

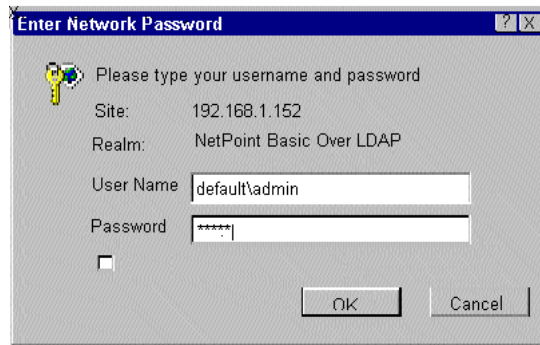
```
\IdentityServer_install_dir\identity\oblix\apps\common\bin\globalparams.xml
\PolicyManager_install_dir\access\oblix\apps\common\bin\globalparams.xml
```

```
WhichAttrIsLogin:ObUniqueId
```

The Configure Directory Server profiles page shows the associated directory profile, which specifies a number of attributes. For example:

```
Machine:
Port Number:
Root DN:
Root Password:
Searchbase:
Configuration base:
Directory Server Security Mode:
Disjoint Searchbase:
ADSI enabled: Yes
```

Following is resulting basic login window related to the single sign-on.



If you are configuring for an Active Directory forest, the domain the user belongs to is determined by the directory profiles configured in the Identity System and used by the Access System. These directory profiles can be enabled or disabled through the Configure Directory Server Profile page accessible from the Identity System Console, Configure Directory Options function.

- If a directory profile is disabled and the user enters that domain name during login through the Access System, then the user is not allowed access.
- If a directory profile is enabled and user enters that name as the domain name, then the user is allowed access.

However, if a user is already authenticated and has a valid session token, after which the directory profile is disabled, then the user is allowed access based on the authorization rules, and so forth. The directory profile state (enabled/disabled) does not take affect during authorization. Only authentication honors the state of the profile.

About Search Filters

Active Directory does not invoke indexed searches when the filter contains constraints that evaluate to *which contains* --filters such as `cn=*Smith`, where the searched for value contains *Smith*. For indexed searches the following constraints are valid:

- Equals (`cn=Smith`)
- Begins with (`cn=Smith*`)

The myGroups tab on the Group Manager may take a long time to evaluate the result if the dynamic groups options is enabled and the dynamic filters specified contain the *which contains* search filter.

To prevent users from using the *which contains* search filter, restrict the available filters by modifying the catalog files. In the following directory:

`Component_install_dir/identity|access/oblix/app/bin/application`

where `Component_install_dir` is the directory where the component is installed and `identity|access` represents either the Identity or Access system, respectively

There are several `xxxparams.xml` files. In these files you can specify the type of valid filters allowed under `vallist - ObEnhanceSearchList`.

See also, "[Resolving Ambiguous Names](#)" on page D-1.

About the Length of the SAMAccountName

The attribute for the Security Access Manager account name (SAMAccountName) in the Active Directory schema is used for backward compatibility with versions of Windows prior to Windows 2000. If you do not need to support these older versions, you can run Active Directory in native mode.

If you need to run Active Directory in mixed mode, Active Directory limits the number of characters that can be specified as the value for the SAMAccountName. By default, Oracle Access Manager accommodates mixed mode and limits the length of SAMAccountName strings to 20 characters.

If you are running Active Directory in native mode, you should edit the value of the samAccountNameLength parameter in the globalparams.xml file. This file is located in the following directory:

Component_install_dir/apps/common/bin/globalparams.xml

Where *Component_install_dir* is the name of the directory where the Oracle Access Manager component is installed. You would edit the value of the following entry in this file:

```
<SimpleList>
  <NameValPair
    ParamName="samAccountNameLength"
    Value="20">
  </NameValPair>
</SimpleList>
```

For more information on globalparams.xml, see the *Oracle Access Manager Customization Guide*.

The user receives an error when defining a person or group with a name that exceeds the limit set for samAccountName in globalparams.xml. The default message is states that this parameter should not exceed 20 characters. The message tag is "MSamAccountNameExceeds20Error"

If you change the value of this parameter, you should also change the error message in the message catalog, globalmsg.xml.

Configuring for .NET Features

Oracle Access Manager provides support for .NET features with Windows Server 2003. See ["Implementing .NET Features"](#) on page D-1 for details.

For details about the following topics, see the *Oracle Access Manager Integration Guide*.

- Integrating with the Authorization Manager
- Integrating with Smart Card Authentication
- Integrating with the Security Connector for ASP.NET

Troubleshooting

For information on troubleshooting, see [Troubleshooting Oracle Access Manager](#) on page F-1.

Microsoft Resources

Active Directory Home Page

<http://www.microsoft.com/windows2000/technologies/directory/ad/default.asp>

ADSI Overview

http://www.microsoft.com/windows2000/techinfo/howitworks/active_directory/adsilinks.asp

Active Directory Programmers Page

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp?frame=true

ADSI Programmers Page

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp?frame=true

Configuring for ADSI

Both the Identity System and the Access System provide support for Active Directory Services Interface (ADSI) client applications. This chapter summarizes requirements and procedures when you are running Oracle Access Manager with Active Directory forests and the Active Directory Services Interface (ADSI).

This appendix contains the following sections:

- [About ADSI with Oracle Access Manager](#)
- [Identity System ADSI Configurations](#)
- [Access System ADSI Configurations](#)
- [Configuring ADSI for the Identity System](#)
- [Enabling ADSI for a Default Directory Profile](#)
- [Enabling ADSI for Other Directory Profiles](#)
- [Configuring ADSI for the Access System](#)
- [Changing the pageSize Parameter](#)

For additional information and procedures, see the *Oracle Access Manager Installation Guide*.

About ADSI with Oracle Access Manager

Active Directory runs on Windows® 2000 and Windows Server 2003 domain controllers. Client applications using ADSI may be written and run on other windows platforms.

ADSI is a set of COM interfaces that enable tight integration with Active Directory. For example, ADSI:

- Abstracts the capabilities of different directory services from multiple vendors to present a single interface for managing network resources.
- Allows administrators and developers to manage the resources in a directory service, regardless of which network environment contains the resource.
- Enables administrators to automate common tasks such as adding users and groups, managing printers, and setting permissions on network resources.

Important: Enabling ADSI allows Oracle Access Manager to take advantage of Active Directory's implicit failover and password-change capabilities.

With ADSI, Oracle Access Manager components do not have to bind to a specific host and port to access Active Directory data. Instead, ADSI allows Oracle Access Manager components to connect to the nearest available domain controller for accessing any user, group, or Oracle Access Manager configuration information.

As described in the *Oracle Access Manager Installation Guide*, the credentials for ADSI are used to bind to the entire forest. A forest can contain multiple Active Directory hosts. When user data and configuration data are stored on separate Active Directory hosts in separate forests, you cannot connect to these simultaneously using ADSI.

ADSI does not require specific host and port numbers for different domains in the forest. ADSI connects to Active Directory hosts using an LDAP URL like the following:

```
LDAP://domain.oblix.com/ou=oblix,dc=domain,dc=oblix,dc=com
```

For details about enabling ADSI during installation, see the *Oracle Access Manager Installation Guide*.

Recommendation

Active Directory replicates the entire tree structure. Due to potential replication delays, Oracle recommends you not replicate the directory tree containing Oracle configuration data. Changes to configuration data may not be immediately available. For example, a change made to a user's access permissions in the Policy Manager may not be available to the Access Server if they are talking to different domain controllers.

If you must replicate the Oblix tree, modify the replication frequency between the domain controllers on Active Directory.

Identity System ADSI Configurations

Oracle Access Manager supports a flexible combination of ADSI and LDAP that relates to your choice of authentication options and binding options.

Note: SSL is not required with ADSI and Oracle Access Manager. However, your business may require SSL for other reasons. For example, directory binds are in clear text and SSL is not automatically provided.

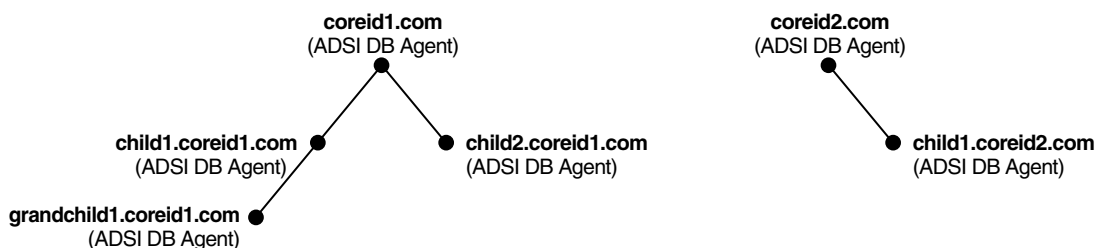
This section includes the following topics:

- [Pure ADSI with ADSI Authentication](#)
- [Mixed ADSI with LDAP Authentication](#)
- [Oracle Access Manager ADSI Configuration Files](#)
- [Bind Mechanisms for the Identity Server](#)

Pure ADSI with ADSI Authentication

With a pure ADSI setup, a single ADSI database agent is created during Identity System setup for the primary domain controller in your Active Directory tree. You must add additional agents for each child domain.

Additionally, if you have a noncontiguous forest of domain trees, you need to associate a separate ADSI database agent for each primary domain controller as shown in [Figure B-1](#).

Figure B–1 Noncontiguous Forest with ADSI

See ["Managing Directory Server Profiles"](#) on page 7-19 for more information about multiple directory profiles and DB agents.

Mixed ADSI with LDAP Authentication

ADSI authentication may be slower than LDAP. For this reason, you may want to use LDAP for authentication when other operations, such as read, write, and search are handled by ADSI. An ADSI agent must be associated with every domain.

To associate an ADSI agent with every domain

1. Select the "Use LDAP for authentication" check box next to Microsoft Active Directory (using ADSI) on the Create Directory Profile page in the Identity System Console.
2. See ["Managing Directory Server Profiles"](#) on page 7-19 for more information about multiple directory profiles and DB agents.

Repeat as needed.

Note: The Global Catalog is not required with this release.

For more information, see ["Bind Mechanisms for the Access Server"](#) on page B-6 and ["Oracle Access Manager ADSI Configuration Files"](#) on page B-4.

Bind Mechanisms for the Identity Server

ADSI provides several ways for the Identity Server to bind to Active Directory. There is no advantage to any particular method. Instead, it depends on which credentials you want to use. For example:

- **Implicitly:** Using the credential of the current process. This is the default for the Identity Server.

This corresponds to the service logon credentials for the Identity Server. For implicit bind, the useImplicitBind flag in the adsi_params.xml file should be set to 0. See ["Oracle Access Manager ADSI Configuration Files"](#) on page B-4 for details.

Note: You must create an account for the Identity Server to bind to Active Directory.

The account that enables the Identity Server to bind to the Active Directory must be equivalent to the root DN that you specify during setup of the Identity Server. It should have all the administrative privileges for operations that are to be

performed using Oracle Access Manager. In an Active Directory forest, this user should be delegated control over all the other domains in the forest.

- **Explicitly Using the DN of the user:** The useImplicitBind flag in the adsi_params.xml file should be set to 1. The user DN should be specified with the adsiCredential parameter located in the adsi_params.xml file. See ["Oracle Access Manager ADSI Configuration Files"](#) on page B-4 for details.
- **Explicitly Using the userPrincipalName:** The useImplicitBind flag in the adsi_params.xml file should be set to 2. The UPN should be specified in the adsiUPN parameter in the adsi_para.xml file. See ["Oracle Access Manager ADSI Configuration Files"](#) on page B-4 for details.

Oracle Access Manager ADSI Configuration Files

ADSI configuration parameters are maintained in two files:

```
\IdentityServer_install_dir\identity\oblix\apps\common\bin\globalparams.xml
\IdentityServer_install_dir\identity\oblix\config\adsi_params.xml
```

where *IdentityServer_install_dir* is the directory in which you installed the Identity Server.

About globalparams

This section shows a sample globalparams.xml file followed by a table of parameter values.

The install program in *\IdentityServer_install_dir\identity\oblix\apps\common\bin\globalparams.xml* creates the adsiEnable parameter and sets its value to true when you enable ADSI for the default directory profile. This parameter refers to a system level directory profile that contains Oracle configuration data.

Note: You must restart the Identity Server after changing any parameters. However, do not change the ADSIEnabled parameter value.

```
<SimpleList>
<NameValPair ParamName="ActiveDirectory" Value="true" />
</SimpleList>
<SimpleList>
<NameValPair ParamName="ADSIEnabled" Value="true" />
</SimpleList>
```

Table B-1 Parameters and Values in globalparams Files

globalparams Parameters	Values
ActiveDirectory	true false True when the Master Administrator selects Active Directory as the directory server type during Identity Server configuration
ADSIEnabled	true false True when the Master Administrator enables ADSI during Identity Server configuration

About adsi_params

This section shows a sample adsi_params.xml file followed by a table of parameter values. By default, adsi_params.xml includes a value for the adsiCredential parameter and the password, as shown in the following example. This enables you to change the bind mechanism to be Explicit after initial setup.

The adsiPassword is encrypted and can only be generated by Oracle Access Manager during setup. The following is an example of this file:

```
<?xml version="1.0" ?>
- <ParamsCtlg xmlns="http://www.oblix.com" CtlgName="adsi_params">
- <CompoundList ListName="adsi_params">
- <ValNameList ListName="adsi_params">
  <NameValPair ParamName="sizeLimit" Value="0" />
  <NameValPair ParamName="timeLimit" Value="0" />
  <NameValPair ParamName="pageSize" Value="100" />
  <NameValPair ParamName="useImplicitBind" Value="0" />
  <NameValPair ParamName="adsiCredential"
Value="cn=Administrator,cn=users,dc=goodwill,dc=oblix,dc=com" />
  <NameValPair ParamName="adsiPassword" Value="0243455B5B5F5C4C5651595D41" />
  <NameValPair ParamName="useGCForAuthn" Value="false" />
  <NameValPair ParamName="encryption" Value="false" />
  <NameValPair ParamName="asynchronousSearch" Value="true" />
  <NameValPair ParamName="useDNSPrefixedLDAPPaths" Value="false" />
</ValNameList>
</CompoundList>
..</ParamsCtlg>
```

By default, encryption is set to false in adsi_params.xml. If you set it to true when running in open mode and restart the Identity Server, the Identity Server will not work.

Note: You must restart the Identity Server after changing any parameters.

Table B-2 describes parameters and values within the adsi_params files, next.

Table B-2 Parameters and Values in adsi_params Files

adsi_params Parameters	Values
sizeLimit	Integer value that limits the number of query results returned for authentication.
timeLimit	Integer value that limits the number of seconds before a query times out.
pageSize	Page size of results that ADSI request from the server.
useImplicitBind	0 = Implicit credentials 1 = Explicit credentials 2 = Use userPrincipalName
adsiCredential	An LDAP specification of a user, such as cn=Administrator,cn=users,dc=myhost,dc=mydomain,dc=com
adsiPassword	An encoded text string representing the LDAP user's password.
useGCForAuthn	true/false False

Table B–2 (Cont.) Parameters and Values in *adsi_params* Files

adsi_params Parameters	Values
asynchronousSearch	true/false By default ADSI is enabled to perform asynchronous searches. If set to false, it does synchronous searches.
adsiUPN	This parameter needs to be added if useImplicitBind is set to 2. The value of the parameter should be the UPN (userPrincipalName) of the user.
chaseReferral	Setting this flag to false turns off LDAP referrals.

Access System ADSI Configurations

Like the Identity System, the Access System provides the support for both ADSI and ADSI with LDAP authentication.

The Access System also supports multiple Active Directory domains and you must perform the steps for enabling ADSI for the default directory profile created during Oracle Access Manager setup.

This section includes the following topics:

- [Pure ADSI with ADSI Authentication](#)
- [Access System ADSI Configuration Files](#)

Pure ADSI with ADSI Authentication

The Access Server authenticates to Active Directory using ADSI. This is the default when you enable ADSI on these components.

- The Policy Manager uses the same authentication mode as the Identity Server. Still, you must enable ADSI for the Policy Manager.
See ["Configuring ADSI for the Access System"](#) on page B-11 for details.
- The Access Server can communicate directly with all directory servers in the forest and no longer requires the Global Catalog for LDAP authentication.

For a list of ADSI installation and setup considerations, see the *Oracle Access Manager Installation Guide* appendix on installing with Active Directory.

Authentication Mechanisms

When users authenticate to Active Directory, the mechanism is the domain controller, which uses respective domain controllers for authentication with ADSI.

See ["Access System ADSI Configuration Files"](#) on page B-7 for details.

Bind Mechanisms for the Access Server

ADSI provides several ways for the Access Server and Policy Manager to bind to Active Directory. There is no advantage to a particular method. Instead, it depends on which credentials you wish to use:

- **Implicitly:** Using the credential of the current process (default for the Access Server).

This corresponds to the service logon credentials for the Access Server. For implicit bind, the useImplicitBind flag in the *adsi_params.xml* file should be set to 0. See ["Access System ADSI Configuration Files"](#) on page B-7 for details.

Note: You need to create an account for the Access Service to bind to Active Directory. This account must be equivalent to the Root DN that you specify during setup of the Access Server. It should have all the administrative privileges for the operations that are to be performed using Oracle Access Manager. In an Active Directory Forest, this user should be delegated control over all the other domains in the forest.

- **Explicitly Using the DN of the User:** The useImplicitBind flag in the adsi_params.xml file should be set to 1.

The user DN should be specified with the adsiCredential parameter located in the adsi_params.xml file. See ["Access System ADSI Configuration Files"](#) on page B-7 for details.

- **Explicitly Using the userPrincipalName:** The useImplicitBind flag in the adsi_params.xml file should be set to 2.

The UPN should be specified in the adsiUPN parameter in the adsi_params.xml file. See ["Access System ADSI Configuration Files"](#) on page B-7 for details.

In a multi-domain Active Directory forest the only supported explicit bind mechanism is userPrincipalName. The Policy Manager supports only this mechanism.

Access System ADSI Configuration Files

Both the Policy Manager and Access Server each have two configuration files for modifying ADSI related parameters. Although the files are maintained in separate locations, and must be modified separately for each component, their contents are the same. The configuration files for the Policy Manager and Access Server are as follows:

```
\PolicyManager_install_dir\access\oblix\apps\common\bin\globalparams.xml
\PolicyManager_install_dir\access\oblix\config\adsi_params.xml
```

```
\AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.xml
\AccessServer_install_dir\access\oblix\config\adsi_params.xml
```

These files are discussed in the following sections.

Policy Manager ADSI Configuration

This section shows a sample global-parameters configuration file, followed by a table of parameter values.

Note: When you install Policy Manager and Access Server, if you do not choose the ADSI option, you do not see the ADSEnabled parameter in globalparams.xml. However, you do still see the useLDAPBind parameter, though it serves no purpose without ADSEnabled.

```
BEGIN:vCompoundList
...
useLDAPBind:false
ADSEnabled:true
ActiveDirectory:true
END:vCompoundList
```

The parameters and their values are described in [Table B-3](#).

Table B-3 Parameters and Values in globalparams Files

globalparams Parameters	Values
useLDAPBind	true false True when the Master Administrator selects "Microsoft Active Directory using LDAP" during Policy Manager configuration. The ADSEnabled flag must be true for this flag to have effect. The default is false.
ADSEnabled	true false True when the Master Administrator enables ADSI during Policy Manager configuration.
ActiveDirectory	true false True when the Master Administrator selects Active Directory as the Directory Server type during Policy Manager configuration.

Access Server ADSI Configuration

This section shows a sample adsi parameters configuration file, followed by [Table B-4](#) of parameter values.

Table B-4 Parameters and Values in adsi_params Files

adsi_params Parameters	Values
sizeLimit	Integer value that limits the number of query results returned for authentication.
timeLimit	Integer value that limits the number of seconds before a query times out.
pageSize	Page size of results that adsi request from the server. The default is 0.
useImplicitBind	0 = Implicit Credentials 1 = Explicit Credentials 2 = Use UserPrincipalName
adsiCredential	An LDAP specification of a user, such as "cn=Administrator,cn=users,dc=myhost,dc=mydomain,dc=com"
adsiPassword	An encoded text string representing the LDAP user's password.
adsiUPN	Text string of UserPrincipalName when use ImplicitBind=2. A UPN string is typically an email address with the format: user@company.com
useGCForAuthn	True/False Change the useGCForAuthentication parameter to false.
asynchronousSearch	True/False By default, ADSI is enabled to perform asynchronous searches. If set to false, it does synchronous searches
asynchronousSearch	adsiUPNThis parameter needs to be added if useImplicitBind is set to 2. The value of the parameter should be the UPN (userPrincipalName) of the user.

