Oracle® Access Manager

Upgrade Guide

10*g* (10.1.4.0.1)

B25354-01

August 2006

Concepts, methods, strategies, and step-by-step instructions for administrators responsible for upgrading an earlier installation (including the schema and data) to 10*g* (10.1.4.0.1).

ORACLE®

Oracle Access Manager Upgrade Guide 10*g* (10.1.4.0.1)

B25354-01

Primary Author: Gail Tiberi

Contributor: Paresh Borkar, Pradnyesh Rane, Manisha Deshpande, Ramakrishna Narla, Steven Frehe, Ashish Kolli, Gurudatt Shashikumar, Frank Villavicencio, Himadri Pal.

Contents

4 System Behavior and Backward Compatibility

6 Upgrading Identity System Schema and Data

9 Upgrading Remaining Identity System Components

10 Upgrading Access System Components

11 Upgrading Integration Components and an Independently Installed SDK

Part IV Upgrading Your Customizations

12 Upgrading Your Identity System Customizations

13 Upgrading Your Access System Customizations

C Manual Schema and Data Upgrades

D Upgrading Sun Web Server Version 4 to Version 6 on Windows 2000

E Planning Worksheets and Tracking Checklists

F Troubleshooting the Upgrade Process

Index

Preface

This Upgrade Guide provides information about upgrading currently installed Oracle Access Manager components to 10*g* (10.1.4.0.1) on supported platforms. Included are considerations, prerequisites, checklists, and step-by-step instructions to help ensure your success.

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

This Preface covers the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This guide targets the needs of anyone who is responsible to upgrade any Oracle Access Manager component to the latest release. If Oracle Access Manager is not installed, see the *Oracle Access Manager Installation Guide*.

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

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, 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 10*g* (10.1.4.0.1) documentation set:

- *Oracle Access Manager Introduction*—Provides an introduction to Oracle Access Manager, a road map to Oracle Access Manager manuals, and a glossary of terms.
- *Oracle Application Server Release Notes*—Read these for late breaking Oracle Access Manager details. 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 Installation Guide*—Explains how to install and configure the components.
- *Oracle Access Manager Upgrade Guide*—Explains how to upgrade earlier versions to the latest version.
- *Oracle Access Manager Identity and Common Administration Guide*—Explains how to configure Identity System applications to display information about users, groups, and organizations; how to assign permissions to users to view and modify the data that is displayed in the Identity System applications; and how to configure workflows that link together Identity application functions, for example, adding basic information about a user, providing additional information about the user, and approving the new user entry, into a chain of automatically performed steps. This book also describes administration functions that are common to the Identity and Access Systems, for example, directory profile configuration, password policy configuration, logging, and auditing.
- *Oracle Access Manager Access Administration Guide*—Describes how to protect resources by defining policy domains, authentication schemes, and authorization schemes; how to allow users to access multiple resources with a single login by configuring single- and multi-domain single sign-on; and how to design custom login forms. This book also describes how to set up and administer the Access System.
- *Oracle Access Manager Deployment Guide*—Provides information for people who plan and manage the environment in which Oracle Access Manager runs. This

guide covers capacity planning, system tuning, failover, load balancing, caching, and migration planning.

- *Oracle Access Manager Customization Guide*—Explains how to change the appearance of Oracle Access Manager applications and how to control Oracle Access Manager 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, the Plumtree portal, and IBM WebSphere.
- *Oracle Access Manager Schema Description*—Provides details about the Oracle Access Manager schema.

Conventions

The following text conventions are used in this document:

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

What's New in Oracle Access Manager?

The following sections describe the new features in Oracle Access Manager 10*g* (10.1.4.0.1) that are described in this manual:

- Product and Component Name Changes
- Upgrade Planning, Methodology, and Deployment Scenarios
- Planning Worksheets and Tracking Checklists
- Upgrade Concepts and Methods
- Automated Upgrade Processes and Manual Tasks
- Support Changes
- Globalization, System Behaviors, and Backward Compatibility
- Upgrade Prerequisites and Preparation
- Upgrading the Schema and Data
- Component Upgrades
- Customization Upgrades
- Auditing and Reporting Changes
- Combining Challenge and Response Attributes on a Panel
- Validating Your Upgraded Installation
- Troubleshooting

> **Note:** For a comprehensive list of new features and functions in Oracle Access Manager 10*g* (10.1.4.0.1), 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
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
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 Basic Over LDAP
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.

Upgrade Planning, Methodology, and Deployment Scenarios

Planning details for typical deployment scenarios have been added to this book to assist you and your team. This chapter includes a methodology that you can follow based on the two deployment scenarios. Downtime assessment considerations are included to help you establish a time frame for the upgrade in your environment.

See Also: Chapter 1, "Upgrade Overview and Planning"

Planning Worksheets and Tracking Checklists

Planning worksheets provide space where you can enter information about your current environment before upgrading. Checklists are provided to help you and your team track the progress of upgrade tasks in your environment.

See Also: Appendix E, "Planning Worksheets and Tracking Checklists"

Upgrade Concepts and Methods

This book provides upgrade concepts and methods, as well as strategies for back up and recovery and for proceeding with an upgrade when certain support has been deprecated.

See Also: Chapter 2, "Upgrade Concepts and Methods"

Automated Upgrade Processes and Manual Tasks

Get a quick tour of the automated processes and manual tasks involved in the upgrade. Discussions include information about what is preserved during automated processing and what must be handled manually

See Also: Chapter 3, "About Automated Processes and Manual Tasks"

Support Changes

Changes in supported platforms and versions are discussed in this book.

See Also:

"Supported Components and Applications" on page 3-1

"Support Deprecated" on page 2-8

"Platform Support" on page 4-1

Globalization, System Behaviors, and Backward Compatibility

A new chapter has been added that describes system behavior changes between earlier Oracle Access Manager releases and 10*g* (10.1.4.0.1). For example, Oracle Access Manager 10*g* (10.1.4.0.1) has undergone a process to provide globalization support for 29 languages through the use of Unicode. This support is discussed in detail. Some file

formats have changed form the proprietary .lst format to .xml as you can see in all guides. Other system changes have also occurred and are summarized in a centralized overview.