Configuring ADSI for the Identity System

There are several tasks involved in configuring ADSI for the Identity System. Details are provided in the *Oracle Access Manager Installation Guide*.

Task overview: Configuring ADSI for the Identity System

- Prepare your Active Directory, as described in your Microsoft documentation and the *Oracle Access Manager Installation Guide*.
- Specify ADSI when you install and set up Oracle Access Manager, as described in the *Oracle Access Manager Installation Guide*.

By default, this creates a pure ADSI configuration in which a single ADSI directory profile (DB agent) enables associated Identity Servers to perform all operations with a primary domain controller in your Active Directory tree using an Implicit Bind.

- Set up Active Directory attributes and enable change-password permissions, as described in the *Oracle Access Manager Installation Guide*.
- Configure the default directory profile, as described in "[Enabling ADSI for a Default Directory Profile](#)" on page B-9.
- Enable ADSI for additional directory profiles, if desired, as discussed under "[Enabling ADSI for Other Directory Profiles](#)" on page B-9.

Enabling ADSI for a Default Directory Profile

The Identity System automatically creates a default directory profile during installation. You can enable ADSI for the default profile during Identity System setup.

The default database agent is automatically assigned a name using the convention default-oid-machine name. You should modify this name to the respective domain name because users must enter this name during authentication.

Enabling ADSI for Other Directory Profiles

If you have a noncontiguous forest of domain trees, you need to associate a separate ADSI database agent for each primary domain controller. Additional directory profiles are configured after Identity System installation and set up, as outlined in the following procedure and described in "[Specifying Identity System Administrators](#)" on page 2-1.

To enable ADSI for additional directory profiles

1. Navigate to the Identity System Console.
`http://hostname:port/identity/oblis`
2. From the Identity System Console, click the System Configuration sub-tab, then click the Directory Profiles link in the left navigation pane.
3. Click the Add button to display the Create Directory Server profile page.

Oracle recommends using the respective domain names as the profile names because users must enter this name during authentication.

4. Enter a name for this directory profile.

You must configure a directory profile for each Domain and Sub_domain controller. For more information, see ["Configuring ADSI for the Identity System"](#) on page B-9.

5. Enter a namespace for this directory profile.

There are multiple choices for the directory type. To use Active Directory without ADSI enabled, you should select Microsoft Active Directory.

Note: You still have the option to enable ADSI or SSL for changing passwords. Also, you can enable LDAP by selecting the secondary check box, Use LDAP for Authentication. When LDAP is enabled, an ADSI DB Agent associates with the primary domain controller. An LDAP agent needs to be configured for whichever Sub_domain controller you want to use to authenticate.

6. Select the appropriate directory type. For example:

Directory Type

☐ COREid Data Anywhere
☐ Microsoft Active Directory Application Mode
☐ Microsoft Active Directory (using ADSI)
☐ Use LDAP for Authentication
☒ Microsoft Active Directory
 AD-Change password using: ☒ ADSI ☐ SSL

7. Select the operations supported for this directory profile, for example:

Dynamic Auxiliary ☐ Yes ☒ No

☒ All Operations

☐ Selected Operations

Operations

Search	☒ Search Entries	☒ Authenticate User
Read	☒ Read Entry	
Write	☒ Create Entry	☒ Modify Entry
	☒ Delete Entry	☒ Change Password

If this is a directory profile configured for a domain controller, select all operations.

- Complete the rest of the directory profile and save it, as usual.

See ["Managing Directory Server Profiles"](#) on page 7-19 for more information about configuring directory profiles. See also ["Managing Directory Server Profiles"](#) on page 7-19 for more information about multiple directory profiles (DB agents).

Configuring ADSI for the Access System

The Policy Manager uses the Identity Server for authentication. Therefore, the login operation uses the same mode (ADSI or LDAP) as the Identity Server it talks to. During Policy Manager setup, by default, you use an Explicit Bind to enable the Policy Manager and Access System Console to perform all operations except authentication in the Active Directory tree.

Note: SSL is not required for ADSI configurations with Oracle Access Manager. However your business may require SSL for other reasons. For example, directory binds are in clear text, and SSL is not automatically provided.

By default, enabling ADSI for the Access Server creates a pure ADSI configuration in which the Access Server performs all operations with a primary domain controller in your Active Directory tree using an Implicit Bind.

Configuring ADSI support in the Access System involves the following tasks.

Task overview: Configure ADSI for the Access System

- Validate your setup, as described in the *Oracle Access Manager Installation Guide* appendix.
- Install and set up the Policy Manager, as described in the *Oracle Access Manager Installation Guide*.
- Install the Access Server and setup ADSI, as described in the *Oracle Access Manager Installation Guide*.
- Install the WebGate, as described in the *Oracle Access Manager Installation Guide*.
- Enable LDAP authentication for the Access Server, if desired, as described in ["Enabling LDAP Authentication for the Access Server"](#) on page B-11.

Enabling LDAP Authentication for the Access Server

ADSI authentication may be slower than LDAP. For that reason, you may wish to use LDAP for authentication while other operations such as authorization and auditing are handled by ADSI.

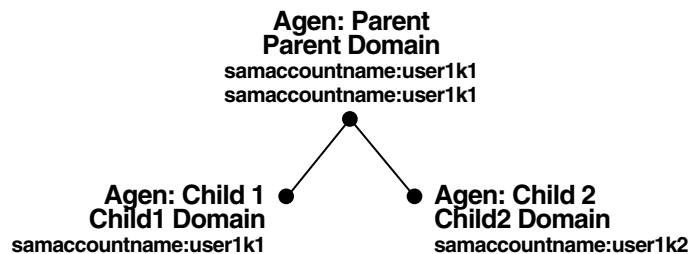
To enable LDAP authentication for the Access Server

1. Open *AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.xml* with a text editor.
2. Change the value of useLDAPBind to true.
3. Save globalparams.xml.
4. Create a copy of ConfigDBfailover.xml located in *AccessServer_install_dir\access\oblix\config* and name it AppDBfailover.xml.
Both files should reside in the same directory.
5. Save.
6. Restart the Access Server.

Changing the pageSize Parameter

Based on your Active Directory forest deployment you may need to change the page-size parameter in the adsi_params file. For example, in [Figure B-2](#) you have a parent-child relationship between your Active Directory domains, and you have user(s) in both the parent and child domain with the same samaccountname.

Figure B-2 Users in Both the Parent and Child Domains



Assume the authentication scheme is Oracle Access and Identity for an Active Directory forest. In this case:

- For user1k1 to log in to the Child1 domain, the user can enter their user ID as Child1\user1k1.
- For user1k2 to log in to the Child2 domain, the user can enter their user ID as Child2\user1k2.

However, if the pageSize parameter is set to 0, for user1k1 from the parent domain to log in entering Parent\user1k1 produces an error: "The credentials Parent\user1k1 used in the login correspond to more than one user profile in the Identity System. The correspondence must be unique."

This is because when the page size is set to 0, ADSI searches the subdomains, therefore finding two users who satisfy the criteria. For user1k1 and user1k2 to log in to the parent domain you need to set the pageSize parameter to a finite value. Oracle recommends using 100.

Troubleshooting

For information on troubleshooting, see "[Troubleshooting Oracle Access Manager](#)" on page F-1.

Configuring for Active Directory with LDAP

This chapter summarizes procedures to set up Oracle Access Manager with Active Directory forests using LDAP as the communication protocol.

This appendix contains the following topics:

- [Overview](#)
- [Setting Up the Policy Manager for LDAP](#)
- [Setting Up the Access Server for LDAP](#)
- [Setting Active Directory Timeouts for LDAP](#)
- [Enabling LDAP Authentication with ADSI](#)

For additional information and procedures, see the *Oracle Access Manager Installation Guide*.

Note: The instructions here apply only if you are using LDAP as the protocol between the Access System and Active Directory. If your environment differs, skip this discussion.

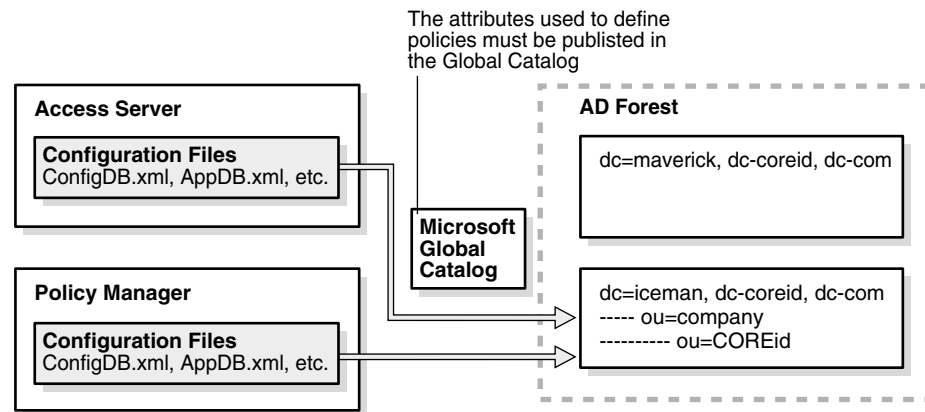
Overview

The Access System supports Active Directory forests with some modifications.

Note: The Microsoft Global Catalog is no longer required for the Access System.

The steps in this section are based on the following example. In this case, the Identity System was configured using the two domains shown in [Figure C-1](#). For example:

- `dc=maverick,dc=oblix,dc=com`
- `dc=iceman,dc=oblix,dc=com`

Figure C-1 Active Directory Forest with Two Domains

Complete the following procedures to set up the Access System for multiple domains using LDAP between the Access System and Active Directory:

- [Setting Up the Policy Manager for LDAP](#)
- [Setting Up the Access Server for LDAP](#)
- [Setting Active Directory Timeouts for LDAP](#)

Note: In the following discussions, *install_dir* refers to the installation directory you specified for the named component. For example, *PolicyManager_install_dir* is the directory where you installed the Policy Manager.

Setting Up the Policy Manager for LDAP

The Oracle Access Manager-related configuration information is located in the `\PolicyManager_install_dir\access\oblix\config\ldap` directory and must be accessed directly. The following are relevant files:

- AppDB.xml
- ConfigDB.xml
- WebResrcDB.xml

Suppose the Identity System was set up as shown earlier with the configuration data in the `dc=iceman,dc=oblix,dc=com` domain.

In this case, the Policy Manager must also be set up for this domain. To accomplish this, you must specify the same configuration DN for the Policy Manager and the Identity Server.

For more information about the configuration DN, see the *Oracle Access Manager Installation Guide*.

To set up the Policy Manager for Active Directory

1. Navigate to the Policy Manager setup page:
`http://hostname:port/access/oblix`
2. Set up the Policy Manager with the same configuration DN as the Identity Server.

For example, against the machine containing the domain:
`dc=iceman,dc=oblix,dc=com`

3. Before starting the Web server, change the port to 3268 (open LDAP) to ensure that users and groups are accessible from both domains.
4. See the appendix ["Deploying with Active Directory"](#) on page A-1 for information about configuring the `credential_mapping` plug-in required in your authentication schemes and setting up SSO.

Setting Up the Access Server for LDAP

This section only applies if you are using LDAP as the protocol between the Access Server and Active Directory.

To set up the Access Server for Active Directory

1. Configure the Access Server using the same configuration DN as the Identity Server.

For example, against the machine containing the domain:
`dc=iceman,dc=oblix,dc=com domain`

2. Make sure Active Directory time out is handled correctly, as described under ["Setting Active Directory Timeouts for LDAP"](#) on page C-3.
3. Create a copy of `ConfigDBfailover.xml` located in `\AccessServer_install_dir\access\oblix\apps\config` and name it `AppDBfailover.xml`.

Both files should reside in the same directory.

Setting Active Directory Timeouts for LDAP

If you are using LDAP, you need to configure timeouts for the Access Server when it is installed against Active Directory.

The Access Server, which runs as a service, opens connections to Active Directory. Active Directory times out idle connections after a period of inactivity, which means that the Access Server can try to access the directory and fail.

If you want to avoid this problem, you need to establish new connections before the Active Directory Idle Session Time is reached. The failover information can be specified when:

- You are prompted to specify failover information at the end of the Access Server installation.
- You reconfigure failover information after installation is complete, using the `configureAAAServer` application in `AccessServer_install_dir/access/oblix/tools/configureAAAServer`, as described in the following procedure.

The following files are created when you use the `ConfigAAAServer.exe` tool to configure failover between a second directory server and the Access Server:

`ConfigDBfailover`

`AppDBfailover`

`Web...DBfailover`

To specify Access Server failover after installation

1. Locate the configureAAAServer application.

```
AccessServer_install_dir\access\oblix\tools\configureAAAServer
```

2. Launch the configureAAAServer application using the following command.

```
configureAAAServer install AS_install_dir
```

3. Answer No, when asked if you want to reconfigure Access Server.
4. Answer Yes, when asked if you want to specify failover information.
5. When asked where different types of data are stored, respond appropriately for your environment.

For example:

- **Separate Directory Servers, Choose Option 8:** If policy and Configuration DN are separate from user data, choose option 8 (Modify Common Parameters) and specify values appropriate for your system.
- **Same Directory Server, Choose Option 4:** If policy, Configuration DN, and user information is on the same directory server, choose option 4 (Modify Common Parameters) and enter values appropriate for your system. For example:

Maximum Connections: 1

Sleep For (seconds): 60

Failover Threshold: 1

Maximum Session Time (seconds): 120

After every Maximum Session Time, a new connection is created to Active Directory, and the old connection is dropped, whether the Access Server was idle or not.

Note: Make sure the Maximum Session Time (in seconds) is less than the Active Directory Idle Timeout (typically less than 600 seconds).

6. Choose the option to quit.
7. When asked if you want to commit the changes, answer Yes.

For more information about failover, see the *Oracle Access Manager Deployment Guide*.

Enabling LDAP Authentication with ADSI

ADSI authentication may be slower than LDAP. For that reason, you may wish to use LDAP for authentication while other operations such as authorization and auditing are handled by ADSI.

To enable LDAP authentication for the Access Server

1. Open globalparams.xml with a text editor.

```
AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.xml
```

2. Change the value of useLDAPBind to true.

3. Save globalparams.xml.
4. Create a copy of ConfigDBfailover.xml located in:
AccessServer_install_dir\access\oblix\config\ldap\ConfigDBfailover.xml
5. Name it AppDBfailover.xml.
Both files should reside in the same directory.
6. Save.

Implementing .NET Features

Oracle Access Manager provides support for .NET features with Windows Server 2003. For details about supported features and their implementation within Oracle Access Manager, see the following topics in this appendix:

- [Resolving Ambiguous Names](#)
- [Configuring for Dynamically Linked Auxiliary Classes](#)
- [Enabling Fast Bind for Access System Authentication](#)
- [Enabling Impersonation](#)
- [Setting Up Integrated Windows Authentication](#)
- [Using Access System Password Management](#)
- [Using Managed Code and Helper Classes](#)
- [Integrating with Authorization Manager Services](#)
- [Integrating with Smart Card Authentication](#)
- [Integrating the Security Connector for ASP.NET](#)
- [Troubleshooting](#)
- [Microsoft Resources](#)

Resolving Ambiguous Names

Active Directory running on Windows Server 2003 provides support for ambiguous name resolution (ANR).

ANR is a search algorithm associated with LDAP clients that must be enabled on both the LDAP client and the LDAP server. ANR allows objects to be bound without complex search filters and is useful when locating objects and attributes that may or may not be known by the client.

In Oracle Access Manager, ANR is a virtual attribute that does not physically exist in the directory server. Oracle Access Manager provides the virtual ANR attribute through the AD_anr.ldif file, which enables Oracle Access Manager to interpret ANR requests, map ANR requests to Boolean functions And and Or that expand to a directory-server filter to broaden the search, and send the query to Active Directory.

Note: The AD_anr.ldif file is included in the Oracle Access Manager schema installation and must be imported manually. See ["Configuring for ANR"](#) on page D-2 for details.

About ANR Attributes, Searches, and Results

By default, the attributes shown in [Table D-1](#) are set for ANR.

Table D-1 ANR Attributes

ANR Attributes
displayName
GivenName
LegacyExchangeDN
msExchMailNickname
name
physicalDeliveryOfficeName
proxyAddress
sAMAccountName
Surname

For a search filter such as (anr=von), the server would return objects that matched any of the previously listed attributes equal to von*. When a space is embedded in the search string, the search is divided at the space and an Or search is also performed on the attributes. The server attempts to perform first/last name processing. When there is only one space, the search divides only at the first space.

For example, if the search filter was (anr=Rob Al), the filter expansion would look like the following.

```
( | (givenName=Rob Al*)
  (sn=Rob Al*)
  (displayName=Rob Al*)
  (legacyExchangeDN=Rob Al*)
  (name=Rob Al*)
  (physicalDeliveryOfficeName=Rob Al*)
  (proxyAddresses=Rob Al*)
  (sAMAccountName=Rob Al*)
  (&(givenName=Rob*) (sn=Al*))
  (&(givenName=Al*) (sn=Rob*))
)
```

Configuring for ANR

The attributes used by ANR are configurable. You can specify other attributes to be included in ANR searches by using the Active Directory Schema Snap-in to check the Ambiguous Name Resolution box for the attribute. You can directly set the searchFlags attribute to 5 in the attributeSchema for the attribute you want to include. To include an attribute to be used for ANR, the attribute must also be indexed.

The following task overview outlines the procedures you must complete to enable ANR within Oracle Access Manager. After you upload the meta-attribute configuration for ANR into the configuration branch in the directory server, the ANR attribute should be configured on the profile page and defined as searchable. Attribute access control can also be configured on the same profile page.

Task overview: Preparing to use ANR during searches

1. Update Oracle configuration data to include the ANR meta-attribute details in the configuration branch of the schema, as described in ["Updating Configuration Data"](#) on page D-3.
2. Make the ANR attribute available to the Oracle Access Manager search function in the Identity Server, as described in ["Configuring ANR in Identity System Panels"](#) on page D-3.
3. Verify Access Control rights, as described in ["Verifying ANR Attribute Access Control"](#) on page D-4.
4. Use ANR-to-Oracle Access Manager authentication and authorization search filters, as described in ["Using ANR in Identity System Searches"](#) on page D-5.

Updating Configuration Data

You first need to update configuration data (Oracle Access Manager configuration data) to include the ANR meta-attribute configuration information in the configuration branch. During this procedure, the following AD_anr.ldif is executed.

```
#File to load ANR meta-attribute configuration to the directory tree.
dn: obattr=anr,obclass=user,OU=Oblix,<domain-dn>
changetype: add
instanceType: 4
distinguishedName:
obattr=anr,obclass=user,OU=Oblix,<domain-dn>
objectClass: oblixmetaattribute
name: anr
obattr: anr
obcardinality: ob_single
obdisplayname: ANR
obdisplaytype: ObDTextS
obsearchable: true
obvisible: true
```

When this procedure is complete, ANR appears as an attribute you can select when configuring Identity System panels.

To update Oracle configuration data

1. Locate the AD_anr.ldif file on the machine hosting the Identity Server:
`\IdentityServer_install_dir\identity\oblix\data.ldap\common\AD_anr.ldif`.
2. Import the AD_anr.ldif file to the configuration directory.

For example:

```
D:\data>ldifde -i -f AD_anr.ldif -a
"cn=administrator,cn=users,dc=name,dc=company,dc=net" password
```

3. Restart the Identity Server.

Configuring ANR in Identity System Panels

After you update the Oracle configuration data with ANR meta-attributes, you are ready to make the ANR attribute available to the Identity System search function on a Tab (panel) and in the list of searchable attributes in the User Manager Selector.

The following procedure guides you through configuring ANR in Identity System panels. For more information, see ["Configuring User, Group, and Organization Manager"](#) on page 4-1.

To configure ANR in Identity System Panels

1. From the Identity System landing page, click the link for the Identity System Console.

If you are already logged in, click the Identity System Console tab.

2. Click the User Manager Configuration sub-tab, then click the Tabs link in the left navigation pane.
3. Click the link for the tab, then click View Object Profile.
4. Click Configure Panels, then click the link for the panel that you want to configure.

A summary appears listing all attributes for the selected panel.

5. Click the Modify button at the bottom of the summary page.

The Modify Panel page appears.

ORACLE Identity Administration

System Configuration | **User Manager Configuration** | Group Manager Configuration | Org Manager Configuration | Common Configuration

Help About

User Manager Group Manager Org Manager

Logged in user: Master F

Configure Header Configure Panels Order Panels

Modify Panel

Panel Label Contact Information

Description Contact Information

Attributes

Telephone Number	Telephone Number
Home Phone	Home Phone
Home Postal Address	Home Postal Address
Mail	Mail
Facsimile Telephone Number	Facsimile Telephone Number

Add

6. Click the Add button, then select ANR from the list in the Attributes column and click Save.

The summary page appears listing all attributes, which should now include ANR.

Next you need to confirm that ANR is a searchable attribute that will appear in the Query Builder's search criteria list.

7. From the Identity System Console, click the User Manager Configuration sub-tab, then click the Tabs link in the left navigation pane.
8. Click the link for the tab.
9. Click the View Search Attributes button at the bottom of the page.

A list of all search attributes appears.

10. Confirm that ANR is in the list. For example:
11. Restart the Identity Server.

Verifying ANR Attribute Access Control

By default, the attribute has read rights. The ANR attribute must not have modify rights. The following procedure shows the Access Control rights for the ANR attribute. See ["Setting and Modifying LDAP Attribute Permissions"](#) on page 4-33 for details.

To verify ANR attribute access control

1. From the Identity System landing page, click the link for the User Manager.
If you are already logged in to the Identity System, click the tab for the User Manager application.
2. Click the Configuration sub-tab, then click the link for Attribute Access Control.
3. Select ANR from the Attribute list, then verify that it has read rights only.
You are ready to use ANR in Identity System searches.

Using ANR in Identity System Searches

When a user invokes the User Manager, they can choose ANR from the search criteria list to perform a directory search.

To use ANR in a search

1. From the Identity System landing page, click the link for the User Manager.
If you are already logged in to the Identity System, click the tab for the User Manager application.
2. Select ANR from the Search list, define other search criteria, then enter your condition.
3. Click Go and check your results.

Configuring for Dynamically Linked Auxiliary Classes

A structural object class can stand on its own and contains basic attributes required for use within Identity System applications. Structural object class examples include person and groupOfNames. The person object class may contain attributes such as name, department, employee ID, and email address. A structural object class must be assigned when you create a tab within an Identity System application.

Auxiliary object classes are mix-in classes that can be added to any structural class. You use an auxiliary object class to add a set of related attributes to an entry that already belongs to a structural class. Items such as a billing address, a challenge phrase, a response to a challenge phrase, and so on may be useful for definition in an auxiliary object class.

With Windows Server 2000, Active Directory supported only statically linked auxiliary classes. A statically-linked auxiliary class is one that is included in the auxiliaryClass or systemAuxiliaryClass attribute of an object class's classSchema definition in the schema. It is part of every instance of the class with which it is associated. Using statically-linked auxiliary classes is the default with Oracle Access Manager installed with Active Directory. All other directories support only dynamically linked auxiliary object classes.

With a Windows 2003 Server, Active Directory and Oracle Access Manager support dynamically linked auxiliary classes. With the schema defined for a particular user, group, or organization, dynamically linked auxiliary classes enable you to store additional attributes with an individual object without the forest-wide impact of extending the schema definition for an entire class. Dynamically linked auxiliary class attributes are mixed in only at runtime.

For example, you can use dynamic linking to attach a sales-specific auxiliary class to the user objects of sales people and other department-specific auxiliary classes to the

user objects of employees in other departments. Or you may want to convert a basic group to a mail group by adding specific attributes dynamically.

Task overview: Setting up for dynamic auxiliary classes

1. Install and set up Oracle Access Manager with dynamic-auxiliary classes enabled, as described in the *Oracle Access Manager Installation Guide*.
2. Specify additional structural object classes for the Organization Manager, as described in ["About Object Classes"](#) on page 3-1.
3. Configure attributes, as described in ["About Object Class Attributes"](#) on page 3-10.
4. Configure User, Group, and Organization application tabs, as described in ["Configuring Tabs"](#) on page 4-2.
5. Configure User, Group, and Organization profile pages, as described in ["Configuring Tab Profile Pages and Panels"](#) on page 4-12.
6. Define workflows, as described in ["Chaining Identity Functions Into Workflows"](#) on page 5-1.
7. Specify additional auxiliary object classes, as described in ["Adding Attributes Dynamically"](#) on page D-6.

Adding Attributes Dynamically

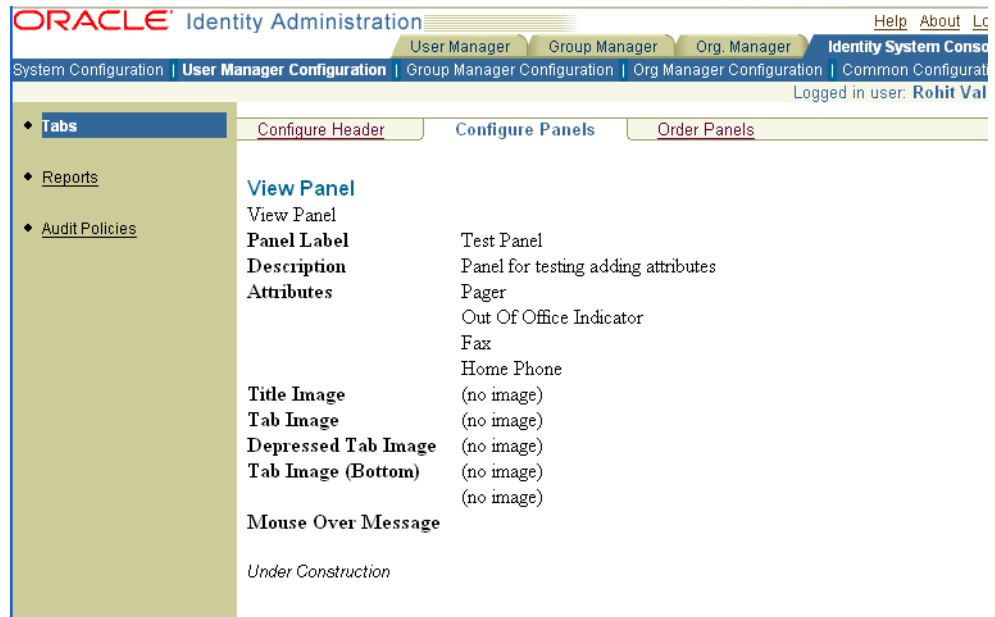
The following procedure provides an example only and assumes that you have created a Tab and Panel in the User Manager. Here you will add desired auxiliary attributes dynamically.

Note: This is only an example. You may be working in the Group Manager or Organization Manager. See also, ["Adding Attributes for a Group"](#) on page D-7.

To specify additional auxiliary object classes in the User Manager

1. From the Identity System Console, click User Manager Configuration, then click Tabs in the left navigation pane.
2. Click the link for the tab.
3. Click the View Object Profile button, then click the Configure Panels link.
4. Click the link for the panel that you want to modify.
5. Click the Modify button to display the Modify Panel page.
6. Click the Add button, select one or more attributes from the list, then click Save.

The View Panel page appears with the attributes you added.



The entry in the directory server has changed, and the new attributes are included.

Adding Attributes for a Group

The example in this procedure dynamically converts a single basic group to a mail group by adding attributes, such as:

Attribute 1	Attribute 2	Attribute 3
MailAlternateAddress	Mailhost	MailRoutingAddress

This example assumes that you have created a Group Panel and a workflow to create a Mail Group. Now you add desired attributes dynamically. This is only an example. You may be working in the User Manager or Organization Manager. See also "[Adding Attributes Dynamically](#)" on page D-6.

To add attributes to a Group Profile panel

- From the Identity System landing page, click the link for the Identity System Console.
If you are already logged in, click the Identity System Console tab.
- Click the Group Manager Configuration sub-tab, then click the Tabs link in the left navigation pane.
- Click View Object Profile, Configure Panels, then click the link for the panel that you want to modify.
The View Panel page appears.
- Click Modify.
The Modify Panel page appears.
- In the Attributes section of the page, click the Add button, select one or more attributes from the list, then click Save and verify that the attributes you added appear in the View Panel page.

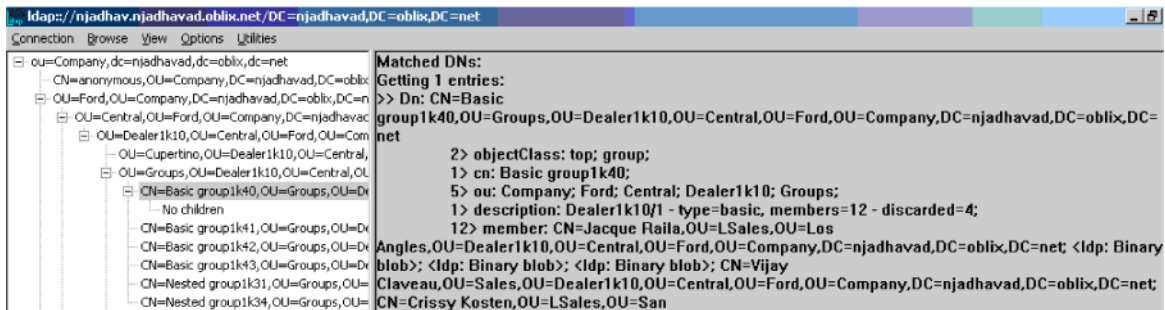
6. Select Group Manager from the Select Application list in the upper right corner.
7. Enter your search criteria in the Selector and click Go.

The results are returned. When you select a Group to review you will notice that the attributes you added dynamically to one group are available only for that group.

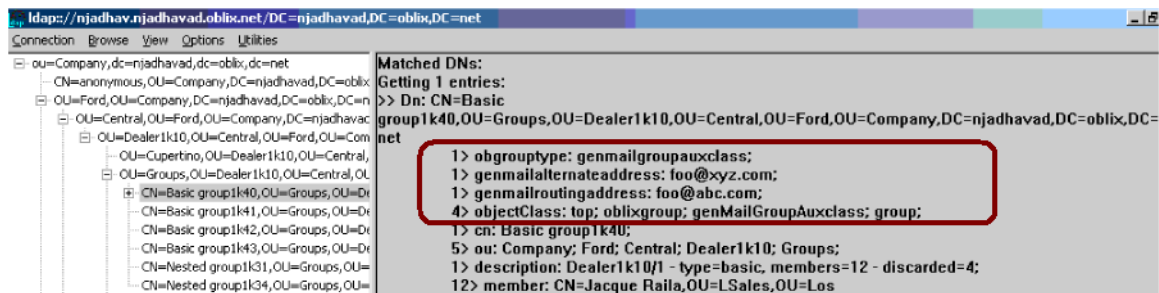
8. Click Modify, click the + button, then add a specific value, and save, as usual.

The entry in the directory has also changed. For example, the following screen shot shows a sample entry before auxiliary classes were added.

Figure D–1 Sample Entry Before Dynamic Auxiliary Classes



The next screen shows the same entry after auxiliary classes were added.



Enabling Fast Bind for Access System Authentication

The Active Directory running on Windows Server 2003 provides a concurrent bind (also known as fast bind) feature that allows multiple authentications over the same LDAP connection.

The Access System supports and uses this feature, which provides the following advantages:

- Fast bind permits two threads to request a bind over one connection at the same time.
- Fast bind provides a faster authentication mechanism because it only validates the password and the account flag and does not build a ticket.

The Fast Bind option must be enabled for each database instance, and is located on individual database profiles in the Access System Console.

To configure the Access System to use a fast bind

1. From the Access System Console, click the System Configuration tab.

- Click the View Server Settings link in the left navigation pane.

The Configure LDAP Directory Server Profile section on this page is where you choose the directory profile to modify.

Configure LDAP Directory Server Profiles

Name	Name Space	Primary Servers	Secondary Servers
☐ default-ID Server 10.1.3 M3 stagh24 6021	o=company,c=us	default	
☐ OracleContext-ID Server 10.1.3 M3 stagh24 6021	cn=Products,cn=OracleContext	default	
☐ AccessManager setup user profile	o=company,c=us	default	
☐ AccessServer default user profile	o=company,c=us	default	

- Click the name of the directory server instance on which you want to enable the Fast Bind feature.

The Modify Directory Server Profile page appears, and you can locate the instance of the directory server profile (also called the database instance) to modify near the bottom of the page.

- Locate and click the name of the directory server profile instance (database instance) that you want. For example:

	Name	Machine	Port number	Server Type
Database Instances	☐ default	stagh24	389	Primary

- Click the link for this instance and check the box beside the Fast Bind option. For example:

The screenshot shows the 'Create Database Instance' form. The 'Fast Bind' checkbox is checked and highlighted with a red box. The form includes the following fields and values:

Field	Value
Name*	FastBind
Machine*	chromium
Port number*	389
Root DN*	o=oracle,c=com
Root password*	
Time Limit	0
Size Limit	0
Flags	☐ SSL ☐ Referral ☒ Fast Bind (only for AD on Windows Server 2003)
Secure Port number	636
Initial Connections	1

- Click Save.
- Confirm that the profile is enabled on the Modify Directory Server profile page.

☒ Enable Profile

- Repeat as needed to enable the Fast Bind option for other database instances.

Enabling Impersonation

In a Windows environment, all processes and threads execute in a security context. Impersonation is the ability of a thread to execute in a security context that is different

from that of the process that owns the thread. The primary purpose of impersonation is to trigger access checks against a client's identity.

For details about enabling impersonation in Oracle Access Manager, which overrides impersonation enabled with IIS, see *Oracle Access Manager Access System Administration Guide*.

Setting Up Integrated Windows Authentication

Oracle Access Manager provides support for integrated Windows authentication (IWA). Your environment may include:

- Windows 2000 Server or Windows Server 2003 or Solaris
- Internet Information Services (IIS) 5.5 or 6.x
- Active Directory or iPlanet directory server

If the user's directory server has, for example, an NT Logon ID, or if the user name is the same everywhere, then a user is able to authenticate into any directory server.

The most common authentication mechanism on Windows 2000 and Windows Server 2003 is Kerberos.

The use of IWA by Oracle Access Manager is seamless. The user won't notice any difference between a typical authentication and IWA when they log on to their desktop, open an Internet Explorer (IE) browser, request a protected web resource, and complete single sign-on.

To see the supported versions and platforms for this integration, refer to Metalink, as follows.

To view information on Metalink

1. Go to the following URL:
<http://metalink.oracle.com>
2. Click the Certify tab.
3. Click View Certifications by Product.
4. Select the Application Server option and click Submit.
5. Choose Oracle Application Server and click Submit.

Process overview: Using IWA authentication

1. The user logs in to the desktop machine, and local authentication is completed using the Windows Domain Administrator authentication scheme.
2. The user opens an Internet Explorer (IE) browser and requests an Access System-protected Web resource.
3. The browser notes the local authentication and sends a token to the IIS Web server.
4. The IIS Web server uses the token to authenticate the user and set up the REMOTE_USER HTTP header variable that specifies the user name supplied by the client and authenticated by the server.
5. The WebGate installed on the IIS Web server uses the hidden feature of external authentication to get the REMOTE_USER header variable value and map it to a DN for the ObSSOCookie generation and authorization.

6. The WebGate creates an ObSSOCookie and sends it back to the browser.
7. The Access System authorization and other processes proceed as usual.

The maximum session timeout period configured for the WebGate is applicable to the generated ObSSOCookie.

Task overview: Setting Up IWA authentication

1. Install a WebGate on the same IIS Web server or servers on which you will set up IWA, as described in the *Oracle Access Manager Installation Guide*.
 - If you installed the WebGate at the Site level, you should perform the tasks at the Site level.
 - If you have multiple WebGates installed at different virtual sites, you should perform the tasks for each virtual site.
2. Enable IWA on the WebGate, as described in ["Enabling IWA on the WebGate Web Server"](#) on page D-11.
3. Configure the WebGate to use IWA, as described in ["Configuring the WebGate for IWA"](#) on page D-12.
4. Create an authentication scheme for IWA in Oracle Access Manager, as described in ["Creating an IWA Authentication Scheme in Oracle Access Manager"](#) on page D-12.
5. Test the IWA implementation, as described in ["Testing IWA Implementation"](#) on page D-13.

Enabling IWA on the WebGate Web Server

The first procedure is to enable IWA on the machine hosting the WebGate.

- If you have installed the WebGate at the Site level, you should perform the tasks at the Site level.
- If you have multiple WebGates installed at different virtual sites, you should perform the tasks for each virtual site.

To enable IWA on the machine hosting the WebGate

1. Start the Internet Services Manager on the machine hosting the WebGate: Start, Programs, Administrative Tools, Internet Services Manager
2. Right-click the Default Web site (or the name of Web server if you changed the name of the Default Web site), then select Properties.

Note: If you installed WebGate at the Site level, right-click the Site then select Properties.

3. Click the Edit button beside Master Properties.
4. Click the Directory Security tab, then click Edit beside "Anonymous access and authentication control."
5. Disable Anonymous Access on the IIS Web Server.
6. Enable Integrated Windows Authentication.
7. Click OK, then click OK again.

8. Restart the IIS Web server.

Configuring the WebGate for IWA

To configure the WebGate for IWA, you must set the user-defined parameter `UseIISBuiltinAuthentication` to true in the Access System Console. See the chapter on configuring the Access System in *Oracle Access Manager Access System Administration Guide* for details.

To modify an AccessGate through the Access System Console

1. Launch the Access System Console, click the Access System Configuration tab, then click the AccessGate Configuration link in the left navigation pane.

The Search for AccessGates page appears.

2. Select the search attribute and condition from the lists, or select All to find all AccessGates.

The Search list is a selection list of attributes that can be searched. The remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

3. Click Go.

The search results are displayed on the page.

4. Click the name of the AccessGate or WebGate that you want to modify.

The AccessGate Details page appears.

5. Click Modify.

The Modify AccessGate page appears. You can enter new information on this page.

You cannot change an AccessGate or WebGate name. To rename it, you must delete it from the Access System Console and then uninstall it. You then create a new AccessGate or WebGate.

6. Type new values as needed.
7. Click Save to save your changes.

Creating an IWA Authentication Scheme in Oracle Access Manager

You must create an IWA authentication scheme for the Access System to use a specific challenge method, challenge parameter, and plug-in, as described in the following procedure.

To create an IWA authentication scheme in the Access System

1. Navigate to the Access System Console, as usual. For example:

`http://hostname:port/access/oblix`

2. Navigate to the Authentication Management page and click Add: Access System Console, Access System Configuration, Authentication Management, Add.
3. Create an Integrated Windows Authentication scheme.

For example:

Name: *Integrated Windows Authentication*

Description: *This scheme is Integrated Windows Authentication, using the built-in Windows authentication mechanism.*

Level: 1

Challenge Method: Ext

Challenge Parameter: creds: REMOTE_USER

SSL Required: No

Challenge Redirect

4. Click the Plug-Ins tab, then click Modify.
5. Select the plug-in name from the list, enter your plug-in parameters and click Add, then save when you are finished.

For example:

Plugin(s)

Plugin Name	Plugin Parameters
credential_mapping	obMappingBase=<"Domain name">, obMappingFilter=" (&(objectclass=user) (samaccountname=%REMOTE_USER%)) "

6. Save the authentication scheme and protect resources using this scheme, as usual.

Testing IWA Implementation

It is always a good idea to test the implementation before you roll it out.

To test IWA

1. Log in to the machine as someone who is a user of both Oracle Access Manager and the Windows operating system.
2. Enter the URL of the protected resource.

Using Access System Password Management

When using the Access System Password Management feature with an Active Directory forest, note the following:

- The Change on Reset, Password Expiration, and Password ExpirationWarning features will work.
- The Number of Retrieves feature will not work.

This limitation applies only if you are using the LDAP mode for Password Management in the Access System and only if you are using Active Directory in a forest configuration.

Using Managed Code and Helper Classes

The .NET Framework provides an object-oriented programming environment to guarantee the safe execution of code and to eliminate performance problems in scripted environments. In the .NET Framework, code that targets the runtime is called managed code.

In addition, MANAGEDLIB actions offer the benefits of managed code, including:

- **Language Choice**--You can write your plug-ins in VisualBasic, C#, Managed C++ (MC++), Java, or PERL.

- **Language Integration**--You can combine MIL modules compiled from different source languages into one assembly or plug-in.

This provides the plug-in writer with a wider range of language choices for plug-in development.

- **Support for Memory Management**--The common language runtime (CLR) provides garbage collection, freeing the plug-in writer from most memory management.

The garbage collector returns memory to the heap when that memory is no longer referenced. However, the plug-in writer should ensure that there are no dangling references to objects. If there are dangling references, garbage collection will not occur for the unused memory.

- **.NET Framework Support**--The .NET framework SDK contains a wide range of functionality. This may reduce the need for third-party support in plug-in code.