> **See Also:** Chapter 4, "System Behavior and Backward Compatibility"

Upgrade Prerequisites and Preparation

Tasks that you must complete to prepare your earlier installation for the upgrade have been expanded and divided according to upgrade type for your convenience.

> **See Also:** Chapter 5, "Preparing for Schema and Data Upgrades"
>
> Chapter 8, "Preparing Components for the Upgrade"

Upgrading the Schema and Data

A new methodology has been developed to assist administrators who are responsible for the schema and data to perform a schema and data upgrade and ensure that it is successful before the rest of the installation is upgraded.

> **See Also:** Chapter 5, "Preparing for Schema and Data Upgrades"
>
> Chapter 6, "Upgrading Identity System Schema and Data"
>
> Chapter 7, "Upgrading Access System Schema and Data"

Component Upgrades

A new section in this book is devoted to upgrading components following a successful schema and data upgrade.

> **See Also:** Chapter 8, "Preparing Components for the Upgrade"
>
> Chapter 9, "Upgrading Remaining Identity System Components"
>
> Chapter 10, "Upgrading Access System Components"

Customization Upgrades

This information has been expanded and divided according to customization types: Identity System customizations and Access System customizations.

> **See Also:** Chapter 12, "Upgrading Your Identity System Customizations"
>
> Chapter 13, "Upgrading Your Access System Customizations"

Auditing and Reporting Changes

The definitions of oblix_audit_events, oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users tables have changed in Oracle Access Manager 10*g* (10.1.4.0.1) to support internationalized characters. The steps you need to take to process internationalized characters depend on the type of database you are using.

See Also: "Upgrading Auditing and Access Reporting for the Identity System" on page 12-2

"Upgrading Auditing and Reporting for the Access Server" on page 13-2

Combining Challenge and Response Attributes on a Panel

In earlier releases, the challenge phrase and response attributes were allowed on different panels of the Profile page of the User Manager, Group Manager, and Organization Manager. In 10*g* (10.1.4.0.1), however, both the challenge phrase and response attributes must be on the *same* panel.

See Also: "Combining Challenge and Response Attributes on a Panel" on page 12-8

Validating Your Upgraded Installation

A new chapter has been added to help you validate the upgraded environment.

See Also: Chapter 14, "Validating the Entire System Upgrade"

Troubleshooting

A new troubleshooting appendix includes information to help you during all upgrades tasks and processes.

Part I

Introduction

This part of the book introduces upgrading from earlier product releases to 10*g* (10.1.4.0.1).

Part I contains the following chapters:

- Chapter 1, "Upgrade Overview and Planning"
- Chapter 2, "Upgrade Concepts and Methods"
- Chapter 3, "About Automated Processes and Manual Tasks"
- Chapter 4, "System Behavior and Backward Compatibility"

1

Upgrade Overview and Planning

This chapter provides an overview of the upgrade task and planning that you must perform to upgrade you existing Oracle Access Manager (formerly known as Oblix NetPoint or Oracle COREid) deployment to Oracle Access Manager 10*g* (10.1.4.0.1). Other chapters that provide additional information are referenced. The following topics are included:

- Typical Deployment Scenarios
- Upgrade Task Overview
- Upgrade Planning and Deliverables
- Planning Considerations for System Downtime
- Planning Considerations for Extranet and Intranet Deployments
- Upgrade Paths

> **Note:** This book primarily uses new product and component names. For details, see "What's New in Oracle Access Manager?" on page -xxi. This book covers upgrades for Oracle Access Manager components only. For details about upgrading Oracle Application Server components, see *Oracle Application Server Upgrade and Compatibility Guide*.

Typical Deployment Scenarios

Oracle Access Manager deployments fall into two categories: Identity System only or joint deployments of both the Identity and Access Systems. The upgrade tasks that must be performed, and the sequence in which you perform these tasks, depend upon the type of deployment you have. For more information, see:

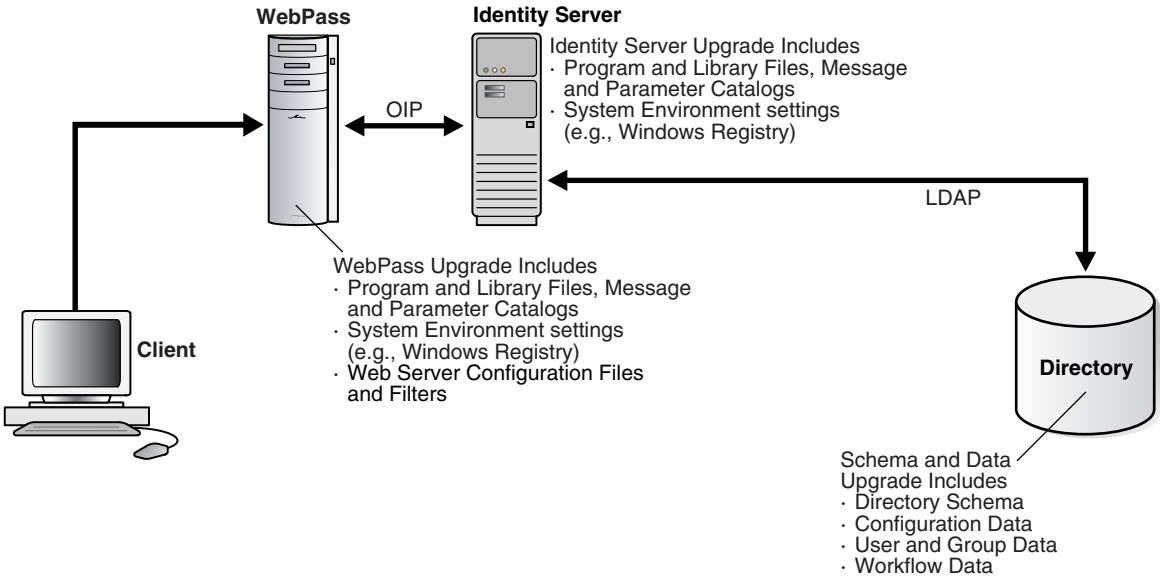
- About Upgrading Identity System Only Deployments
- About Upgrading Joint Identity System and Access System Deployments

> **Note:** During component upgrades, the same kind of Oracle Access Manager information is upgraded for each component.