Oracle Access Manager can use and call APIs in many languages, including managed code and languages such as C, Managed C++ (MC++), and Visual Basic.Net.

For more information about managed code and managed helper classes, see the *Oracle Access Manager Developer Guide*.

Integrating with Authorization Manager Services

The Access System provides an authorization plug-in that uses the Microsoft Windows Server 2003 Authorization Manager (AzMan) services to make authorization decisions for Access Server clients, including WebGates and callers of the Access Server API.

See the *Oracle Access Manager Integration Guide* for details about configuring a policy domain for the AzMan plug-in.

Integrating with Smart Card Authentication

Oracle Access Manager supports smart card authentication with Active Directory and IIS Web servers in homogeneous Windows environments. Using a smart card provides a stronger form of authentication than a user name and password alone because it is based on *something the user knows* and *something the user has*.

- *Something the user knows* is the user's secret personal identification number (PIN), similar in concept to a personal bank code PIN.
- *Something the user has* is the cryptographically-based identification and proof-of-possession generated by the smart card device that you insert into the smart card reader attached to a computer.

See the *Oracle Access Manager Integration Guide* for details about configuring integrating with smart card authentication.

Integrating the Security Connector for ASP.NET

Oracle Access Manager supports the ASP.NET component of the Microsoft .NET Framework, which developers can use to build, deploy, and run Web applications and distributed applications. The Oracle Access Manager Security Connector for ASP.NET supports and enhances native .NET role-based security.

See the *Oracle Access Manager Integration Guide* for details about how to use the Oracle Access Manager Security Connector for ASP.NET to instantiate a new `OblixPrincipal`

object and populate it with roles (Access System authorization rules) and the native WindowsPrincipal object.

Troubleshooting

For more information on troubleshooting, see "[Troubleshooting Oracle Access Manager](#)" on page F-1.

Microsoft Resources

Active Directory Home Page

<http://www.microsoft.com/windows2000/technologies/directory/ad/default.asp>

ADSI Overview

http://www.microsoft.com/windows2000/techinfo/howitworks/active_directory/adsilinks.asp

Active Directory Programmers Page

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp?frame=true

ADSI Programmers Page

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp?frame=true

Oracle Access Manager Parameter Files

Oracle Access Manager provides a simple means for users to modify the way it operates, by changing the content of specified parameter files, also called catalog files. This appendix describes the file format, provides a list of the files, and describes values within them that you can change to customize Oracle Access Manager system operation.

File Categories

All of the parameter files are located relative to the Identity System installation directory, which could be, for example:

on Windows:

```
c:\COREid\identity\oblix
```

on Unix:

```
/var/COREid/identity/oblix
```

The parameter files can be viewed as belonging to one of several categories, distinguished by the type of parameters they contain:

- Parameters that affect the administrative applications: *User Manager Admin; Group Manager Admin; Organization Manager Admin.*
- Parameters that affect the user applications: *User Manager; Group Manager; Organization Manager; Asynch Mailer; Password Management; Query Builder; Selector.*
- Parameters whose effect is common across many applications: the user applications, the administrative applications and the *Comm Server* (a binary streaming data module).
- Parameters that affect Oracle Access Manager interaction with the directory database (DB), further subcategorized as follows: *user, group, organization, application, configuration, workflow, and LDAP referential integrity* .
- Parameters that affect Oracle Access Manager multitier architecture (for example, the WebPass Web application, or the Identity Server engine).

For More Information on the Parameter Files

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

Troubleshooting Oracle Access Manager

This appendix explains typical problems that you could encounter while running or installing Oracle Access Manager. It contains these sections:

- [Directory and Database Problems and Solutions](#)
- [Other Problems and Solutions](#)
- [Error Messages and Recommendations for Handling Them](#)
- [Capturing Diagnostic Information](#)
- [Need More Help?](#)

Directory and Database Problems and Solutions

This section describes common problems and solutions for the directory and database. It contains the following topics:

- [Problem Setting up SSL Between the Identity Server and Active Directory](#)
- [Error Message to Check if the Directory Server is Running or Responding](#)
- [Access Control and Searchbase Support for eDirectory 8.7.3](#)
- [Unable to Save a Directory Server Profile](#)
- [Active Directory: Adding Members Causes the Group Size to Shrink](#)
- [ADSI Cannot Be Enabled for a Directory Profile](#)
- [Database Validation Fails](#)

Problem Setting up SSL Between the Identity Server and Active Directory

When configuring SSL between the Identity Server and Active Directory, the Identity Server crashes.

Problem

Active Directory uses the Lightweight Directory Access Protocol (LDAP) for read and write operations. By default, LDAP traffic is sent in the clear over an open channel. You can enable a secure version of LDAP using the Secure Sockets Layer (SSL) Transport Layer Security (TLS). The secure version of LDAP that uses SSL is known as LDAPS.

If you have not configured SSL, or if there is a problem with the configuration, the Identity Server crashes when establishing an SSL connection with an LDAPS-enabled Active Directory Server. The Event Viewer application log produces a report similar to the following:

Faulting application ois_server.exe, version 0.0.0.0, faulting module NSLDAPSSL32V41.dll, version 16512.0.0.10521, fault address 0x0004d3cd.

Solution

Install a valid certificate on a domain controller that enables the LDAP service to listen for, and automatically accept, SSL connections for LDAP and global catalog traffic.

For the component to establish an SSL connection to an LDAPS-enabled Active Directory server, the server certificate must contain the fully qualified domain name for the Active Directory domain controller, for example, server.domain.com. This information must appear in the Common Name (CN) of the Subject field.

After receiving the client certificate the LDAP server determines if the CA who issued the certificate is trusted. If the CA is trusted, the server uses the subject name in the certificate to determine if the client has access rights to perform the requested operation.

Error Message to Check if the Directory Server is Running or Responding

During normal operations, you can receive an error indicating that the directory server may not be operational.

Problem

The Access Server or Policy Manager may issue one of the following error messages:

- "Please verify that the Directory Server is running."
- "Please verify that the Directory Server is responding."

These error messages are generated when the Oracle Access Manager component does not receive a response from the directory server within a user-configurable amount of time.

Solution

The following are possible solutions to the problem:

- Check the value for the LDAPOperationTimeout parameter in globalparams.xml.
This parameter enables the Oracle Access Manager component to fail over to a secondary directory server when the primary one takes too long to respond. See the appendix on parameter files in the *Oracle Access Manager Customization Guide* for details.
- Ensure that failover has been configured for this directory server.
See the information on configuring directory server profiles in the *Oracle Access Manager Identity and Common Administration Guide*. Also see the chapter on failover in the *Oracle Access Manager Deployment Guide*.

Access Control and Searchbase Support for eDirectory 8.7.3

You may need to apply patches for access control and searchbase support on Novell eDirectory.

Problem

When conducting searches using Novell eDirectory 8.7.3, attribute access controls and searchbase filters do not work as expected. For example, using eDirectory 8.7.3, you

can configure filters to return organizational units (ou's) below the top node of the DIT, as follows:

```
(&(objectclass=*) (!(|(objectclass=oblixconfig)(objectclass=oblixlocation)(objectclass=genSiteOrgPerson)(objectclass=genSiteGroup)))(objectclass=*))
```

However, these searches return information that you were trying to exclude. For example, users may be returned.

Solution

To work around this issue, apply the eDirectory patch 8.7.3.7. See the following URLs for details:

<http://www.novell.com>

http://support.novell.com/servlet/downloadfile?file=/uns/ftf/edir8737ftf_1.exe

Unable to Save a Directory Server Profile

When saving a directory server profile for use by Identity System and Access System components, you may receive an error similar to the following:

"Unable to save the Directory Server Profile. The applications require a Directory Server Profile to access Policy base with search, modify, and delete operations to function properly. This Directory Server Profile cannot load balance between its servers as well."

Problem

When you install the Access System (at least the Policy Manager), you are asked to identify a location in the directory for policy information. This branch in the directory may or may not be the same as the branch where the Identity System configuration data is stored. Also during Policy Manager installation, a directory profile is created that provides the Identity Servers rights over the policy branch.

The Identity Servers require the ability to search, modify and delete objects in the Access System's policy branch to ensure referential integrity between the Identity and Access Systems. For example, suppose that you allow a user access to a particular resource in the Allow Access page of a policy in the Access System. If you delete the user from the Identity System, referential integrity ensures that the user is also deleted from the policies in the Access System.

If there is no directory profile that provides referential integrity between the Identity and Access Systems, you receive the "Unable to save. . ." error. If you receive this message, you have probably deleted or edited this profile.

Solution

Create another directory server profile with access to the policy branch of the directory.

Active Directory: Adding Members Causes the Group Size to Shrink

Adding users to static groups works properly only up to a point.

Problem

Continuing to add members to static groups causes the group size to shrink.

Solution

Change the value for the parameter `maxForRangedMemberRetrieval` in `globalparams.xml` to a number higher than the desired group membership size:

- If you are using Active Directory on Windows 2003, set the parameter `maxForRangedMemberRetrieval` in `globalparams.xml` to 1500.
- If you are using Active Directory on Windows 2000, set it to 1000.

ADSI Cannot Be Enabled for a Directory Profile

When using Active Directory, you can use the Identity System Console to change the directory profile for user data from ADSI to LDAP or LDAP to ADSI. However, you cannot do this for configuration or policy data.

Problem

When you attempt to change the directory profile for policy or configuration data from the Identity System Console, you get an error. For example, suppose that you store user data in an Active Directory forest using LDAP, and you store configuration and policy data in a different Active Directory forest using ADSI. If you use the Identity System Console to change the ADSI flag in the configuration data database profile to LDAP, after restarting the Oracle Access Manager servers and services, the ADSI flag remains enabled and the following message appears:

"ADSI can be enabled for either user or configuration DB Profile if they are in a separate forest. ADSI Cannot be Enabled for this DB Profile."

Any attempts to modify the directory profile for configuration or policy data to ADSI produces an error because Oracle Access Manager recognizes the profile as ADSI-enabled.

Solution

To modify the directory profile for configuration and policy data, rerun the setup program. See ["Rerunning Setup Manually"](#) on page 7-29 for details.

Database Validation Fails

In the Identity System Console, when you attempt to save a new database instance for an RDBMS profile you may receive a "Database Validation failed" message.

Problem

This problem occurs when creating an RDBMS profile, as described in ["Managing RDBMS Profiles"](#) on page 7-36. Usually, the problem arises because of an incorrect value for the `SQLDBType` parameter in the following file:

`Component_install_dir/identity/apps/common/bin/globalparams.xml`

Where `component_install_dir` is the location where the Identity Server was installed.

Solution

Set the value for the `SQLDBType` parameter as follows:

- For an ODBC connection type, set the value to `Oracle`.
- For an OCI connection type, set the value to `Oracle_OCI`.
- For SQL Server database, set the value to `SQLServer`.

Other Problems and Solutions

This section describes common problems and solutions for components other than the directory and database. It contains the following topics:

- [Auditing for the Identity System Ceases to Work](#)
- [Identity Server Crashes if It Cannot Find a Stylesheet](#)
- [Identity System Deletes a User Entry When an RDN Is Modified](#)
- [JPEG Photo Images Are Not Updated](#)
- [Memory Usage Rises for an Identity Server After Configuring a Directory Server Profile](#)
- [Pass Phrase for Simple Transport Security Has Been Lost](#)
- [Performance of a Component Performance is Slow](#)
- [Reports With Non-ASCII Characters Are Not Imported Correctly in Excel](#)
- [Simple Transport Security Mode Expires After One Year](#)
- [Stylesheet Validation Fails](#)
- [User Creation Might Fail When You Have Multi-byte Characters in the Password](#)
- [WebPass Is Unable to Connect to Its Associated Identity Server](#)

Auditing for the Identity System Ceases to Work

When you configure auditing for multiple Read Application Cluster (RAC) databases, auditing may not work correctly for a while.

Problem

After shutting down and restarting a RAC instance other than the one that was shut down the last time, auditing stops.

Solution

Restart the Identity Server.

Identity Server Crashes if It Cannot Find a Stylesheet

After you customize a stylesheet, the Identity Server crashes or issues an error about a Win32 exception being caught.

Problem

You may have used backslash characters as path separators in your stylesheets in `xsl:include` constructs.

Solution

If you have used backslash characters as path separators in your stylesheets in `xsl:include` constructs, replace the backslashes with forward slash characters. For example, you would want to change the following:

```
<xsl:include href=". \style.xsl" />
```

To this:

```
<xsl:include href="./style.xsl" />
```

Identity System Deletes a User Entry When an RDN Is Modified

The Identity System deletes user entries when you attempt to modify an RDN attribute value. The RDN is the leftmost attribute in a DN. Typically, the RDN attribute is `cn` or `Full Name`.

Problem

This problem occurs when you use Oracle Internet Directory as the back-end repository. A referential integrity setting has not been configured for the Identity Server.

Solution

To fix this problem:

1. Edit the file `ldapreferentialintegrityparams.xml` in the following directory:
Identity_Server_installation_directory\identity\oblix\data\common
2. Change the value of the parameter `referential_integrity_using` from `oblix` to `ds`, as follows:

```
<NameValPair ParamName="referential_integrity_using" Value="ds"/>
```

3. Save the file.
4. Restart the Identity Server for the changes to take effect.
You should be able to modify the RDN attribute value without any problem.
5. If you have multiple instances of the Identity Server installed, make this change to every instance of the Identity Server.

JPEG Photo Images Are Not Updated

When attempting to modify a photo in an Identity application, JPEG photo images are not being updated.

Problem

This problem occurs when a user who has write permission to the Photo attribute does the following:

1. Open the User Manager.
2. View a user profile that contains a photo.
3. Select Panel View.
4. Try to upload a new photo.

Expected result: The photo is updated.

Actual result: The photo does not change.

Solution

Modify JPEG photo images in the page view.

Memory Usage Rises for an Identity Server After Configuring a Directory Server Profile

After configuring a directory server profile, the memory usage for the Identity Server becomes too high. Note that this problem can also apply to an Access Server or Policy Manager.

Problem

When you configure a directory server profile, you are prompted to provide a maximum session time. The default value for the session time is 0 (unlimited). This may cause a performance issue, because the size of the caches for LDAP connections to the Identity Server increase over time. Oracle Access Manager does not control these caches directly.

Solution

To prevent the cache size from causing a performance problem, set the value of the Maximum Session Time (Minutes) for the directory server profile to a finite value, for example, 10 hours, as follows:

1. From the Identity System Console click System Configuration, then click Directory Profiles.
2. Click the link for the profile that you want to modify.
3. In the Max. Session Time (Min.) field, set the value to 600.

Pass Phrase for Simple Transport Security Has Been Lost

If you need to retrieve a Simple mode passphrase, perform the following steps:

1. Locate the password.xml file in the following directory:

webgate\access\oblix\config

```
<?xml version="1.0"?>
<ParamsCtlg xmlns="http://www.oblix.com" CtlgName="password">
  <ValNameList ListName="">
    <NameValPair ParamName="passwd" Value="0243454458404644514154475C"/>
  </ValNameList>
</ParamsCtlg>
```

2. Use obdecrypt.exe on the hash, for example:

```
C:\Documents and Settings\Administrator>cd C:\1014\aaa\access\oblix\tools\ldap_tools
C:\1014\aaa\access\oblix\tools\ldap_tools>obdecrypt 0243454458404644514154475C
Input parameter = 0243454458404644514154475C Decoded output = passphrase
```

Performance of a Component Performance is Slow

The performance of the system or a particular component can be slower than you think it should be.

Problem

You need to determine where to add servers or what component should be tuned for performance.

Solution

Use the information logs to identify components that are processing a heavier load or are taking a particularly long time to service requests. See ["Logging"](#) on page 10-1 for details. In particular, you may want to focus on call processing times. See ["Logging the Amount of Time to Process Requests"](#) on page 10-25 for details.

Reports With Non-ASCII Characters Are Not Imported Correctly in Excel

After modifying and exporting object class attributes, a report.csv file is created. In some languages, the report may have encoding problems.

Problem

In the Japanese Locale or Simplified Chinese Locale, there are encoding problems due to a Microsoft Excel limitation that cannot process CSV files containing data in UTF-8 encoding.

Solution

To process the exported report, complete the process below.

1. Rename report.csv to report.txt.
2. Open report.txt Excel 2003 (Excel 2000 does not support UTF-8 encoding).
3. In the text import wizard, choose encoding as UTF- 8 and comma as the field separator.
4. Click Finish.

Simple Transport Security Mode Expires After One Year

The default value for validity period for Simple transport security mode certificates is 365 days.

Problem

When you configure transport security among Oracle Access Manager components, you can choose between Open, Simple, and Cert modes. See ["Changing Transport Security Modes"](#) on page 8-1 for details.

By default, Simple mode is only operational for one year.

Solution

You can extend the life of the Simple mode certificate as follows. In the following procedure, update the configuration files get changed for all Identity Servers, Access Servers, WebPasses, and WebGates:

1. Open the following files:

`component_install_dir/identity | access/oblix/tools/openssl/openssl_silent.cnf`

`component_install_dir/identity | access/oblix/tools/openssl/openssl.cnf`

Where `component_install_dir` is the directory where the Access or Identity System component was installed.

2. In these files, look for the parameter named `default_days`.

By default, the value for this parameter is 365 days, as follows:

```
default_days = 730 # Duration to certify for
```

3. You can extend the life of the certificate by increasing the number of default days available until expiration.

Update both files with the same value for default days. For example, you can increase the life of the certificate to two years as follows:

```
default_days = 730 # Duration for the certificate
```

4. To regenerate the simple mode certificates with the duration you set in the `openssl_silent.cnf` and `openssl.cnf` files, reconfigure and restart the component using one of the following tools:
 - Access Server: use `configureAAAServer.exe`.
See the *Oracle Access Manager Access System Administration Guide*. for details.
 - WebGate: Use `configureWebGate`.
See the *Oracle Access Manager Access System Administration Guide*. for details.
 - WebPass: Use `setup_webpass.exe`
See "[Changing Transport Security Modes](#)" on page 8-1 for details.
 - Identity Server: `setup_ois.exe`
See "[Changing Transport Security Modes](#)" on page 8-1 for details.

Stylesheet Validation Fails

When you create or customize a stylesheet using Presentation XML, the stylesheet has compilation errors.

Problem

This problem occurs when you do the following:

1. Open a stylesheet in a text editor or (preferably) an XML editor.
2. Change some parameters in the file and save the changes.
3. Open an Identity application, for example, the User Manager, to see the changes.

Expected result: Changes appear as expected.

Actual result: The Identity System issues a bug report.

Solution

This problem can occur for any variety of reasons, but chances are good that there are errors in the way the stylesheet is coded.

Open the XSL file in an Internet Explorer window. If there is an error in the code, the browser will show the line number that contains the error. For more information on Presentation XML, see *Oracle Access Manager Customization Guide*.

User Creation Might Fail When You Have Multi-byte Characters in the Password

Problem:

When you create a user with multi-byte characters in the password using a non-English keyboard, user creation might fail. You might see the error: Directory Server Password Policy violated.

Cause

This problem will occur when you have the 7-bit check plug-in enabled for the "uid" and "userpassword" attributes. In this case, modifying a password for an existing user forces the "7-bit check" for the newly entered password. If the newly entered password contains multi-byte characters, then it does not qualify as "7-bit clean". The product is designed to function in this way.

For example, when creating a workflow, the values are stored under the "obcontainerId=workflowInstances,o=Oblivion,o=company,c=us" node. The password value is stored as "obattrvals: <value>" and is encoded as "7-bit clean". When the Approver approves the workflow, the password value is decrypted and stored under the "userpassword" attribute.

Solution

If you want "7-bit check" to be enabled for workflow steps you need to write your own plug-ins.

Note: Your directory server might not support the 7-bit check. In any case, you must be able to create a user with multi-byte characters.

If you want a user password (or any other attribute) to contain multi-byte characters, you must disable the "7-bit check" for the specific attribute. The following procedure refers to steps for a Sun (formerly iPlanet) directory server. Your details and steps might be different. See your vendor documentation for more information.

To disable the 7-bit check

1. Log in to your directory server as an administrator.
2. Click your directory server instance under "Server Group".
3. Go to the configuration tab for the directory server instance.
4. Expand the "Plug-ins" node to display the list of plug-ins that are applied to your directory server instances.
5. Click "7-bit check" to display the list of attributes that are acted upon by this plug-in.
6. Remove the required attributes or disable the plug-in entirely, as follows:
 - Remove "obattrvals".
 - Disable the plug-in by clicking the Advanced button and set "nsslapd-pluginenabled" to "off".

WebPass Is Unable to Connect to Its Associated Identity Server

If you have installed a WebPass on IIS 6 and enabled logging, the WebPass may be unable to connect to its associated Identity Server.

Problem

This problem occurs when you send logs to an MPFileLogWriter. It does not occur when you send logs to a FileLogWriter.

The problem occurs with the MPFileLogWriter when there is no anonymous user with access to the directory that contains the log files. MPFileLogWriter uses a file named

logfile name.lck to synchronize multiple processes that write to the corresponding log file. The MPFileLogWriter write-locks the .lck file before writing to the oblog.log file.

Solution

Configure an anonymous user with access to the directory that contains the log files. In some circumstances, the user context used to acquire the write-lock will be the IIS Anonymous web user. By default, this user is named IUSR_<computer name>, but you can configure any anonymous user for this purpose.

Error Messages and Recommendations for Handling Them

The following are error messages and troubleshooting tips for handling them.

"Cannot Find xenroll.cab" Error Is Issued When Using a Workflow

When running a workflow, a user may receive a 404 error that states "Cannot find xenroll.cab."

Problem

This problem occurs when a user runs a workflow in an Identity System application, for example:

1. Open the User Manager.
2. View a user profile.
3. Click a Modify button on the profile that invokes an Enroll Certificate Workflow.

In older versions of Oracle Access Manager, the file xenroll.cab was used for certificate enrollment workflows and certificate revocation workflows. However, Oracle has removed support for these workflows. This file is not used anymore.

Solution

You can safely remove the references to xenroll.cab from the stylesheet. The following is an example of this reference. See the *Oracle Access Manager Customization Guide* for details:

```
<head>
... <object id="cenroll" classid="clsid:43F8F289-7A20-11D0-8F06-00C04FC295E1"
codebase="/identity/oblix/apps/common/bin/xenroll.cab" />
... <script
src="http://km.oraclecorp.com/identity/oblix/apps/common/bin/installCert.vbx"
language="VBScript" />
</head>
```

"Enable Failed" Error Is Issued When Using a Workflow

The workflow fails when a user runs it.

Problem

This problem occurs when a user runs a workflow in an Identity System application, for example:

1. Open the User Manager.
2. View a user profile.

3. Click a Modify button on the profile that invokes a Change Attribute Workflow.

Expected result: The workflow behaves as expected.

Actual result: The user receives an "Enabled failed" error.

Solution

There is no definitive solution to this problem, since workflow configuration can fail for a number of reasons. However, a likely candidate is selecting an invalid searchbase during workflow configuration. Delete the searchbase and re-configure the workflow. See ["About the Searchbase"](#) on page 4-23 for details.

"There is No Profile Configured for This Kind of Object" Error Is Issued

You may receive this error when managing administrators in an environment that uses Oracle Internet Directory.

Problem

In Oracle Internet Directory, the orcladmin user (dn: cn=orcladmin) can be thought of as a pseudo user with administrative privileges. There is no LDAP entry corresponding to this user in Oracle Internet Directory. This user is part of special groups that are created in Oracle Internet Directory. The Identity Server requires that every user exist as an independent entry in the directory. When these special groups are viewed or modified using Group Manager, you may see the message, "There is no profile configured for this kind of object."

Solution

If you have this issue, view and update these special Oracle Internet Directory groups using the Oracle Directory Manager application.

Note that there are some special groups in Oracle Internet Directory that exhibit cyclic behavior. Using Oracle Directory Manager to manage these groups is recommended, not the Group Manager or the Identity Server.

"Warning: Page Has Expired" Error Is Issued

You may receive this error when clicking the Back button after conducting a search in an Identity System application.

Problem

By default, Identity System application pages are not cached as a security measure. Caching is disabled to prevent a user from clicking the Back button and seeing what the previously logged-in user was viewing. When caching is turned off, users receive a "page expired" error if they click the Back button from a search results page.

Solution

If you want to cache Identity System application pages, set the value of the parameter browserNoCache to false in the following file:

Identity_Server_install_dir/apps/common/bin/globalparams.xml

```
<SimpleList>
  <NameValPair ParamName="browserNoCache" Value="false"></NameValPair>
</SimpleList>
```

Capturing Diagnostic Information

As of Oracle Access Manager 10.1.4.2, the Access Server and Identity Server provide diagnostic tools to help you work with an Oracle Technical Support representative to troubleshoot problems.

These tools are not for day-to-day administration. Their purpose is to help you investigate problems that require assistance from Oracle Technical Support.

The diagnostic tools enable you to do the following:

- Obtain hard-to-locate information about component configuration and behavior.
For example, if you want to investigate memory consumption or unresponsive servers, you can run diagnostics that provide information on caches, threads and mutexes. This information can help you work with Oracle Technical Support on pinpointing the cause of the problem.
- Automatically capture events that immediately precede a core dump.
Oracle Access Manager automatically writes stack traces of these events to the log files. You can send these log files to Oracle for analysis by the Technical Support team.
- Manually capture a stack trace of any event in the Identity or Access System.
A diagnostic tool enables you to capture ad-hoc stack traces that you can forward to Oracle for analysis by the Technical Support team. For example, you can invoke a stack trace when a server crashes.

The rest of this section discusses the following topics:

- [About Oracle Access Manager Diagnostics](#)
- [Collecting Diagnostic Data](#)
- [Interpreting Diagnostic Output](#)
- [Automatically Writing Stack Traces to a Log File After a Core Dump](#)
- [Manually Requesting a Stack Trace](#)

About Oracle Access Manager Diagnostics

You should only collect diagnostic data to analyze serious problems. In general, you would only run diagnostics when working with a Technical Support representative.

The diagnostic tools are installed with the Access Server and the Identity Server. The diagnostics can focus on three areas: caches, mutexes, and threads. For example, if a server has hung, you can collect diagnostic information on all of the mutexes to see if one has been locked. The diagnostic tool displays the returned data on-screen and saves the data in a file.

As a best practice, you should run the same diagnostics a few times to determine if a problem is temporary or persistent. For example, you can compare diagnostic output to see if memory or cache size is increasing.

Collecting Diagnostic Data

The following tools enable you to collect diagnostic data:

- The tool `aaa_mon.exe` enables you to collect diagnostic data for the Access System.

- The tool `ois_mon.exe` enables you to collect diagnostic data for the Identity System.

The following procedures describe how to use these tools. In the following procedures, a vertical bar ("`|`") represents a choice of command options.

Note: Running the diagnostic tool produces overhead and can affect system performance. The more detail you request when running a diagnostic tool, the more memory is consumed.

For this reason, **do not** request detailed diagnostics for cache queries.

To view a list of supported diagnostic operations

1. Navigate to the following directory:

```
install_dir\identity | access\oblix\tools\ois_mon | aaa_mon
```

Where *install_dir* is the directory where the Identity Server or Access Server is installed.

2. Issue the following command:

```
aaa_mon.exe | ois_mon.exe -s server -p port -i install_dir -m mode  
open | simple | cert -o optype=GetListOfSupportedOperations
```

Where:

- *Server* is the name of the host machine where you want to collect the information.
- *Port* is the listen port for the host.
- *Install_dir* is the installation directory for the component being diagnosed.
- On the `-m` parameter, specify `open`, `simple`, or `cert` for the transport security mode.

See "[Changing Transport Security Modes](#)" on page 8-1 for details. If you specify `cert` you must also specify the following parameter:

```
-c path_for_certs
```

Where *path_for_certs* is the fully qualified path to the certificate files.

To retrieve the name of a diagnostic object

1. Navigate to the following directory:

```
install_dir\identity | access\oblix\tools\ois_mon | aaa_mon
```

Where *install_dir* is the directory where the Identity Server or Access Server is installed.

2. To retrieve the names of all objects of a particular type, issue the following command, using one of the options separated by a vertical bar:

```
aaa_mon.exe | ois_mon.exe -s server -p port -i install_dir -m open | simple | cert  
-o optype=GetDiagnosticInformation,object=cache | mutex | thread,mode=list
```

Where:

- *Server* is the name of the host machine where you want to collect the information.

- *Port* is the listen port for the host.
- *Install_dir* is the installation directory for the component being diagnosed.
- On the *-m* parameter, specify *open*, *simple*, or *cert* for the transport security mode.

See "Changing Transport Security Modes" on page 8-1 for details. If you specify *cert* you must also specify the following parameter:

-c path_for_certs

Where *path_for_certs* is the fully qualified path to the certificate files.

- The *object* parameter limits the results to a particular object type. By default, information is fetched for all diagnosable objects. The following are possible values:
 - *cache*: A cache can hold copies of recently accessed data in memory. A cache can also store data on disk to prevent the program from having to download the same information from the internet. Also see [Table F-1](#) on page F-19 and [Table F-2](#) on page F-20 for a list of cache names and a brief description of each cache.
 - *mutex*: A mutual exclusion object enables multiple program threads to share the same resource, although not at the same time. When a program is started, a mutex is created with a unique name.
 - *thread*: A program can split itself into two or more simultaneously running threads that run concurrently, each performing a different job.

To collect diagnostic information for a particular object

1. Navigate to the following directory:

install_dir\identity\access\oblix\tools\ois_mon\aaa_mon

Where *install_dir* is the directory where the Identity Server or Access Server is installed.

2. Issue the following command:

```
aaa_mon.exe|ois_mon.exe -s server -p port -i install_dir
-m open|simple|cert -o optype=GetDiagnosticInformation[,
object=cache|mutex|thread, mode=brief|detail|list|usage, name=name]
```

Where:

- *Server* is the name of the host machine where you want to collect the information.
- *Port* is the listen port for the host.
- *Install_dir* is the installation directory for the component being diagnosed.
- On the *-m* parameter, specify *open*, *simple*, or *cert* for the transport security mode.

See "Changing Transport Security Modes" on page 8-1 for details. If you specify *cert* you must also specify the following parameter:

-c path_for_certs

Where *path_for_certs* is the fully qualified path to the certificate files.

- The optional `object` parameter for `GetDiagnosticInformation` limits the results to a particular object type.

By default, information is fetched for all diagnosable objects. The following are possible values:

- `cache`: A cache can hold copies of recently accessed data in memory. A cache can also store data on disk to prevent the program from having to download the same information from the internet.
- `mutex`: A mutual exclusion object enables multiple program threads to share the same resource, although not at the same time. When a program is started, a mutex is created with a unique name.
- `thread`: A program can split itself into two or more simultaneously running threads that run concurrently, each performing a different job.

- The optional `mode` parameter for `GetDiagnosticInformation` determines the amount and type of detail returned.

The following are possible values:

- `brief`: Prints a summary of the diagnosable objects on the screen.
- `detail`: Writes the the current values of all diagnosable objects of the specified type to a file in the following directory:

```
install_dir\identity\access\oblix\tools\ois_mon\aaa_mon\OISDiag_  
datetime\AADiag_datetime
```

Where *install_dir* is the directory where the Identity Server or Access Server is installed and *datetime* is the date and time when the file was created.

Do not specify `detail` mode for caches.

- `list`: Shows a list of all diagnosable objects of the specified type.
 - `usage`: Displays help text for the diagnostics tool.
- The optional `name` parameter for `GetDiagnosticInformation` limits the command output to a particular object.

Interpreting Diagnostic Output

You should only attempt to interpret diagnostic output when working with a Technical Support Representative. This section provides simple guidelines for how to read the diagnostic XML output file.

This section discusses the following topics:

- [Interpreting Diagnostic Output When the `list` Parameter Is Used](#)
- [Interpreting Diagnostic Output When the `detail` Parameter Is Used](#)
- [Interpreting the Diagnostic Data for Caches](#)

Interpreting Diagnostic Output When the `list` Parameter Is Used

When you use the `list` parameter on a diagnostic command, the output is written to the screen and to the log file. The most important information is returned in the `name` element of the output file. Also see [Table F-1](#) on page F-19 and [Table F-2](#) on page F-20 for a list of cache names and a brief description of each cache.

The following command excerpt uses `list` mode to retrieve a list of cache names:

```
-o optype=GetDiagnosticInformation,mode=list,object=cache
```

Example F-1 shows output for the following diagnostic command excerpt.

Example F-1 Sample Output for a Diagnostic Command That Used list Mode

```
<?xml version="1.0" encoding="utf-8"?>
<DiagnosticReport>
  <Command>
    <CommandArg
      name="optype">
        <Value>GetDiagnosticInformation</Value>
      </CommandArg>
    <CommandArg
      name="mode">
        <Value>list</Value>
      </CommandArg>
    <CommandArg
      name="object">
        <Value>cache</Value>
      </CommandArg>
    </Command>
  <CommandOutput>
    <Objects>
      <Object
        type="cache">
          <Name>UidInfoCache</Name>
          <Name>Person000IndicatorCache</Name>
          <Name>AuditCache</Name>
          <Name>AuditUserCache</Name>
          <Name>AuditMasterAuditPolicyCache</Name>
          <Name>AuditServerInfoCache</Name>
          <Name>WfDefCache</Name>
          <Name>WfDefSetCache</Name>
          <Name>xslLib_stylesheet</Name>
          <Name>PortalIdCache</Name>
        </Object>
      </Objects>
    </CommandOutput>
  </DiagnosticReport>
```

Interpreting Diagnostic Output When the detail Parameter Is Used

When you use the `detail` parameter on the command to get diagnostic information, the output file contains information on the values set for the object, its current state, and multiple data points about its performance. Items of interest in the output file depend on the type of object. For example, with a cache object, you may be interested in the cache size and if the cache has been flushed.

In general, you should walk through detailed output with a support representative, who can interpret the meaning of the diagnostic.

Example F-2 shows the generic structure of an xml file that is generated using the `list` parameter:

Example F-2 Generic XML Output for a Diagnostic Using mode=detail or mode=brief

```
<?xml version="1.0" encoding="utf-8"?>
<DiagnosticReport>
  <Command type=CommandType1>
    <CommandArg name=Arg1>
```

```

        <Value>ArgValue1</Value>
    </CommandArg>
    <CommandArg name=Arg2>
        <Value>ArgValue2-1</Value>
        <Value>ArgValue2-2</Value>
    </CommandArg>
</Command>
<CommandOutput>
<Objects>
    <Object type=ObjectType1 name=ObjectName1>
        <Attribute name=Attribute1>
            <Value>Value1</Value>
        </Attribute>
        <Attribute name=Attribute2>
            <Value> Value2-1</Value>
            <Value> Value2-2</Value>
            -----
        </Attribute>
    </Object>
    <Object type=ObjectType2 name=ObjectName2>
        <Attribute name=Attribute3>
            <Value>Value3</Value>
        </Attribute>
        -----
        -----
    </Object>
    -----
    -----
</Objects>
<CommandOutput>
</DiagnosticReport>

```

For example, the following is a command excerpt for generating detailed diagnostic data:

```
. . . -o optype=GetDiagnosticInformation,mode=detail,object=cache,name=XSLXDKCache
```

The previous command excerpt produces the output in [Example F-3](#):

Example F-3 Output for a Diagnostic Command Using mode=detail

```

<?xml version="1.0" encoding="utf-8" ?>
<DiagnosticReport>
  <Command>
    <CommandArg name="optype">
      <Value>GetDiagnosticInformation</Value>
    </CommandArg>
    <CommandArg name="mode">
      <Value>detail</Value>
    </CommandArg>
    <CommandArg name="object">
      <Value>cache</Value>
    </CommandArg>
    <CommandArg name="name">
      <Value>XSLXDKCache</Value>
    </CommandArg>
  </Command>
  <CommandOutput>
    <Objects>
      <Object type="cache" name="XSLXDKCache">
        <Attribute name="state">

```

```

        <Value>active</Value>
    </Attribute>
    <Attribute name="Maximum Elements">
        <Value>200</Value>
    </Attribute>
    <Attribute name="Current Elements">
        <Value>11</Value>
    </Attribute>
    <Attribute name="Timeout">
        <Value>0</Value>
    </Attribute>
    <Attribute name="Hit Count">
        <Value>4</Value>
    </Attribute>
    <Attribute name="Miss Count">
        <Value>11</Value>
    </Attribute>
    <Attribute name="Expire Count">
        <Value>0</Value>
    </Attribute>
    <Attribute name="Flush Count">
        <Value>0</Value>
    </Attribute>
    <Attribute name="Memory footprint">
        <Value>945008</Value>
    </Attribute>
    <Attribute name="keys">
        <Value>../../../../lang/en-us/style0/predefinedreports.xml</Value>
        <Value>../../../../lang/en-us/style0/reports.xml</Value>
        <Value>../../../../lang/en-us/style0/usc_admin_main.xml</Value>
        <Value>../../../../lang/en-us/style0/admin_wf_definition.xml</Value>
        <Value>../../../../lang/en-us/style0/wf_quickstart_report.xml</Value>
        <Value>../../../../lang/en-us/style0/reportresults.xml</Value>
        <Value>../../../../lang/en-us/style0/usc_searchresults.xml</Value>
        <Value>../../../../lang/en-us/style0/usc_profile.xml</Value>
        <Value>../../../../lang/en-us/style0/login.xml</Value>
        <Value>../../../../lang/en-us/style0/qbmodify.xml</Value>
        <Value>../../../../lang/en-us/style0/wf_quickstart.xml</Value>
    </Attribute>
</Object>
</Objects>
</CommandOutput>
</DiagnosticReport>

```

Interpreting the Diagnostic Data for Caches

An Oracle Technical Support representative can help you interpret diagnostic data. If you are running diagnostics for a cache, [Table F-1](#) and [Table F-2](#) can help you understand the diagnostic output.

Table F-1 Identity System Cache Names and Descriptions

Cache Name	Description
UidInfoCache	Stores information on the structural object class for each user
PersonOOOIndicatorCache	Caches the out-of-office indicator of each user.
WfDefCache	This is a cache of all workflow definitions.
WfDefSetCache	This is a cache of all workflow definition of a particular type, for example, the create user type of workflow.

Table F–1 (Cont.) Identity System Cache Names and Descriptions

Cache Name	Description
xslLib_stylesheet	This is not used in production environment.
PortalIdCache	This is not used in production environment.
AuditCache	This is a cache of the auditing configuration information for the server.
AuditUserCache	This is a cache of the User Manager audit policy.
AuditMasterAuditPolicy Cache	This is a cache of the master audit policy.
XMLStructureCache	This is a cache of the internal XML data structure that represents each page rendered on the browser.
XSLXDKCache	This cache contains the compiled form of each stylesheet (XSL) used in the Identity system.

Table F–2 Access System Cache Names and Descriptions

Cache Name	Description
UserAccessCache	Used during the authorization phase of evaluation. Holds a hash table of rules and groups that the user satisfies.
AAASyncRequestCache	Used to maintain cache coherency in the Access System and to provide updates to Access clients.
AAAUserCache	Holds list of user profile attributes. Used in the Authenticate, IsAuthorized and Audit Event phases of request evaluation.
AAAUserCredCache	Holds user passwords. Used in the authentication phase of request evaluation to validate passwords.
AuditPolicyCache	Used during processing of an IsResrcProtected event to retrieve the audit mask. Also used during an audit event. Stores audit policies.
AuthentPluginCache	A wrapper for a custom authentication plug-in. Used during authentication.
AuthentSchemeCache	Used during processing of an IsResrcProtected event and the authentication phase of a resource request. Stores authentication scheme details.
AuthzDSOCache	A wrapper for a custom authorization plug-in. Used when processing IsAuthorized events.
AuthzRuleCache	Used when processing an IsAuthorized event to evaluate a user's permissions to access a resource and get action information. Stores policy details.
AuthzSchemeCache	Holds custom authorization scheme information. Used in processing an IsAuthorized event.
ClientConfigCache	Used when processing UpdateConfiguration requests from Access clients.
DomainPasswdPolicyID Mapper	Holds password policy IDs. Used when processing Authenticate events to retrieve password policy information.
GrpQueryCache	Holds a list of groups that the user belongs to. Used during evaluation of ObMyGroups.
HostIdHashString	Used when processing an IsResrcProtected event to find the matching policy. Stores host identifiers.

Table F–2 (Cont.) Access System Cache Names and Descriptions

Cache Name	Description
LPMPolicyCache	Used during the authenticate phase of evaluation to retrieve Lost Password Management policy information.
PSCGrpDefnCache	Holds nested groups and rules for dynamic groups.
PSCUid2OcCache	Used in the authorization phase of evaluation. Contains a hash table of rules and groups that the user satisfies.
PasswdPolicyCache	Used in the authentication phase of evaluation to retrieve password policy information.
PasswdPolicyUserCache	Used in the authentication phase of evaluation to retrieve password policy information.
PolicyCache	Used during the IsAuthorized phase of processing to evaluate access to a resource for user and retrieve action information. Stores policy rules.
SDCache	Used during the IsResrcProtected phase to find the matching policy and authentication scheme. Stores site domain objects.
SessionTokenCache	Stores the decrypted session token. Reduces overhead for decrypting the session token.
URLPrefixCache	Used to find which policy domain the resource maps to. Stores policy domain IDs.
WRORCache	Used when processing an IsResrcProtected event to find the authentication challenge scheme and authentication specifics. Stores authentication scheme details.
jobstatuscache	Holds the status of reporting jobs.