About Upgrading Identity System Only Deployments

Figure 1–1 illustrates a very simple Identity System-only deployment. Identified in the figure are the types of information that are upgraded for each Identity System component. As you can see, the Identity System schema and data are also upgraded.

Figure 1–1 Identity System Deployment Overview



The Oracle Access Manager schema and Identity System data reside in the directory server. Identity System schema and data upgrades are performed only once and require write access to information in the directory server.

Identity System Schema and Data Upgrades

Identity System schema and data upgrades include updating the following information types to meet requirements of the latest release:

- Oracle Access Manager schema
- Oracle Access Manager configuration data
- Oracle Access Manager user and group data and runtime information
- Oracle Access Manager workflow data

Component information resides in the installation directory of the specific Oracle Access Manager component. The type of information that is upgraded depends on the component type: Oracle Access Manager Server or Oracle Access Manager Web component. For example:

Identity Server Component Upgrades

Each Identity Server component upgrade brings the following information up to release 10*g* (10.1.4.0.1):

- Program and library files, including message and parameter catalogs, are replaced with the latest versions
- Configuration settings, as well as system environment settings (in the Windows registry, for example), are updated to comply with requirements for the latest Identity Server release

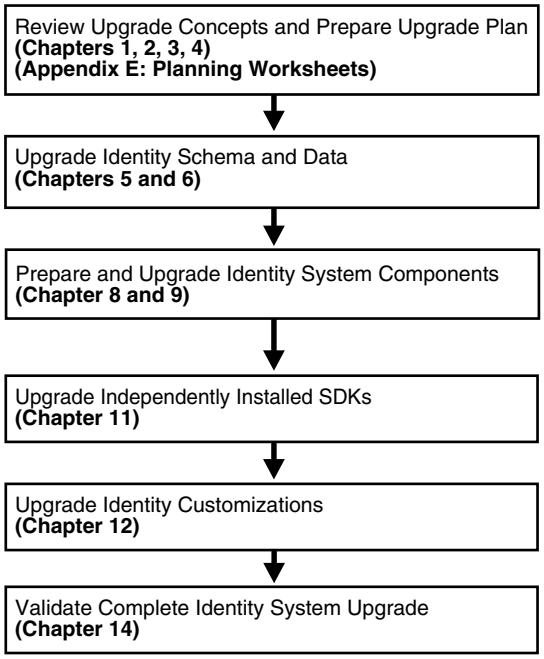
Identity System Web Component Upgrades

WebPass is the Identity System Web component. Each WebPass component upgrade brings the following information up to release 10*g* (10.1.4.0.1):

- Program and library files, including message and parameter catalogs, are replaced with the latest versions
- WebPass configuration settings, as well as system environment settings (in the Windows registry, for example), are updated to comply with requirements for the latest WebPass release
- Configuration files and filters for the Web server hosting the WebPass plug-in are updated to accommodate requirements for the latest WebPass release

Figure 1–2 illustrates the sequence of upgrade tasks that you must perform when you have an Identity System-only deployment.

Figure 1–2 Identity System Only Upgrade Tasks and Sequence

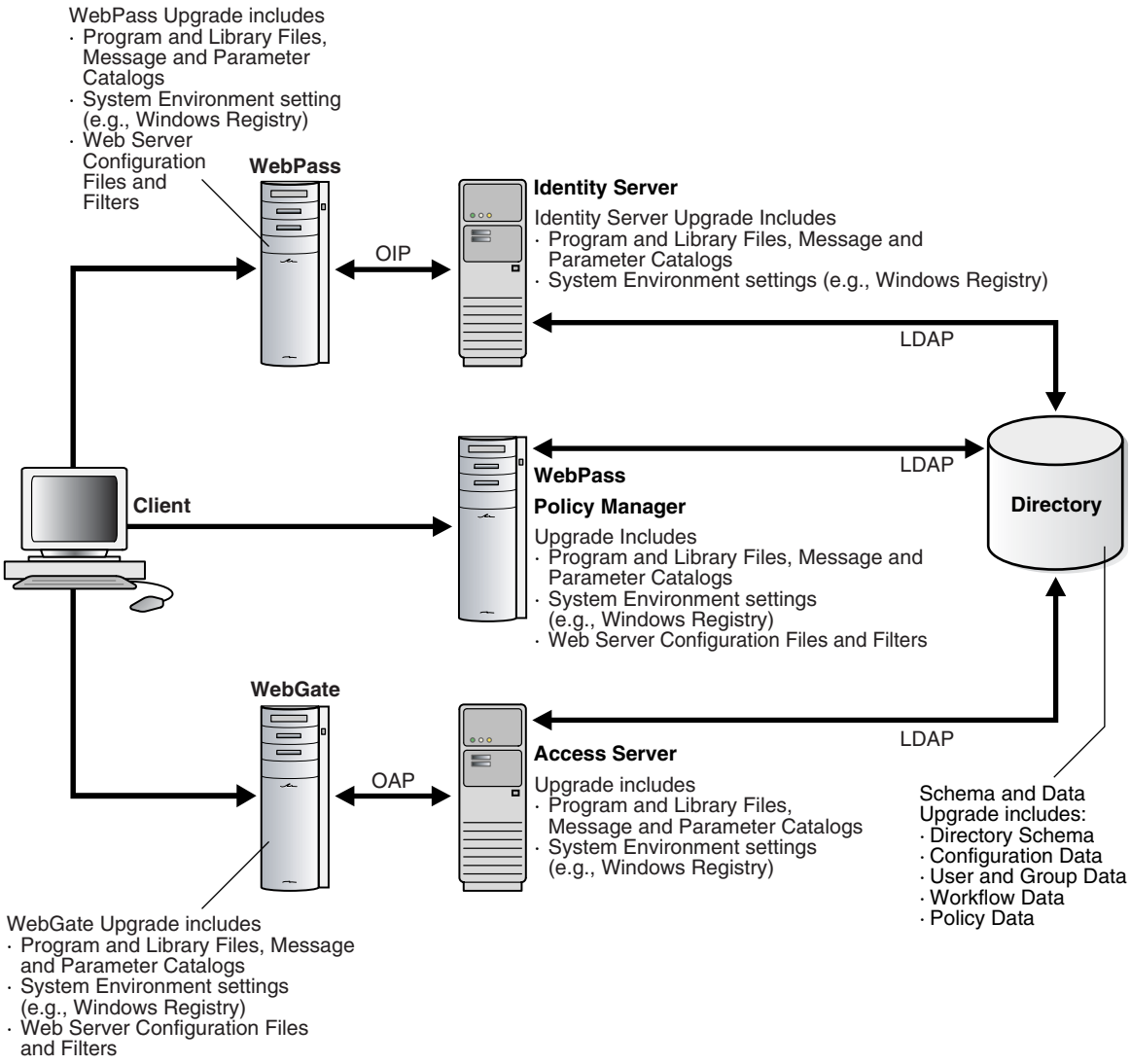


An introduction to each task is described in "About the Execution Stage" on page 1-8

About Upgrading Joint Identity System and Access System Deployments

Figure 1–3 illustrates a very simple joint deployment of both the Identity System and Access System. Identified in the figure are the types of information that are upgraded for each component. As you can see, both the Identity System and Access System schema and data are also upgraded.

Figure 1–3 Joint Identity and Access System Deployment Overview



The Oracle Access Manager schema and Identity and Access System data reside in the directory server. Schema and data upgrades are performed as described next and require write access to information in the directory server.

Identity System Schema and Data Upgrades

Even in a joint Identity and Access System deployment, Identity System schema and data upgrades include updating the following information types to meet requirements of the latest release:

- Oracle Access Manager schema
- Oracle Access Manager configuration data
- Oracle Access Manager user and group data and runtime information
- Oracle Access Manager workflow data

Access System Schema and Data Upgrades

Access System schema and data upgrades include updating the following information types to meet requirements of the latest release:

- Oracle Access Manager policy data
- Additional schema updates are not typically required for the Access System unless you have directory instances configured for use by only the Access System

Component information resides in the installation directory of the specific Oracle Access Manager component. The type of information that is upgraded depends on the component type: Oracle Access Manager Server or Oracle Access Manager Web component. For example:

Identity Server and Access Server Upgrades

Each Identity Server and Access Server component upgrade brings the following information up to release 10*g* (10.1.4.0.1)

- Program and library files, including message and parameter catalogs, are replaced with the latest versions
- Configuration settings, as well as system environment settings (in the Windows registry, for example), are updated to comply with requirements for the latest Identity Server release

Policy Manager, WebPass, and WebGate Upgrades

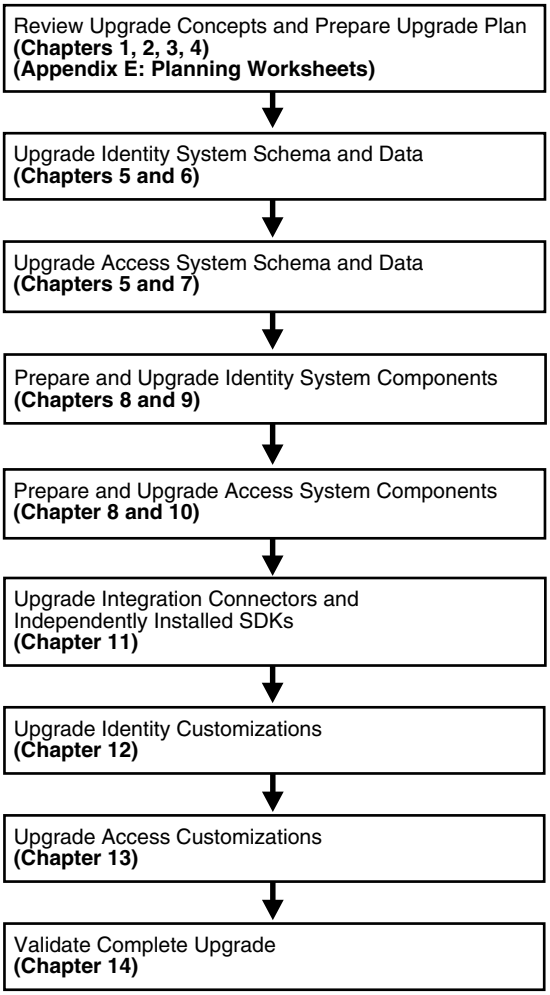
Each Oracle Access Manager Web component upgrade (Policy Manager, WebPass, and WebGate) brings the following information up to release 10*g* (10.1.4.0.1):

- Program and library files, including message and parameter catalogs, are replaced with the latest versions
- Configuration settings, as well as system environment settings (in the Windows registry, for example), are updated to comply with requirements for the latest WebPass release
- Configuration files and filters for the Web server hosting the Web component plug-in are updated to accommodate requirements for the latest release

A WebPass must also be installed with each Policy Manager on the same Web server instance, at the same directory level.

Figure 1–4 illustrates the sequence of upgrade tasks that you must perform when you have joint Identity and Access System deployment.