Automatically Writing Stack Traces to a Log File After a Core Dump

If Oracle Access Manager experiences a core dump, you can write a stack trace of the core dump to a log file. The stack trace lists the functions that were called immediately before the dump. The information in the trace can help with troubleshooting. For example, a stack trace can show if a thread has been mired in a process and is not receiving a response from a directory server, or if a message reader thread is alive.

You must enable logging if you want stack traces to be written to a log file. Any logging level is permitted when writing a stack trace. See ["Logging"](#) on page 10-1 for details.

You can send the log file that contains the stack trace to Oracle Technical Support to help with diagnosing the core dump. The first three entries of the stack trace are always the same. These entries belong to the stack trace functionality. The fourth entry is the function that failed in the case of a core dump, or it was the last function that was executed by the thread at the time that you initiated the stack trace.

This information should only be interpreted by an Oracle Technical Support representative.

Manually Requesting a Stack Trace

The following procedure describes how to manually request a stack trace.

To manually request a stack trace

1. Navigate to the following directory:

```
install_dir\identity\access\oblix\tools\ois_mon\aaa_mon
```

Where *install_dir* is the directory where the Identity Server or Access Server is installed.

2. Issue the following command:

```
aaa_mon.exe|ois_mon.exe -s server -p port -i install_dir -m open|simple|cert -o  
optype=GenerateStackTrace
```

Where a vertical bar ("|") represents a choice of commands, *server* is the name of the host machine where you want to collect the information, *port* is the listen port for the host, and *install_dir* is the installation directory for the component being diagnosed.

On the *-m* parameter, specify *open*, *simple*, or *cert* for the transport security mode. See "[Changing Transport Security Modes](#)" on page 8-1 for details. If you specify *cert* you must also specify the following parameter:

```
-c path_for_certs
```

Where *path_for_certs* is the fully qualified path to the certificate files.

3. To view the results, open the log file for the server where the trace was run.
See "[Logging](#)" on page 10-1 for details.

Need More Help?

You can find more solutions on Oracle MetaLink, <http://metalink.oracle.com>. If you do not find a solution for your problem, log a service request.

Index

A

AAA Server

see Access Server

aaa_mon.exe, F-13

About page, 1-11

Access Domain

formerly named NetPoint or COREid Access
Manager Domain, xxii

Access Management API

now named Policy Manager API, xxii

Access Manager

now named Policy Manager, xxii
SDK, configuring, 7-69

Access Manager API

formerly named Access Server API, xxii

Access Manager SDK, 7-69

formerly named Access Server SDK, xxii

Access Server, 1-2

AAA Server configuration option

auditing, 9-2

cache flush, caveat, 8-2

cache updates, 7-68

changing the security password, 8-24

changing to Cert mode, 8-15

changing to Open mode, 8-12

changing to Simple mode, 8-13

changing transport security modes for, 8-10

configureAAAServer tool, 7-31, 8-24

file-based auditing for, 11-18

location for user and configuration data, 7-20

logging, 10-1

reconfiguring after setting up Policy
Manager, 7-31

setting up redirect URLs for account locout, 7-68

SNMP monitoring of, 12-2

transport security for, 8-3

Access Server API

now named Access Manager API, xxii

Access Server SDK

now named Access Manager SDK, xxii

Access System

changing to Open security mode, 8-12

changing to Simple transport security, 8-13

changing transport security modes for, 8-9

log in, 1-7

transport security for, 8-2, 8-3

AccessGate

changing transport security modes for, 8-10

logging, 10-1

transport security for, 8-3

Active Directory, 7-34

Access Server with LDAP, C-3

ADSI configuration with Oracle Access
Manager, B-1

authentication, A-4

authorization, A-4

backward compatibility, A-9

configuring with LDAP, C-1

credential_mapping plug-in, A-6

defining directory server profiles, A-2

deleting a disjoint searchbase, A-3

deploying with, A-1

disjoint searchbases for, A-2

group-search read operations, A-3

increasing the length of the SAM account
name, xxvi

LDAP authentication with ADSI, C-4

Microsoft Global Catalog, C-1

.NET features, A-9

ObMyGroups action attribute, A-5

parent-child authentication, A-4

parent-child authorization, A-5

Policy Manager setup with LDAP, C-2

required schema modifications, 3-17

setting up profiles and searchbases, A-1

timeouts for LDAP, C-3

troubleshooting, A-9

what's new in this release, xxvi

administration

preparing for, 1-1

administrators

about Identity administrators, 2-1

administrators with access to all attributes, 4-32

configuring, 2-1, 2-4

delegated administration

adding delegated administrators, 2-8

configuring, task overview, 2-5

models of, 2-6

delegated administrators, 2-5

Delegated Identity Administrators, 2-1, 2-3

deleting administrators, 2-4

- Identity System, 2-1
- Master Administrators, 2-1, 2-2
- Master Identity Administrators, 2-1, 2-3
- substitute administrators, 2-10
 - assuming another administrator's identity, 2-11
- temporarily granting your rights to another person, 2-10
- ADSI, 7-34
 - bind mechanisms for the Access Server, B-6
 - bind mechanisms for the Identity Server, B-3
 - configuration files, B-4, B-7
 - configuration for Access System, B-6
 - configuring for, B-1
 - configuring for the Access System, B-11
 - configuring for the Identity System, B-9
 - mixed ADSI and LDAP configuration, B-3
 - pageSize parameter, B-12
 - troubleshooting, B-12
- AM Service State
 - now named Policy Manager API Support Mode, xxii
- Anonymous access, 4-27
- Anonymous authentication scheme
 - formerly named NetPoint or COREid None, xxii
- ANR, D-1
- attributes
 - class attribute, 3-6
 - selecting, 3-7
 - class attributes for template object classes, 3-7
 - configuring, 3-10, 3-17
 - configuring lists of radio buttons, check boxes, and so on, 3-19
 - configuring lists of values using directory queries (filters), 3-19
 - data types, 3-10
 - binary, 3-11
 - distinguished name, 3-11
 - integer, 3-11
 - postal address, 3-11
 - string, 3-11
 - telephone, 3-11
 - derived
 - about, 3-27
 - adding to a User Manager tab, 3-29
 - caveats, 3-28
 - example of, 3-28
 - display names, localizing, 3-20
 - display types, 3-14
 - boolean, 3-14
 - check box, 3-15
 - date, 3-15
 - email, 3-15
 - filter builder, 3-15
 - GIF image, 3-15
 - GIF image URL, 3-15
 - location, 3-15
 - media, 3-15
 - multi-line text, 3-15
 - None, 3-14
 - numeric string, 3-15
 - object selector, 3-15
 - postal address, 3-15
 - radio button, 3-16
 - selection menu, 3-16
 - single line text, 3-16
 - S/MIME certificate, 3-16
 - filters to use with searches, 3-22
 - giving users access to the class attribute, 3-7
 - how used in Identity applications, 3-2
 - lists, defining, 3-20
 - localizing, 3-20
 - password, 3-13
 - provisioning (template) attributes, 3-5
 - rules for static lists of attributes, defining, 3-19
 - search
 - attribute used as the key in searches, 3-6
 - based on finding the same value in different attributes, 3-27
 - dynamic, 3-25
 - dynamic with multiple values, 3-26
 - dynamic with wild cards, 3-26
 - returning results that match an attribute on a profile page, 3-25
 - returning targets that match the DN of the logged-in user, 3-25
 - search key (class) attribute, 3-6
 - selecting what can be searched, 4-6
 - selection keys, 4-35
 - semantic types, 3-11
 - challenge, 3-14
 - defined during setup, 3-12
 - DN prefix, 3-12
 - full name, 3-12, 3-13
 - Group Dynamic Member, 3-13
 - Group Owner, 3-13
 - Group Static Member, 3-13
 - location coordinates, 3-13
 - login, 3-12
 - lost passwords, attributes for managing, 3-13
 - map, 3-14
 - none, 3-14
 - password, 3-12
 - photo, 3-12
 - preferred email address, 3-14
 - response, 3-14
 - title, 3-13
 - used in Group Manager, 3-13
 - used in profile pages, 3-12
 - template attributes, 3-5
 - viewing, 3-16
- auditing, 11-1
 - about, 11-1
 - actions Identity applications, 4-45
 - audit database, about, 11-15
 - audit database, creating, 11-21
 - audit database, setting up, 11-19
 - audit reports, setting up, 11-44
 - authentication events, 7-52
 - connecting Access and Identity Servers to the

- database, 11-29
- Crystal Reports, 11-4
- Crystal Reports templates, using, 11-44
- Crystal repository, 11-15
- database auditing architecture, 11-9
- database auditing requirements, 11-8
- database auditing, setting up, 11-19
- diagnostics, 11-7
- diagnostics, on-screen, 11-5
- dynamic, 11-4
- enabling for Access Servers, 11-6
- enabling on Identity Servers, 11-5
- file vs database auditing, 11-2
- file-based, setting up, 11-17
- formatting, 11-5
- Global User Access Privilege Report, 11-7
- GUI location for auditing functions, 11-4
- Identity events to be audited, 11-5
- master audit rule, 11-7
- new features, xxv
- OCI connection type, 11-14
- ODBC data source definitions, 11-12
- ODBC drivers, 11-13
- Oracle Database as the audit repository, xxv
- output type and amount, 11-4
- performance considerations, 11-2
- policy information, 11-3
- profile information, 11-3
- RDBMS profile configuration, 11-14
- RDBMS profiles for, 11-5
- RDMBS profiles for, 11-6
- reports, types of, 11-15
- requirements for, 11-7
- security considerations, 11-2
- SQL Server, installing, 11-20
- static reports, 11-3
- success and failure of Identity System
 - actions, 11-6
- authentication, xviii
 - auditing authentication events, 7-52, 11-15
 - Fast Bind for, 7-34
 - for ADSI, B-11, C-4
 - for transport security, 7-16, 7-42
 - monitoring authentication actions, 12-9
 - monitoring authentication plug-ins, 12-9, 12-10
 - not required for self-registration, 5-63
 - plug-in APIs, xix
 - reports on authentication attempts, 9-3
 - scheme
 - default schemes, xxii
 - scheme for disjoint searchbases, 7-35, 7-36
 - schemes, modifying to include a password policy, 7-64
 - with Active Directory, A-4
- authorization, xviii
 - auditing authorization events, 11-16
 - authorization plug-ins MIB objects, 12-11
 - AzMan plug-in, D-14
 - for ADSI, C-4
 - monitoring authorization events, 12-9

- monitoring authorization plug-ins, 12-9
- plug-in APIs, xix
- with Active Directory, A-4
- auxiliary object classes
 - adding to a tab, 4-8

B

- backURL, 7-66, 7-67

C

- CA certificates
 - importing multiple, 8-23
 - security
 - CA certificates, 8-4
- cache
 - Access cache flush, caveat, 8-2
 - Access Server cache updates, 7-68
 - managing Identity Server caches, 7-13
- Cert mode
 - about, 8-1
 - changing the Access System to, 8-15
 - installing a certificate for, 8-8
- cert7.db, 8-2
- cert8.db, 8-2
- certificates
 - installing, 8-8
- challenge attribute, 7-61
- challenge phrase
 - deleting, 7-60
- change attribute workflow, 5-4
- cloning, 7-70
- components
 - copying, 7-70
- configuration data
 - formerly named Oblix data, xxii
 - pointing to a new directory server, 7-31
 - profile for storing, 7-20
- configuration tree
 - formerly named Oblix tree, xxii
- configureAAAServer command, 8-13, 8-14
- configureAAAServer tool, 8-2
- configureAccessGate, 8-14
- core dump, F-13, F-21
- COREid
 - now named Oracle Access Manager, xxi
- COREid Access Manager Domain
 - now named Access Domain, xxii
- COREid Administrator
 - now named Master Administrator, xxii
- COREid Basic Over LDAP authentication
 - now named Oracle Access and Identity, xxii
- COREid for AD Forest Basic Over LDAP authentication
 - now named Oracle Access and Identity for AD Forest Basic over LDAP, xxii
- COREid Identity Domain
 - now named Identity Domain, xxii
- COREid None authentication

- now named Anonymous authentication, xxii
- COREid System Console
 - now named Identity System Console, xxii
- crash, F-21
- create group workflow, 5-5
- create object workflow, 5-5
- create user workflow, 5-4

D

- data types, 3-10
 - binary, 3-11
 - distinguished name, 3-11
 - integer, 3-11
 - postal address, 3-11
 - string, 3-11
 - telephone, 3-11
- data, exporting
 - see exporting data
- database instance
 - adding, 7-31
 - configuring, 7-32
 - deleting, 7-34
 - for an LDAP profile, 7-31
 - for an RDBMS profile, 7-31
- deactivate user workflow, 5-4
- delegated administration
 - adding delegated administrators, 2-8
 - ASP model, 2-8
 - extranet model, 2-6
 - intranet model, 2-7
 - models of, 2-6
 - what can be delegated, 2-5
- Delegated Identity Administrators
 - definition, 2-1
 - tasks performed by, 2-3
 - see also delegated administration
- delete group workflow, 5-5
- delete object workflow, 5-5
- derived attributes
 - see also attributes, derived
- diagnostics, F-13
 - about capturing, F-13
 - collecting in the log files, 10-20
 - manually recording a stack trace, F-21
 - stack traces, F-21
 - tools for running, F-13
- directory server profile
 - creating, 7-22
 - database instance, configuring, 7-32
 - deleting a database instance from, 7-34
 - modifying, 7-29
 - re-running setup after modifying, 7-29
 - sharing profiles, 8-4
 - viewing, 7-28
- directory servers
 - profiles for, 7-20
 - transport security changes for, 8-19
 - transport security for, 8-3
 - working with multiple searchbases, 7-34

- disjoint searchbases, 7-34

display types

- boolean, 3-14
- check box, 3-15
- date, 3-15
- email, 3-15
- filter builder, 3-15
- GIF image, 3-15
- GIF image URL, 3-15
- location, 3-15
- media, 3-15
- multi-line text, 3-15
- None, 3-14
- numeric string, 3-15
- object selector, 3-15, 3-22
- password, 3-15
- postal address, 3-15
- radio button, 3-16
- selection menu, 3-16
- single line text, 3-16
- S/MIME certificate, 3-16

DIT

- nonoverlapping directory trees, 7-34
- searching multiple branches of, 7-34

E

email

- setting addresses for user feedback, 7-11

exporting data

- about, 3-2, 6-1
- IdentityXML actions, 6-3
- limitations of, 5-20
- object templates, 6-1
- using a workflow, 6-2

F

- failover, 7-37

- Fast Bind, 7-33

features

- new, xxi

filters

- see LDAP filters
- static LDAP, 3-24
- static with wild cards, 3-24
- usage, 3-23

- full name, 3-13

G

- genCert utility, 8-4, 8-14

GIF

- data type, 3-11
- display type, 3-11, 3-15
 - for photos, 4-36
- display type, configuring, 3-27
- files in the Chystal Repository database, 11-13
- image, referencing in a file system, 4-38
- image, used in a location map in a workflow, 5-64

- images used in the Identity System u.i., 7-2
 - location coordinates semantic type, 3-13
 - semantic type, 3-12
 - semantic type for, 3-14
 - tab image, 4-4
 - title image, 4-16
 - globalization
 - support for, xxiii
 - see also localization
 - Group Manager
 - ability to create groups, 4-39
 - about, 4-2
 - adding auxiliary object classes, 4-8
 - adding derived attributes to, 3-29
 - allowing users to view and modify data, 4-32
 - class attribute, 3-7
 - configuring, 4-10
 - configuring multiple languages for, 7-7
 - configuring objects for, 3-1
 - configuring search fields for, 4-6
 - configuring what is returned on a search, 4-7
 - configuring, about, 4-1
 - controlling read access to an object class via a class attribute, 3-7
 - displaying configured object classes, 3-4
 - Dynamic Members Only, 4-44
 - expanding a dynamic group, 4-50
 - Group Manager Configuration tab, 4-2
 - group type panels, 4-18
 - localizing, 4-5
 - My Groups, 4-10
 - My Groups tab, 4-2, 4-18
 - objects configured during installation, 3-3
 - only one tab for, 4-3
 - panels, 4-12
 - configuring, 4-14
 - deleting, 4-14
 - localizing, 4-21
 - ordering, 4-18
 - viewing, 4-13
 - performance consideration, 4-1, 4-23, 4-29, 4-40, 4-41
 - profile pages, 4-12
 - reports, 4-46
 - search
 - starting point for searches in the DIT, 4-22
 - sending data to back-end systems, 6-1
 - subscribing to groups, 4-44
 - supported workflow types, 5-8
 - tabs
 - configuring, 4-2
 - modifying, 4-3
 - viewing, 4-3
 - View Member Profiles, 4-10
 - workflow types for, 5-8
 - groups, 4-43
 - ability to create, 4-39
 - adding members, 4-42
 - change attribute, 5-12
 - configuring legacy groups, xxiv, 4-39
 - create group, 5-12
 - customize using URL parameters, 4-11
 - delete group, 5-12
 - deleting members, 4-42
 - dynamic, 4-10, 4-49
 - dynamic members, showing, 4-11
 - expanding a dynamic group, 4-50
 - finding, 4-41
 - Group Manager application, 1-2
 - group type panels, 4-18
 - mail server for notifications, 7-12
 - managed in the Group Manager, 1-2
 - managed in the Identity System, 1-1
 - managing, 4-40
 - managing large groups, 4-1, 4-23, 4-29, 4-40, 4-41
 - membership determined by an LDAP filter, 4-49
 - nested, 4-10
 - nested members, showing, 4-11
 - static, 4-10
 - static members, showing, 4-11
 - subscribing to, 4-43, 4-44
 - you are a member of, 4-10
 - you are an administrator of, 4-11
- ## H
-
- header panels, 4-12
 - help, 1-11
- ## I
-
- Identity applications
 - see Group Manager
 - see Org. Manager
 - see User Manager
 - about, 1-3
 - configuration, examples of, 4-36
 - configuring
 - see objects and attributes
 - example of configuring, 4-36
 - purpose of, 1-3
 - tabs, 4-2, 4-3
 - modifying, 4-3
 - Identity Domain
 - formerly named COREid Identity Domain, xxii
 - formerly named NetPoint Identity Domain, xxii
 - Identity Server
 - adding, 7-15
 - auditing, 9-2
 - auditing, configuration, 11-36
 - cache flush caveat, 8-2
 - caches, 7-13
 - configuration, 7-9
 - definition, 1-2
 - deleting parameters, 7-18
 - email address for feedback, setting, 7-11
 - Group Manager application, 1-2
 - installation, 1-3
 - logging, 10-1
 - mail server alerts, configuring, 7-12

- managing, 7-14
- managing from the command line, 7-19
- modifying, 7-18
- modifying settings, 7-10
- multiple, setting up, 7-14
- Organization Manager application, 1-2
- session timeout setting, 7-10
- settings, configuring, 7-9
- SNMP monitoring of, 12-2
- timeout threshold, 7-46
- transport security
 - changing, 8-5
- User Manager application, 1-2
- viewing, 7-18
- viewing settings, 7-10
- WebPass connections
 - limiting connection attempts, 7-46
- WebPass plug-in, 1-2
- who configures, 2-2

Identity System

- administration, about, 1-1
- administrators, 2-1
- ADSI configuration, B-2
- components, 1-2
- configuration, about, 1-1
- configuration, overview, 1-4
- configuring, 0-xix
- configuring the Access Manager SDK, 7-69
- configuring, about, 1-3
- Identity Server, 1-2
- installation summary, 1-3
- login, 1-6
- managing, about, 1-5
- transport security
 - changing, 8-5
- transport security for, 8-2, 8-3
- WebPass, 1-2

Identity System Console

- formerly named COREid System Console, xxii

impersonation

- enabling, D-9

installation, 7-70

L

languages
see localization

LDAP

- data
 - configuring for Oracle Access Manager, 3-1
 - process overview, 3-2
 - read and write access to, 4-32
 - viewing on a profile page, 3-2
- filters
 - advanced, 4-31
 - for searches, 4-29
 - query builder, 4-29
- objects in a workflow, 5-7
- objects, on a panel, 4-12
- profiles, 7-20