Figure 1–4 Upgrade Tasks and Sequence in Joint Identity and Access System Deployments



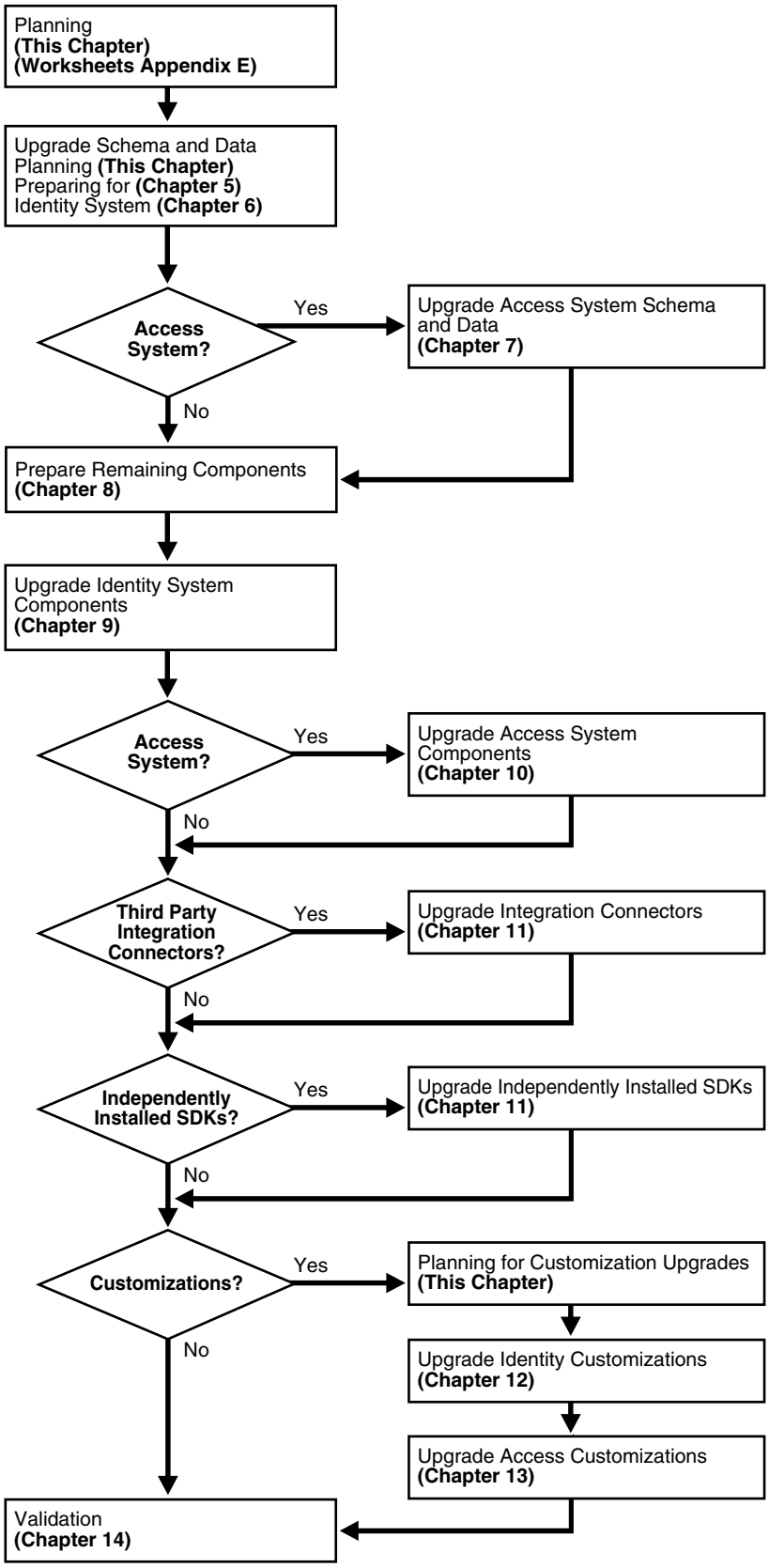
An introduction to each task is described in "About the Execution Stage" on page 1-8.

Upgrade Task Overview

This discussion provides a very high level introduction to the sequence of tasks that you must perform. This is only a starting point in your planning. You perform the entire upgrade task in a sequential order in relation to the deployment approach adopted in your organization: Identity System only or Joint Identity and Access System deployment.

Figure 1–5 provides a high-level view of the upgrade tasks that you need to complete in each of your deployment environments, and the order in which these tasks must be performed. Additional information is provided in "About the Execution Stage" on page 1-8.

Figure 1–5 High-Level Upgrade Task Overview



About the Planning Stage

Before you start any upgrade activities, it is important to read through this entire chapter. In addition you need to collect and record specific details about your existing deployment.

For more information and specific details about planning, see "Upgrade Planning and Deliverables" on page 1-10.

For downtime assessment planning, see "Planning Considerations for System Downtime" on page 1-17.

Worksheets that you can use to record details about your existing deployment are provided in Appendix E, "Planning Worksheets and Tracking Checklists".

About the Execution Stage

This stage is illustrated in Figure 1–5 and outlined next. The sequence of tasks that you must complete is critical to your success. Checklists that can help you track the progress of upgrade tasks in your environment are provided in Appendix E, "Planning Worksheets and Tracking Checklists".

Note: Task overviews like the one here outline the tasks that you must perform and provide a pointer to the discussion that provides the information you need to perform the task.