- redirecting client requests, 7-32
- referrals, 7-32, 7-33

lists

- about, 3-19
- defining, 3-20

localization, 3-20

- about, 7-7
- attribute display names, 4-22
- enabling languages, 7-14
- language evaluation order, 7-8
- managing multiple languages, 7-14
- of administrative pages, 7-8
- of panels, 4-14
- of search results, 4-7
- overview, 7-8
- panel display names, 4-20
- reports, 4-49
- tabs, 4-5

log out

- from the Identity System, 1-11

Logging

- log level, 10-11

logging

- about, 10-1
- activating a log-handler, 10-12
- automatic updates, xxv, xxvi
- autosync, 10-13
- autosync parameter, 10-11
- Buffer_Size, 10-15
- configuration file, 10-4
 - comments in, 10-6
 - modifying, 10-5
 - names, 10-5
 - order of elements, 10-12
 - order of evaluation of entries, 10-13
 - parameters, 10-14
 - structure, 10-9
- configuration file structure, 10-9
- configuring in the Identity System
 - Console, 10-11, 10-17
- default configuration file, 10-6
- default log configuration file, 10-16
- default settings for logs, 10-15
- directing log output, 10-11
- File_Name, 10-15
- FileLogWriter, 10-9
- global log level threshold, 10-10
- levels, 10-13
- ListName, 10-14
- log configuration file path, 10-5
- log file buffer size, 10-12
- log file name, 10-12
- log file rotation, 10-12
- log levels, about, 10-1
- log levels, setting a level for each module, 10-19
- log levels, table of, 10-2
- log output destinations, 10-8
- log writer, 10-11
- log writers, 10-8
- Log_Level, 10-14

- Log_Status, 10-13, 10-14
- Log_Threshold_Level, 10-13
- Log_Writer, 10-14
- Max_Rotation_Size, 10-15
- Max_Rotation_Time, 10-15
- MODULE_CONFIG section, 10-20
- modules within a log file, 10-21
- MPFileLogWriter, 10-9
- new features in this release, xxv, xxvi
- order of elements in the configuration file, 10-12
- output, where sent, 10-8
- per-module thresholds, 10-19
- per-module thresholds, configuring, 10-24
- request processing time, logs of, 10-25
- sending log data to multiple destinations, 10-8
- SNMP, 12-17
- stack traces in log files, 10-4
- synchronizing the configuration file and the Identity System Console settings, 10-13
- SysLogWriter, 10-9
- system log on Solaris and Linux, 10-15
- system log on Windows, 10-15
- thresholds, 10-20
- timing of requests, 10-28
- what's new in this release, xxv, xxvi
- when a server restart is needed, 10-5
- where log data is sent, 10-8
- xmlns, 10-14
- login, 1-6
 - to the Access System, 1-7
 - to the Identity System, 1-6
- logout, 1-11
- lost password management
 - about, 7-57
 - challenge phrases and responses, 7-49
 - configuring, 7-62
 - deleting challenge phrases, 7-60
 - enabling, 7-62
 - new features in this release, xxiii
 - overview of configuring, 7-58
 - presenting multiple challenges phrases, 7-59
 - redirection to a password reset page, 7-65
 - redirection URL, 7-51
 - semantic types for challenge and response, 3-13
 - stylesheets for lost password management, 7-53
 - stylesheets for password reset pages, 7-50
 - URL, syntax, 7-59
 - viewing policies for, 7-62

M

- managing subscriptions, 4-43
- Master Administrator
 - definition, 2-1
 - formerly named COREid Administrator, xxii
 - formerly named NetPoint Administrator, xxii
 - tasks performed by, 2-2
- Master Identity Administrators
 - definition, 2-1
 - tasks performed by, 2-3

- MODULE_CONFIG section, 10-20
- monitoring
 - see SNMP
- MTHML, 7-12
- My Groups, 4-10

N

- name changes, xxi
- names, new, xxi
- .NET, A-9
 - about, D-1
 - adding attributes dynamically, D-6
 - adding attributes for a group, D-7
 - ambiguous names, resolving, D-1
 - ANR, D-1
 - dynamically linked auxiliary classes, D-5
 - enabling Fast Bind, D-8
 - enabling impersonation, D-9
 - Integrated Windows Authentication, D-10
 - integrating the Security Connector for ASP.NET, D-14
 - integration with AzMan, D-14
 - integration with Smart Card authentication, D-14
 - managed code and helper classes, D-13
 - Microsoft Resources, D-15
 - troubleshooting, D-15
 - with Access System password management, D-13
- NetPoint
 - now named Oracle Access Manager, xxi
- NetPoint Access Manager Domain
 - now named Access Domain, xxii
- NetPoint Access Protocol
 - now named Oracle Access Protocol, xxii
- NetPoint Administrator
 - now named Master Administrator, xxii
- NetPoint Basic Over LDAP authentication
 - now named Oracle Access and Identity, xxii
- NetPoint for AD Forest Basic Over LDAP authentication
 - now named Oracle Access and Identity for AD Forest Basic over LDAP, xxii
- NetPoint Identity Domain
 - now named Identity Domain, xxii
- NetPoint Identity Protocol
 - now named Oracle Identity Protocol, xxii
- NetPoint None authentication
 - now named Anonymous authentication, xxii
- NetPoint SAML Services
 - now named Oracle Identity Federation, xxii
- new features
 - auditing to Oracle Database, xxv
 - logging, xxv, xxvi
- new features in this release, xxi
- Novell Directory Server
 - requirements for configuration, 3-18

O

- object class kind, 3-6

- object class type, 3-6
- Object Class(es) field, 4-4
- object classes
 - about, 3-1
 - auxiliary, 3-3
 - structural, 3-3
 - template object classes, 3-4
- Object Selector display type
 - search filters for, 3-22
- object templates
 - see also provisioning
 - see also template objects
 - configuration, 6-4
 - elements in the file, 6-6
 - file, example of, 6-6
 - file, format of, 6-4
 - object template file, 6-4
- objects
 - see also attributes
 - see also object classes
 - adding object classes, 3-8
 - attribute used as the key in searches, 3-6
 - auxiliary object classes, 3-4, 3-9
 - changing the structural object class, 3-8
 - class attribute, 3-6
 - selecting, 3-7
 - class attributes, about, 3-10
 - class kind, 3-6
 - class type, 3-6
 - class types, 3-5
 - configured at installation, 3-3
 - defaults configured at installation, 3-3
 - deleting object classes, 3-9
 - displayed on profile pages, 3-2
 - enabling users to view and modify, about, 3-1
 - generic, 3-5
 - group, 3-5
 - in a workflow, 5-7
 - inheritance of, 3-4
 - location, 3-5
 - mix-in, 3-4
 - modifying, 3-6
 - object templates, 6-1
 - person, 3-5
 - process for configuring, 3-2
 - template object classes, 3-4
 - template objects, 3-2
 - note about modifying, 6-2
 - template objects, about, 6-1
 - template objects, how used in the Identity System, 3-5
 - used for provisioning, 3-2
 - viewing, 3-5
- Oblix data
 - now named configuration data, xxii
- Oblix tree
 - now named configuration tree, xxii
- oblixAdvancedGroup, 4-43
- oblixppcatalog.lst, 5-39
- obtaining information from the Identity Server, 1-2
- OctetString Virtual Directory Engine (VDE)
 - now named Oracle Virtual Directory, xxii
- ois_cert.pem, 8-4
- ois_chain.pem, 8-4
- ois_key.pem, 8-4
- ois_mon.exe, F-13
- Open mode
 - about, 8-1
- Oracle Access and Identity authentication
 - formerly named NetPoint or COREid Basic Over LDAP, xxii
- Oracle Access and Identity for AD Forest Basic over LDAP
 - formerly named NetPoint or COREid for AD Forest Basic Over LDAP, xxii
- Oracle Access Manager
 - formerly NetPoint or COREid, xxi
 - introduction, xviii, 1-5
- Oracle Access Manager List of Bugs Fixed Release 10.1.4 Patchset 1 (10.1.4.2.0), xviii
- Oracle Access Protocol
 - formerly named NetPoint Access Protocol, xxii
- Oracle Application Server 10g Release 2 (10.1.2)
 - also available as Oracle COREid 7.0.4, xxii
- Oracle COREid release 7.0.4
 - also available as part of Oracle Application Server 10g Release 2 (10.1.2), xxii
- Oracle Identity Federation, xxii
 - formerly SHAREid, xxii
- Oracle Identity Protocol
 - formerly named NetPoint Identity Protocol, xxii
- Oracle Virtual Directory Server, 7-26
 - formerly OctetString Virtual Directory Engine (VDE), xxii
- Org. Manager
 - about, 4-2
 - adding auxiliary object classes, 4-8
 - adding derived attributes to, 3-29
 - adding tabs to, 4-6
 - allowing users to view and modify data, 4-32
 - arbitrary tabs in, 4-2
 - change attribute, 5-12
 - class attribute, 3-7
 - configuring at least one panel for, xxiv
 - configuring multiple languages for, 7-7
 - configuring objects for, 3-1
 - configuring search fields for, 4-6
 - configuring what is returned on a search, 4-7
 - configuring, about, 4-1
 - container limits, 4-53
 - container limits, deleting, 4-56
 - controlling read access to an object class via a class attribute, 3-7
 - copying container limits across domains, 4-55
 - create object, 5-13
 - definition, 1-2
 - delete object, 5-13
 - deleting a tab, 4-11
 - displaying configured object classes, 3-4
 - header panels, 4-12

- localizing, 4-5
- location tab, 4-38
- modifying a container limit, 4-55
- multiple tabs for, 4-3
- objects configured during installation, 3-3
- ordering a tab, 4-11
- Org. Manager Configuration tab, 4-2
- panels
 - configuring, 4-14
 - deleting, 4-14
 - localizing, 4-21
 - viewing, 4-13
- reports, 4-46
- search
 - starting point for searches in the DIT, 4-22
- sending data to back-end systems, 6-1
- supported workflow types, 5-8
- tabs, 3-4
 - configuring, 4-2
 - modifying, 4-3
 - viewing, 4-3
- workflow types for, 5-8
- out of office flag, 5-43

P

- panels, 4-14
 - about, 4-12
 - adding, 4-14
 - deleting, 4-14
 - group type panels, 4-18
 - adding, localizing, modifying, deleting, 4-19
 - modifying, 4-14
 - ordering, 4-18
 - using objects on a panel, 4-12
 - viewing, 4-13
- parameter files, E-1
 - about, E-1
- password policies
 - see passwords
- passwords, 3-15
 - see also lost password management
 - challenge phrases, deleting, 7-60
 - changing the Access Server password, 8-24
 - configuring, 7-51
 - configuring policies for, about, 7-49
 - expiration notification, 7-12
 - for Access Server security, 8-24
 - for transport security, changing, 8-21
 - Global Pass Phrase, 8-7
 - lost password management
 - new features, xxiii
 - lost password management attributes, 7-61
 - new features in this release, xxiii
 - notification of expiration, 7-12
 - order of password policy evaluation, 7-50
 - password policies
 - account lockout duration, 7-55
 - account lockout URL, 7-56
 - applying to resources that the Access System

- protects, 7-64
- configuring the default policy, 7-52
- creating, 7-53
- custom account lockout redirect URL, 7-52
- defaults for, 7-51
- deleting, 7-57
- enabling, 7-56
- expiry notification, 7-54
- expiry warning URL, 7-56
- externally provided validation rules, 7-54
- for a specific domain, 7-53
- forcing a change after an administrator
 - reset, 7-54
- implementing in the Access System, 7-64
- including in an authentication scheme, 7-64
- lost password redirect URL, 7-51
- minimum age, 7-54
- minimum length, 7-54
- minimum number of numeric or
 - nonalphanumeric characters, 7-54
- minimum number of upper or lowercase
 - characters, 7-54
- modifying, 7-56
- new in this release, xxiii
- number of allowed login attempts, 7-55
- password change redirect URL, 7-51
- password expiration warning URL, 7-51
- password history, 7-55
- restricting to a domain, 7-53
- stylesheets for the lost password notification
 - page, 7-55
- stylesheets for the password reset page, 7-56
- successful authentication events, 7-52
- unsuccessful authentication events, 7-52
- validity period, 7-54
- viewing, 7-51
- Password semantic type, 3-11, 3-12, 6-6
- password.xml, 8-14
- required semantic type for the person object
 - class, 3-12
- semantic types for lost password
 - management, 3-13
- Sun iPlanet restrictions, 5-64
- transport security passwords, 8-21
- PEM files, 8-4
- performance, 4-1, 4-23, 4-29, 4-40, 4-41
 - processing times for requests, 10-25
- photos
 - default image, 4-38
 - displaying in user profiles, 4-36
 - importing to the directory, 4-37
 - referencing in a file system, 4-38
- plug-ins
 - for Active Directory, A-6
 - logging, 10-1
- policy data
 - profile for storing, 7-20
- policy domain
 - default, xxii
- Policy Manager

- changing to Cert mode, 8-15
- changing to Open mode., 8-12
- changing to Simple mode, 8-13
- formerly named Access Manager, xxii
- location for configuration and user data, 7-20
- logging, 10-1
- transport security for, 8-3
- Policy Manager API, xxii
 - formerly named Access Management API, xxii
- Policy Manager API Support Mode
 - formerly named AM Service State, xxii
- preparing for administration, 1-1
- Procedure
 - Access Manager SDK
 - To configure the Access Manager SDK, 7-69
 - Access System
 - To configure the Access System to work with a disjoint searchbase, 7-35
 - Active Directory
 - To add a disjoint searchbase for the Disjoint_ domain (AD), A-3
 - To configure group-search read operations on Windows 2003, A-3
 - To configure SSO with the Identity or Access System (AD), A-7
 - To configure the credential_mapping plug-in (AD), A-6
 - To enable LDAP authentication for the Access Server, C-4
 - To set up additional directory server profiles, A-2
 - To set up additional directory server profiles (AD), A-2
 - To set up the Access Server for Active Directory, C-3
 - To set up the Policy Manager for Active Directory, C-2
 - To specify Access Server failover after installation, C-4
 - administrators
 - To assign or remove a substitute, 2-10
 - To assume an identity, 2-11
 - To delegate administration, 2-8
 - To delete an administrator, 2-4
 - To revert to your own identity, 2-11
 - To specify Master Administrators and Master Identity Administrators, 2-4
 - ADSI
 - To associate an ADSI agent with every domain, B-3
 - To enable ADSI for additional directory profiles, B-9
 - To enable LDAP authentication for the Access Server, B-12
 - auditing
 - To modify an existing audit schema, 11-23
 - audits, logs, and reports
 - To add an SNMP Manager directly after general parameters, 12-15
 - To add an SNMP trap destination in silent mode, 12-14
 - To add or delete a log-handler
 - definition, 10-19
 - To add or delete log-handler definitions, 10-19
 - To change the formatting of a report, 4-48
 - To configure a collection of SNMP
 - statistics, 12-14
 - To configure a report, 4-46
 - To configure collection of SNMP
 - statistics, 12-14
 - To configure file-based auditing for an Access Server, 11-18
 - To configure file-based auditing for an Identity Server, 11-17
 - To configure general parameters first, 12-15
 - To configure the SNMP Agent and trap
 - destinations, 12-14
 - To connect Crystal Reports to the audit database, 11-46
 - To copy the audit schema to the audit database host, 11-24
 - To copy the Oracle Access Manager-specific Crystal resources, 11-45
 - To create a primary RDBMS instance, 11-33
 - To create an ODBC data source definition (Windows), 11-30
 - To create an ODBC data source definition to connect Crystal Reports to the Oracle/Crystal Repository, 11-46
 - To create an RDBMS profile, 11-33
 - To create and manage user access privilege reports, 11-43
 - To create the audit database (Oracle Database on Linux), 11-22
 - To create the audit database (Oracle Database on Windows), 11-21
 - To create the audit database (SQL Server or Windows), 11-21
 - To delete an SNMP Manager directly after adding one, 12-15
 - To delete an SNMP trap destination in silent mode, 12-15
 - To delete reports, 4-49
 - To edit orMap.ini, 11-46
 - To enable and configure auditing for each Access Server, 11-41
 - To enable and configure auditing for each Identity Server, 11-36
 - To install Crystal Reports, 11-45
 - To install the patch for Crystal Reports, 11-45
 - To localize reports, 4-49
 - To make the RDBMS profile visible (Linux), 11-35
 - To make the RDBMS profile visible (Windows), 11-35
 - To modify audit output formatting for the Access System, 11-42
 - To modify audit output formatting for the Identity System, 11-37
 - To modify the log threshold from the Identity

- System Console, 10-19
- To set or modify auditing policies, 4-45
- To specify global Identity System events and profile attributes for audit, 11-38
- To specify User, Group, or Org. Manager events for audit, 11-39
- To upload and verify the audit schema (Oracle Database on Windows or Linux, 11-28, 11-29)
- To upload the audit schema (SQL Server on Windows), 11-24, 11-27
- To verify that all Identity Servers can record data to the audit database (supported Unix platforms), 11-41
- To verify that all Identity Servers can record data to the audit database (Windows), 11-40
- To verify the audit schema (SQL Server on Windows), 11-26
- To view auditing policies, 4-45
- To view or modify log-handler definitions, 10-18
- To view or modify reports, 4-48
- basics
 - To log in to the Access System, 1-7
 - To log in to the Identity System, 1-7
 - To use the Query Builder, 4-29
 - To use the search function, 1-9
- diagnostics
 - To collect diagnostic information, F-15
 - To retrieve the name of a diagnostic object, F-14
 - To view a list of supported diagnostic operations, F-14
- directory
 - To configure the Access System to work with a disjoint searchbase, 7-35
 - To configure the Identity System to work with a disjoint searchbase, 7-35
- disagnostics
 - To manually request a stack trace, F-21
- Identity applications
 - To add a derived attribute to an application tab, 3-29
 - To add a disjoint searchbase for a disjoint domain, 4-28
 - To add a tab, 4-6
 - To add an auxiliary or template object class to a tab, 4-8
 - To add group members, 4-42
 - To add, modify, or delete a Group Type panel, 4-19
 - To build a complex filter, 4-31
 - To change the formatting of a report, 4-48
 - To change the order in which panels are displayed, 4-18
 - To configure a report, 4-46
 - To configure photos for importing to a directory, 4-37
 - To configure the header panel, 4-13
 - To copy container limits from one domain to another, 4-55
 - To create or add a panel, 4-15
 - To create, view, and modify localized tab configuration, 4-5
 - To delete a container limit, 4-56
 - To delete a disjoint searchbase, 4-28
 - To delete a tab, 4-11
 - To delete group members, 4-42
 - To delete reports, 4-49
 - To expand a dynamic group, 4-50
 - To import photos to the directory, 4-37
 - To localize a panel, 4-17
 - To localize attribute display names, 4-22
 - To localize panel display names, 4-20
 - To localize reports, 4-49
 - To localize search results, 4-7
 - To modify a container limit, 4-55
 - To modify attributes specific to the User, Group, or Org. Manager, 4-21
 - To order the tabs in the Organization Manager, 4-11
 - To reference photos that reside in a file system, 4-38
 - To select what users see in My Groups and View Member Profiles, 4-10
 - To set or modify attribute permissions, 4-33
 - To set or modify auditing policies, 4-45
 - To set the globalparams.xml file, 4-51
 - To set the searchbase, 4-25
 - To specify what attribute can be used in a search, 4-7
 - To subscribe to a group, 4-44
 - To subscribe to multiple groups, 4-44
 - To use the Query Builder, 4-29
 - To view a group, 4-41
 - To view a panel in an end user Identity System application, 4-14
 - To view and add container limits, 4-53
 - To view auditing policies, 4-45
 - To view group members, 4-41
 - To view Group Type panels, 4-18
 - To view or modify a panel's configuration, 4-16
 - To view or modify reports, 4-48
 - To view or modify tab configuration information, 4-3
 - To view the search result attributes, 4-7
- Identity System
 - To change a style, 7-6
 - To change a style name, 7-6
 - To configure the Identity System to work with a disjoint searchbase, 7-35
 - To delete a custom style, 7-6
 - To deploy a style, 7-6
 - To set the default style, 7-7
 - To view currently configured styles, 7-2
- logging
 - To configure a module-specific log threshold, 10-24