Task overview: Performing the upgrade includes

1. **Planning**: Develop a planning document that defines a detailed approach for each of your installed environments is described in:.
   - Upgrade Planning and Deliverables on page 1-10 outline details you need to record for all earlier installed Identity and Access System components, directory servers, Web servers, and applications
   - Schema and Data Upgrade Planning on page 1-12 introduces considerations and sequences for schema and data upgrades
   - Customization Upgrade Planning on page 1-13 discusses considerations and sequences to upgrade earlier customizations
   - Planning Considerations for System Downtime on page 1-17 introduces strategies to minimize system downtime during the entire upgrade procedure
   - Planning Considerations for Extranet and Intranet Deployments are described on page 1-22
   - Upgrade Paths on page 1-24 outlines starting releases and strategies for each one
2. **Upgrading the Schema and Data**: Prepare for and upgrade the earlier Oracle Access Manager schema and data as described in:
   - Schema and Data Upgrade Planning on page 1-12
   - Chapter 5, "Preparing for Schema and Data Upgrades"
   - Chapter 6, "Upgrading Identity System Schema and Data"
   - Chapter 7, "Upgrading Access System Schema and Data"

3. **Preparing Remaining Components**: After the schema and data upgrade, you must prepare other components for the upgrade as described in Chapter 8, "Preparing Components for the Upgrade".
4. **Upgrading Identity System Components**: Perform Identity System component upgrades as described in Chapter 9, "Upgrading Remaining Identity System Components" and outlined in the following list:
   - Upgrading Remaining Identity Servers, one by one, as described on page 9-3

     Note: When you have auditing and access reporting configured for a database in the earlier environment, immediately following each Identity Server upgrade you must import earlier Identity Server auditing data to a new database instance, as discussed in "Upgrading Auditing and Access Reporting for the Identity System" on page 12-2.

   - Upgrading Remaining WebPass Instances, one by one, on page 9-8
   - Validating the Identity System Upgrade on page 9-11
   - "Backing Up Upgraded Identity Component Information" on page 9-12
5. **Upgrading Access System Components**: Perform Access System component upgrade as described in Chapter 10, "Upgrading Access System Components" and outlined in the following list:
   - Creating a Temporary Directory Profile For Access System Upgrades must be performed after upgrading the Access System schema and data and before upgrading any other Access System components as described on page 7-10
   - Upgrading Remaining Policy Managers, one by one, as described on page 10-2
   - Upgrading Access Servers on page 10-6

     Note: When you have auditing and access reporting configured for a database in the earlier environment, immediately following each Identity Server upgrade you must import earlier Identity Server auditing data to a new database instance, as discussed in "Upgrading Auditing and Reporting for the Access Server" on page 13-2.

     Upgraded Access Servers are automatically backward compatible with earlier WebGates. For more information, see Chapter 4, "System Behavior and Backward Compatibility".

   - Upgrading WebGates can be staggered and performed gradually over time, as discussed in on page 10-9.
6. **Upgrading Third-Party Integration Connectors**: Upgrade any Oracle Access Manager connectors for third-party integration components and for the J2EE Application Server (if any are being used) as described on page 11-1.
7. **Upgrading Independently Installed Software Developer Kits**: Perform this upgrade to ensure the older APIs are upgraded (and ensure that any plug-ins developed using those APIs are compatible and working properly in the upgraded environment) as described on page 11-4.
8. **Upgrading Customizations**: This task can be started well in advance of other tasks and performed in a separate environment to reduce the amount of system downtime as described in:

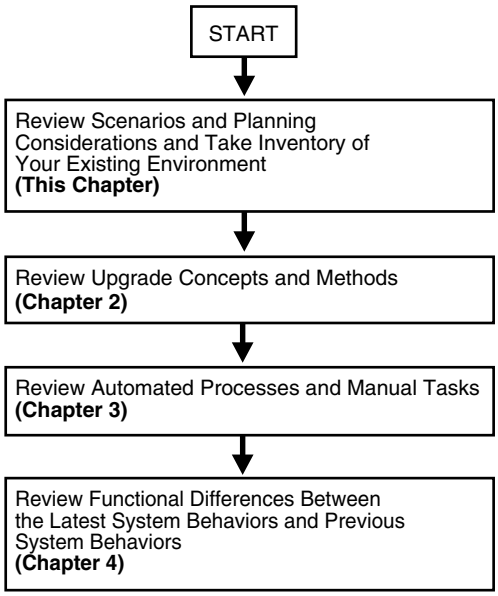
- Customization Upgrade Planning on page 1-13
- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"

9. **Validating the Upgrade**: After all other work is completed, you may verify system operation as described in Chapter 14, "Validating the Entire System Upgrade".

Upgrade Planning and Deliverables

Oracle strongly recommends that before starting any upgrade task, you and your team become familiar with all topics suggested in Figure 1–6, and the overview that follows the figure.

Figure 1–6 Upgrade Planning Overview



Task overview: Planning for your upgrade

1. Review the following information in this chapter to get a high-level overview of the upgrade task, considerations, planning deliverables, deployment scenarios, and starting points. For more information, see:
   - Planning Considerations
   - Planning Deliverables
   - Schema and Data Upgrade Planning
   - Customization Upgrade Planning
   - Planning Considerations for System Downtime
   - Planning Considerations for Extranet and Intranet Deployments
   - Upgrade Paths
2. Review Chapter 2, "Upgrade Concepts and Methods" to gain a deeper understanding of the methods and strategies you will use, and to learn about any applications and components that have been deprecated (no longer officially supported).

3. Review Chapter 3, "About Automated Processes and Manual Tasks" to learn about the sequence of events that occur during the program-driven component upgrade, as well as what is preserved and what requires manual handling by you.
4. Investigate the functional differences between earlier releases and Oracle Access Manager 10*g* (10.1.4.0.1) in the centralized summary provided in Chapter 4, "System Behavior and Backward Compatibility".

> **Note:** During component upgrades, backward compatibility with earlier plug-ins and WebGates is automatically enabled. However, the system may behave differently than in earlier releases.

Planning Considerations

As you begin to plan for the upgrade in your environment, be sure to take the following considerations in to account:

- **Deployment Scenarios**: The upgrade task should be performed in a sequential order in relation to the deployment approach adopted in your organization: Identity System only versus Joint Identity and Access System; intranet versus extranet; number and type of installed environments.

  For example, if your earlier Identity System-only release is deployed only in an intranet environment with in three environments (Development, Test/Demonstration, and Production), the upgrade process should be performed and fine tuned in smaller environments before ultimately being performed in your production environment. For more information, see "Planning Considerations for Extranet and Intranet Deployments" on page 1-22.

- **Stability**: Each environment that you upgrade should currently be running a stable and appropriately installed release. In other words earlier Oracle Access Manager configurations in each existing environment must be confirmed to be stable and complete before you start upgrading.

  A good approach to validate that the existing environment is stable is to develop a deterministic test script and run it both before and after the upgrade. For example, the script could exercise a full end-to-end transaction by requesting a single page that requires authentication and authorization and a workflow request (all triggered by a single page request).

- **Administrative Access**: Schema upgrade operations (as well as other upgrade operations) require administrative access with write permissions to the directory server and Oracle Access Manager files.

- **Schema and Data Upgrades**: Preparing an earlier master Identity System (formerly known as the COREid System) and Policy Manager (formerly known as the Access Manager component) is a critical first step to performing a schema and data upgrade. For more information, see "Schema and Data Upgrade Planning" on page 1-12.

- **Customization Upgrades**: This is primarily a manual process. Oracle recommends that you complete any testing and alterations in a development environment before redeploying these in a shared or production environment. For more information, see "Customization Upgrade Planning" on page 1-13.

Schema and Data Upgrade Planning

The schema and data upgrade must be performed by someone with administrator privileges that include write access to the directory and files. The sequence of tasks you must perform to upgrade your earlier Oracle Access Manager schema and data to 10*g* (10.1.4.0.1) depend on the type of deployment you have (Identity System only or both the Identity and Access Systems).

The methodology for upgrading the schema and data is new and designed to help you ensure that the schema and data are properly upgraded before you start upgrading other components.

When you have only the Identity System deployed (with one or more Identity Servers and WebPass instances), you perform the schema and data upgrade, then complete other upgrade tasks as indicated in the next overview.

Task overview: Upgrading the schema and data when you have only an Identity System installed

1. Prepare and backup directory instances and data for the Identity System as described in Chapter 5, "Preparing for Schema and Data Upgrades".
2. Add an earlier instance of the following components to create a master environment for the schema and data upgrade:
   - One earlier Identity Server instance (formerly known as the NetPoint or COREid Server) as a secondary server for your original master read/write directory server instances. The directory server administrator will use this instance as the Master Identity Server when upgrading the schema and data.
   - One earlier WebPass to communicate with the master Identity Server you added

     For more information, see "Adding An Earlier Identity System to Use as a Master" on page 5-18.
3. Upgrade the added master Identity System components and accept the automatic schema and data upgrade in the sequence in the following list:
   - Upgrade the master Identity Server, schema, and data (then upload directory index files).
   - Upgrade the master WebPass (there is no schema nor data upgrade here).
   - Validate the Identity System schema and data upgrade.

     For more information, see Chapter 6, "Upgrading Identity System Schema and Data".
4. Prepare, then upgrade (and verify) remaining Identity System components, then integration components, then independently installed SDKs, and redeploy upgraded Identity System customizations in the sequence shown in Figure 1–5, "High-Level Upgrade Task Overview".

When your installation includes both the Identity *and* Access Systems, the overall process is a bit different as outlined in the next overview. In both cases, the directory server administrator will use the master environment that is added before upgrading the Identity and Access System schema and data. However, the sequence differs after that.

Task overview: Upgrading both the Identity and Access System schema and data

1. Prepare and backup directory instances and data for both the Identity and Access System as described in Chapter 5, "Preparing for Schema and Data Upgrades"
2. Add an earlier instance of the following components:
   - One earlier Identity Server instance (formerly known as the NetPoint or COREid Server) as a secondary server for your original master read/write directory server instances.
   - One earlier WebPass to communicate with the master Identity Server you added

     For more information, see "Adding An Earlier Identity System to Use as a Master" on page 5-18.
   - One earlier Policy Manager instance (formerly known as the Access Manager component) as a secondary server for your original master read/write directory server instances. For more information, see "Adding an Earlier Access Manager to Use as a Master" on page 5-24.
3. **Upgrade Identity Schema and Data**: Upgrade the added master components and accept the automatic schema and data upgrade in the sequence in the following list:
   - Upgrade the master Identity Server, schema, and data (then upload directory index files).
   - Upgrade the master WebPass (there is no schema nor data upgrade here).
   - Validate the Identity System schema and data upgrade.

     For more information, see Chapter 6, "Upgrading Identity System Schema and Data".
4. **Upgrade Access System Schema and Data**: Upgrade the added master component and accept the automatic Access System schema and data upgrade in the following sequence:
   - Upgrade the master Policy Manager, Access System schema and policy data, then upload directory index files.
   - Validate the Access System schema and data upgrade.

     For more information, see Chapter 7, "Upgrading Access System Schema and Data".
5. **Access System**: Create a temporary directory profile to provide write access to policy data by the Access Server for later WebGate upgrades, as described in "Creating a Temporary Directory Profile For Access System Upgrades" on page 7-10
6. Prepare, then upgrade (and verify) remaining Identity components, then Access System components, then integration components, then independently installed SDKs, and redeploy Identity System customizations then Access System customizations, as shown in Figure 1–5, "High-Level Upgrade Task Overview".

Customization Upgrade Planning

Customized configurations built around your earlier Oracle Access Manager installation must be manually tested for compatibility and upgraded for 10*g* (10.1.4.0.1). These include front-end customizations created using IdentityXML,

PresentationXML, and the Access Manager API (formerly known as the Access Server API or simply as the Access API). Also included are back-end customizations created with the Identity Event API, Authentication API, Authorization API (including AccessGates and plug-ins).