- To log the timing of external requests, 10-28
- .NET
 - To add attributes to a Group Profile panel, D-7
 - To configure ANR in Identity System panels, D-4
 - To configure the Access System to use Fast Bind, D-8
 - To create an IWA authentication scheme in the Access System, D-12
 - To enable IWA on the machine hosting the WebGate, D-11
 - To modify an AccessGate through the Access System Console, D-12
 - To specify additional auxiliary object classes in the User Manager, D-6
 - To test IWA, D-13
 - To update configuration data, D-3
 - To use ANR in a search, D-5
- objects
 - To add a derived attribute to an application tab, 3-29
 - To add an object class, 3-8
 - To change user or group structural object classes, 3-8
 - To configure a derived attribute, 3-28
 - To configure a GIF image display type, 3-27
 - To configure an attribute, 3-17
 - To create a filter, 3-22
 - To create a static filter, 3-24
 - To create a static search filter using a wild card, 3-24
 - To create, view, or modify localized attribute display names, 3-21
 - To define a list, 3-20
 - To define a rule, 3-19
 - To delete an auxiliary object class, 3-9
 - To modify an object class type, 3-6
 - To select the class attribute, 3-7
 - To view an application-specific Modify Attribute page, 3-17
 - To view configured object classes, 3-6
 - To view the Modify Attribute page from the System Console, 3-16
- passwords, 7-57
 - To configure lost password management for a password policy domain, 7-62
 - To configure the Lost Password Management attributes, 7-61
 - To create the default password policy, 7-52
 - To enable or disable Lost Password Management, 7-62
 - To enter a password change redirect URL, 7-66
 - To modify a password policy's parameters, 7-56
 - To modify an authentication scheme to include a password policy, 7-64
 - To set up a default password expiry warning redirect URL, 7-67
 - To set up the account lockout URL, 7-68
 - To view a list of password policies, 7-51
 - To view lost password policies, 7-62
- servers
 - To add a style, 7-5
 - To add or modify a database instance for an LDAP directory server profile, 7-32
 - To add or modify a database instance for an RDBMS profile, 7-39
 - To add or modify an RDBMS profile, 7-37
 - To change a style, 7-6
 - To change a style name, 7-6
 - To configure a mail server, 7-12
 - To configure the length of a user's Identity System session, 7-11
 - To create a directory server profile, 7-24
 - To customize email destinations, 7-11
 - To delete a custom style, 7-6
 - To delete a directory server instance for an LDAP directory server profile, 7-34
 - To delete an Identity Server's parameters, 7-19
 - To deploy a style, 7-6
 - To manage a language, 7-14
 - To modify an LDAP Directory Server Profile, 7-29
 - To reconfigure the Access Server, 7-31
 - To rerun Identity System setup, 7-30
 - To rerun Policy Manager setup, 7-30
 - To set the default style, 7-7
 - To view an LDAP directory server profile, 7-28
 - To view currently configured styles, 7-2
 - To view Identity System details, 7-13
 - To view or modify an Identity Server's parameters, 7-18
 - To view or modify server settings, 7-10
- transport security
 - To change the certificate password for the Access System, 8-22
 - To change the certificate password for the Identity System, 8-21
 - To change the Identity Server transport security mode, 8-5
 - To change the transport security mode password, 8-24
 - To change to Cert security mode, 8-15
 - To change to Open security mode, 8-12
 - To change to Simple security mode, 8-13
 - To change transport security between Access Server and the directory server, 8-20
 - To change transport security between the Identity Server and directory server, 8-20
 - To change transport security to SSL between Policy Manager and the directory server, 8-20
 - To install the signed certificate for Cert mode, 8-18
- troubleshooting
 - To collect diagnostic information, F-15
 - To manually request a stack trace, F-21

- To retrieve the name of a diagnostic object, F-14
 - To view a list of supported diagnostic operations, F-14
 - WebPass
 - To add a WebPass, 7-41
 - To change the transport security mode password, 7-45
 - To disassociate an Identity Server from a WebPass, 7-47
 - To modify a WebPass, 7-43
 - To modify a WebPass through the command line, 7-44
 - To reconfigure transport security mode through the command line, 7-44
 - To remove a WebPass, 7-43
 - To update the WebPass poll tracking refresh parameter, 7-48
 - To view a configured WebPass, 7-40
 - workflows
 - To access the Workflow Definition applet, 5-20
 - To add roles to a workflow definition, 5-62
 - To allow a user to perform an asynchronous operation, 5-46
 - To archive a workflow, 5-52
 - To associate a subflow with a workflow, 5-32
 - To associate an Out of Office attribute with a semantic type, 5-41
 - To begin a new workflow definition, 5-21
 - To configure a role, 5-62
 - To configure email notification for workflow steps, 5-35
 - To configure language-specific workflow panel information, 5-59
 - To configure workflow attribute properties, 5-28
 - To copy a workflow as a starting point for a new workflow, 5-54
 - To copy a workflow as an alternative to modifying it, 5-55
 - To create a self-registration workflow, 5-63
 - To create a subflow, 5-32
 - To create this (example) workflow, 5-31
 - To define a self-registration workflow using the QuickStart tool, 5-19
 - To define a workflow target, 5-23
 - To define a workflow using the QuickStart tool, 5-17
 - To define subsequent steps in a workflow, 5-29
 - To define the first step in a workflow, 5-25
 - To delete a workflow, 5-56
 - To delete requests, 5-52
 - To enable a workflow, 5-30
 - To enable time-based escalation, 5-43
 - To export workflows, 5-56
 - To find a workflow ticket, 5-48
 - To invoke a change attribute workflow, 5-48
 - To lock or unlock a ticket, 5-53
 - To make use of the Out of Office flag, 5-43
 - To modify a workflow, 5-55
 - To modify a workflow panel, 5-57
 - To modify oblixpppcatalog.lst, 5-39
 - To modify the workflow parameter files, 5-45
 - To monitor a workflow, 5-51
 - To preload the User, Group, and Organization Managers, 5-47
 - To prepare a workflow step for dynamic participants, 5-38
 - To process a workflow ticket, 5-49
 - To reactivate a deactivated user, 5-50
 - To run a workflow in Group Manager, 5-31
 - To select attributes available for a workflow step, 5-27
 - To specify a surrogate, 5-42
 - To test a workflow, 5-31
 - To view and export a workflow summary, 5-53
 - To view current workflow panel settings, 5-56
 - To view language-specific workflow panel information, 5-58
 - Process overview
 - A Create User workflow example, 5-15
 - Creating and using a Create User workflow, 5-6
 - Using IWA authentication, D-10
 - profile pages, 4-12
 - exporting data
 - see also object templates
- ## Q
-
- Query Builder
 - about, 4-29
 - advanced filters, 4-31
 - QuickStart tool, 5-16
 - example, 5-19
- ## R
-
- RDBMS profile
 - adding, 7-36, 7-37
 - database instance for, 7-39
 - database instance for, adding, 7-39
 - modifying, 7-37
 - reactivate user workflow, 5-5
 - read permission, 4-22
 - realms, 7-34
 - reporting, 9-1
 - see also auditing
 - see also logging
 - see also SNMP
 - reports
 - attributes not viewable via an Identity application, 4-46
 - ResourceFilterSerachScope, 4-51
 - response attribute, 7-61
 - rules
 - about, 3-19
 - defining, 3-19

usage, 3-23

S

SAMAccountName, xxvi, A-9

schema data

 configuring, 3-2

search, 1-9

 see also attributes, search

 aggregating search results, 1-10

 basic, 1-9

 changing the scope of a search, 4-51

 defining search filters, 3-22

 filters, 4-29

 dynamic, 3-25

 dynamic with multiple values, 3-26

 dynamic with wild cards, 3-26

 for Object Selector display type, 3-22

 static, 3-24

 static with multiple targets, 3-24

 static with wild cards, 3-24

 substitution syntax, 3-25

 finding data not viewable in an Identity

 application, 4-46

 finding users at the same level of the DIT as the

 logged in user, 3-25

 for group members, 4-41

 for multiple branches of the DIT, 4-28

 for multiple targets, 3-23, 3-24

 levels of the DIT to search, 4-51

 number of levels of the DIT to include, 4-50

 results, localizing, 4-7

 returning results that match an attribute on a

 profile page, 3-25

 scope, 4-50

 search results attributes, 4-7

 search results, configuring, 4-7

 searchbase

 about, 4-23

 guidelines for setting, 4-23

 setting, 4-25

 selecting items returned on a search, 1-10

 selecting what attributes are returned, 4-7

 setting the searchbase, 4-23

 substitution syntax for, 3-25

 via the selector, 1-10

 working with multiple search bases, 7-34

searchbase

 about, 4-23

 configuring multiple searchbases, xxiv

 deleting, A-3

 disjoint, 4-28

 disjoint searchbases for Active Directory, A-2

 for multiple branches of the DIT, 4-28

 guidelines for setting, 4-23

 multiple, xxiv

 setting, 4-25

 setting for a group, 4-28

security

 see also transport security

Selector, 3-22

self registration, 4-27

self registration workflow, 5-5

semantic type

 challenge, 3-14

 full name, 3-12

 group dynamic member, 3-13

 group owner, 3-13

 group static member, 3-13

 login, 3-12

 map, 3-14

 none, 3-14

 password, 3-12

 photo, 3-12

 preferred email address, 3-14

 response, 3-14

 title, 3-13

session timeout, 7-10

setup

 re-running manually, 7-29

setup_accessmanager, 8-2

setup_ois, 8-2

setup_ois command, 8-7

setup_ois utility, 8-8

SHAREid

 now named Oracle Identity Federation, xxii

Simple mode

 about, 8-1

single sign-on

 configuring for Active Directory, A-7

SMTP server configuration, 7-12

SNMP, 12-1

 about SNMP monitoring, 12-1

 Access Server MIB objects, 12-8

 Access Server traps, 12-11

 Access System Directory Server MIB

 objects, 12-10

 agent, about, 12-2

 agents, destinations for, 12-14

 authentication plug-ins MIB objects, 12-10

 configuration settings, 12-15

 configuring, 12-14

 data, destinations for, 12-14

 destinations for agents and traps, 12-14

 disabling monitoring, 12-13

 enabling monitoring, 12-13

 event traps, 12-2

 Identity Event API MIB objects, 12-6

 Identity Server MIB Objects, 12-4

 Identity Server traps, 12-7

 Identity System directory MIB objects, 12-7

 logging for, 12-17

 Management Information Base, 12-3

 messages, 12-17

 MIB hierarchy, illustration of, 12-3

 MIB index fields, 12-3

 monitoring, disabling, 12-13

 monitoring, enabling, 12-13

 Netstat vs SNMP values, 12-22

 network management station, 12-1

- NMS, use in SNMP monitoring, 12-1
- number of live connections, 12-22
- obscoreboard_params.xml, 12-15
- Oracle Access Manager MIB, 12-3
- polling, 12-2
- polling interval, 12-13
- prerequisites, 12-1
- request queue MIB objects, 12-11
- shutdown interval, 12-22
- statistics, collecting, 12-14
- traps, 12-2
- traps, destinations for, 12-14
- version supported, 12-2
- stack trace, F-13
 - requesting manually, F-21
- stack traces, 10-4, F-21
- styles
 - adding, 7-5
 - adding a custom style directory, 7-3
 - changing, 7-6
 - configuring, 7-2
 - configuring for multiple languages, 7-3
 - deleting, 7-6
 - deploying, 7-5
 - directories for styles, 7-4
 - setting the default, 7-7
 - viewing, 7-2
- subflows
 - about, 5-15
- subscriptions
 - enabling legacy groups to accept, xxiv, 4-39
- substitution syntax, 3-25, 5-23
- synchronization, 7-70

T

- Tab Filter field, 4-5
- tabs, 1-8, 3-4
 - adding auxiliary and template object classes, 4-8
 - adding to Org. Manager, 4-6
 - configuring, 4-2
 - deleting, 4-11
 - localizing, 4-5
 - modifying, 4-3
 - Object Class(es) configuration field, 4-4
 - ordering, 4-11
 - panels, configuring, 4-12
 - profile pages, configuring, 4-12
 - searching, 4-6
 - tab filter field, 4-5
 - viewing, 4-3
- Task overview
 - Assigning dynamic participants to a workflow step, 5-37
 - Configuring a legacy group to support subscriptions, 4-39
 - Configuring ADSI for the Access System, B-11
 - Configuring ADSI for the Identity System, B-9
 - Configuring multi-language functionality, 7-8
 - Creating a plug-in or application to select dynamic

- participants, 5-40
- Creating a workflow definition, 5-4
- Defining a Create Location workflow, 5-65
- Defining a workflow using the workflow applet, 5-19
- Delegating administrators, 2-5
- Displaying information on an application, 4-2
- Enabling database auditing, 11-19
- Enabling Location functionality, 4-39
- Enabling Location functionality and users, 5-65
- Enabling Oracle Access Manager servers to connect to the audit database, 11-30
- Enabling surrogates, 5-41
- Preparing for the audit database, 11-19
- Preparing to use ANR during searches, D-3
- Setting up an RDBMS profile, 11-32
- Setting up for dynamix auxiliary classes, D-6
- Setting up IWA authentication, D-11
- Setting up multiple Identity Servers, 7-15
- To configure auditing, 11-36
- To connect Crystal Reports to the Oracle Repository, 11-46
- To create a secondary RDBMS instance, 11-34
- To set up audit reports, 11-44
- Uploading the audit schema, 11-23
- template attributes
 - in a workflow, 5-7
- template objects
 - about, 3-2
 - classes
 - adding to a tab, 4-8
 - how viewed in the Identity System, 6-5
 - note about modifying, 6-2
 - used in workflows, 6-3
 - used on a panel, 4-12
- timeout
 - enforcing, 7-46
- To, 7-41
- To delete a password policy, 7-57
- To set the globalparams.xml file, 4-51
- transport security
 - about, 8-1
 - changing for the Access Server, 8-10
 - changing for the AccessGate, 8-10
 - changing for the Identity Server, 8-5
 - changing for the WebPass, 8-5
 - changing to Cert for the Access System, 8-15
 - changing to Cert mode, 8-7
 - changing to Open mode for the Access System, 8-12
 - changing to Simple for the Access System, 8-13
 - changing to Simple mode, 8-6
 - passwords, 8-21
 - PEM files, 8-4
 - setting between components, 8-2
 - specifying during installation, 8-2
- troubleshooting, F-1
 - capturing diagnostic information, F-13
 - diagnostic tools, F-13
 - stack traces, 10-4

- typical problems in Oracle Access Manager, F-1
- tuning
 - processing times for requests, 10-25

U

- user data
 - profile for storing, 7-20
- user interface, 1-8
 - styles
 - see also styles
 - customizing, 7-2
 - navigation elements, 1-8
 - styles
 - viewing, 7-2
- User Manager
 - about, 4-2
 - adding auxiliary object classes, 4-8
 - adding derived attributes to, 3-29
 - allowing users to view and modify data, 4-32
 - class attribute, 3-7
 - configuring multiple languages for, 7-7
 - configuring objects for, 3-1
 - configuring search fields for, 4-6
 - configuring what is returned on a search, 4-7
 - configuring, about, 4-1
 - controlling read access to an object class via a class attribute, 3-7
 - definition of, 1-2
 - displaying configured object classes, 3-4
 - header panels, 4-12
 - localizing, 4-5
 - My Identity tab, 4-2
 - objects configured during installation, 3-3
 - only one tab for, 4-3
 - panels
 - configuring, 4-14
 - deleting, 4-14
 - localizing, 4-21
 - viewing, 4-13
 - profile pages, 4-12
 - reports, 4-46
 - sample profile page, 3-2
 - search
 - starting point for searches in the DIT, 4-22
 - sending data to back-end systems, 6-1
 - supported workflow types, 5-8
 - tabs
 - configuring, 4-2
 - modifying, 4-3
 - viewing, 4-3
 - User Manager Configuration tab, 4-2
 - workflow example for, 5-2, 5-6
 - workflow types for, 5-8
- User Manger
 - panels, 4-12
- users
 - adding
 - at the same level of the DIT as the logged in user, xxiv, 3-25, 5-22

- dynamically, xxiv, 3-25, 5-22
 - via substitution syntax, 5-23
- administrative, 2-1
- authentication of, xviii
- authorization of, xviii
- change attribute, 5-10
- configuring data that users see, 3-1
- create, 5-11
- delete, 5-11
- group membership in Group Manager, 1-2
- Identity System sessions, 7-10
- information typically displayed about a user, 3-2
- LDAP attribute permissions, 4-33
- mail server for notifications, 7-12
- managing via User Manager, 1-2
- modify permissions, 4-22, 4-33
- permissions, 4-22, 4-33
- person object class type, 3-5
- reactivate, 5-11
- read and write permissions, 1-1, 4-22, 4-33
- self-registration, 1-1, 5-19
- session timeout, 7-10
- user applications, 4-1
- view permissions, 4-5, 4-22, 4-33

V

- VDS, 7-26
- View Member Profiles, 4-10

W

- WebGate
 - and session timeouts, 7-11
 - certificate request for, 8-5
 - changing to Cert mode, 8-15
 - changing to Open mode, 8-12
 - changing to Simple mode, 8-13
 - logging, 10-1
- WebPass
 - adding, 7-41
 - associating with an Identity Server, 7-45
 - configuring, 7-40
 - definition, 1-2
 - deleting, 7-43
 - disassociating from an Identity Server, 7-47
 - Identity Server
 - limiting attempts to connect to, 7-46
 - install after the Identity Server, 7-40
 - logging, 10-1
 - modifying, 7-41
 - modifying from the command line, 7-43
 - setup_webpass command, 7-44
 - transport security
 - changing, 8-5
 - viewing, 7-40
 - viewing associations with Identity Servers, 7-45
 - who configures, 2-2
- what's new in this release, xxi
- workflows

- about, 5-1
- actions
 - change attribute, 5-10, 5-12
 - create group, 5-12
 - create object, 5-13
 - create user, 5-11
 - delete group, 5-12
 - delete object, 5-13
 - delete user, 5-11
 - reactivate user, 5-11
- actions you can perform in a step, 5-13
- actions, about, 5-7, 5-8
- adding roles to, 5-61
- Anyone role, 5-61
- application for selecting participants, 5-40
- archiving requests, 5-51
- asynchronous operations, 5-46
- committing data via, 5-30
- copying, 5-54
- creating a location workflow, 5-64
- deactivating and reactivating users, 5-49
- defining, 5-20
- defining a target, 5-23
- deleting, 5-55
- dynamic participants, 5-3, 5-35, 5-36
 - overview of assigning, 5-37
- dynamically assigning users to locations in the
 - DIT, xxiv
- enabling, 5-30
- end use of, 5-47
- entry conditions, 5-9
- escalation of, 5-43
- example of, 5-15
- example of creating, 5-31
- example of defining, 5-31
- examples of, 5-2
- exporting, 5-56
- external actions, 5-60
- finding and processing a ticket, 5-48
- how users access workflows, 5-4
- illustration of create user workflow, 5-3
- invoking, 5-48
- localizing, 5-58
- locking a ticket, 5-52
- mail server for ticket processing, 7-12
- modifying the appearance of workflow
 - panels, 5-57
- monitoring, 5-51
- notifications, 5-9
- notifying step participants, 5-34
- Out of Office attribute, 5-41
- out of office flag, 5-43
- overview of creating, 5-4
- participants, 5-9
- performance of, 5-59
- picking a DIT location for the object being
 - created, 5-22
- plug-ins for selecting participants, 5-40
- pre- and post- actions, 5-60
- pre and post processing, 5-9

- QuickStart tool, 5-16
- self-registration, 5-19
 - mail notification, 5-15
- self-registration, creating, 5-62
- sending workflow data to back-end systems, 3-4
- starting a definition, 5-21
- static participants, 5-35, 5-36
- step actions, 5-10
- step actions, about, 5-8
- steps, 5-29
 - attributes, 5-26
 - committing, 5-30
- steps, about, 5-7, 5-8
- subflows, 5-32
- subflows, about, 5-2, 5-15
- subflows, approving, 5-33
- subflows, associating with a workflow step, 5-32
- summary reports, exporting, 5-53
- summary reports, viewing, 5-53
- surrogate participants, 5-3
- surrogate participants, about, 5-41
- surrogates in, 5-42
- target, 5-22
- targets, 5-9
- template objects in, 3-4
- testing, 5-31
- ticket routing, 5-35
- tickets, about, 5-5
- tickets, advanced routing, 5-33
- time-based escalation, 5-43
- type of workflows, 5-8
- types
 - change attribute, 5-4
 - create group, 5-5
 - create object, 5-5
 - create user, 5-4
 - deactivate user, 5-4
 - delete group, 5-5
 - delete object, 5-5
 - reactivate user, 5-5
 - self-registration, 5-5
- types of, 5-3
- use of template objects in, 6-3
- using the workflow applet, 5-19
- viewing workflow panel settings, 5-56
- who initiates, 5-2
- write permission, 4-22