Testing and upgrading earlier customizations is primarily a manual process that may take some development time. It is important to plan ahead to ensure that your customizations can be redeployed into a shared environment quickly (for example, for QA, Integration, or Production).

Recommendation: Upgrading customizations and plug-ins

1. Start well in advance of other upgrades and review customization considerations in "Planning Considerations for System Downtime" on page 1-17.
2. Develop deterministic test scripts to run both before and after the upgrade to exercise a full end-to-end transaction.

   For example, the script could request a single page that requires authentication and authorization and a workflow request (all triggered by a single page request). Test scripts that verify the behavior of your earlier customizations help you ensure that these work as expected and produce the same result, both before and after the upgrade. Your test scripts will depend on the specific customization being exercised.
3. Compile and test the code, and the instructions you developed to explain how to configure the customization in a given environment.
4. In your existing environment, test the earlier customization (styles, AccessGates, or plug-ins for example) to ensure it is working as expected.
5. Install 10*g* (10.1.4.0.1) in a small development environment (ideally a *sandbox*-type setting) where the dependency on the overall Oracle Access Manager deployment is minimal. For details, see the *Oracle Access Manager Installation Guide*.
6. In the 10*g* (10.1.4.0.1) sandbox, test the earlier customization and perform any manual steps needed to upgrade the customization to operate with 10*g* (10.1.4.0.1) functionality.
7. Upgrade Oracle Access Manager in the test or development environment, as described in "Upgrade Task Overview" on page 1-6.
8. When the test or development environment upgrade is successful, you can redeploy the compiled binaries and custom components, then upgrade your production environment.

For information about specific customizations, see:

- Planning Considerations for System Downtime on page 1-17
- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"

Planning Deliverables

Planning deliverables include preparing a document where you define and record a detailed plan that identifies how the upgrade tasks will be performed within each environment. You can reduce the amount of system downtime by fine tuning the plan and tasks to meet the specific needs of each environment and to take into account the number of servers, downtime windows, and the like.

In addition, Oracle recommends that you prepare a detailed inventory of all earlier components and customizations. The details that you need to record for each component and the environment are described next. Planning worksheets that you can copy and fill in are provided in Appendix E, "Planning Worksheets and Tracking Checklists".

Task overview: Developing your planning deliverables

1. **Create a Planning Document**: Define and record a detailed plan identifying how the upgrade process will be performed for each environment. For more information, see also:
   - Planning Considerations for System Downtime on page 1-17
   - Planning Considerations for Extranet and Intranet Deployments on page 1-22
2. **Take Inventory of the Earlier Environment**: If you have not already recorded the exact details of the earlier environment, do so now and include both Access and Identity components, directory servers, Web servers, and applications as indicated next. Planning worksheets that you can copy and fill in are provided in Appendix E, "Planning Worksheets and Tracking Checklists".
   - **Environment Details**:

     Transport security mode; Simple, Cert, or Open
     Root CA details if certificate mode is used
     Any host definition type entries relevant to NetPoint (for example, /etc/host)

     Identity Server Inventory:
     Workflows, search bases and ACLs
     Object definition details for all objects managed through NetPoint, if possible
     Auditing configuration details
     Password policy configuration

     Access Server Inventory:
     Policy domains, authentication schemes, resource definitions, host identifiers
     Auditing configuration
     Directory profile information

   - **Application Tier Details** that will be impacted by the upgrade. For example:

     WebGate protected integration that uses Cookies or header variables (the impact on these should be minimal).
     Custom AccessGate integration created using the API, which may have a more noticeable impact.
     Applications exposing Oracle Access Manager Identity Portal Inserts (such as portals).
     Look carefully at these to ensure that "service temporary unavailable" pages can be displayed during the upgrade process when access to workflows is unavailable.
     Applications relying on IdentityXML have the highest impact, because the IdentityXML service may be unavailable altogether (it could be complicated to separate read-only calls from write calls and might be best to disable the entire application during the upgrade process)

- **Administration and Presentation tier details for each WebGate, WebPass, and Web server**:

  Web server type, version, operating system,
  WebPass or WebGate identifier and exact patch version of the binary (for example, 6.1.1.19 or 7.0.4.2)
  Exact Oracle Access Manager patch version (for example, 6.1.1.19 or 7.0.4.2)
  WebPass or WebGate installation directory
  Connection information between the component and corresponding Oracle Access Manager Server, including primary or secondary status and number of connections

- **Details for each and every AccessGate, WebGate, Policy Manager** (Policy Manager was formerly known as the Access Manager component), application server integration (such as WebLogic SSPI, WebSphere, and Oracle OC4J to name a few) such as:

  HTTP Cookie domain, preferred host name, cache timeout and size, failover threshold
  Inventory any IdentityXML client that has been custom developed
  Inventory any virtual IP and DNS aliases used to reference the WebPass or Web server farm protected with WebGate, such that it would be feasible to alter their definition in cases where staged upgrade of the web server components (WebPass and WebGate be planned/required)

- **Oracle Access Manager Server Tier (for each Identity and Access Server)**:

  Exact patch level (6.1.1.19 or 7.0.4.2, for example)
  Installation directory for the Identity or Access Server
  Installation directory for the associated WebPass or WebGate
  TCP port number for the service for example, port 6021)
  Host name (DNS) and Identity (formerly COREid) Server identifier
  For the Access Server note the status of the Access Management flag (on or off)
  Inventory any customizations performed
  Identify any Identity Event plug-ins
  For the Access Server, note any customized authentication or authorization plug-ins
  Record any PresentationXML and XSL stylesheet customizations
  Record any file-based changes such as changes in globalparams.xml or .lst files
  Are the Identity Server (and Access Server) configured to audit to files or a database
  For Unix systems, record the user name and group membership for the Identity Server (formerly known as the COREid Server)

- **Directory Server Tier**:

  Exact directory server version and patch level for example, Sun ONE v5.2 SP2)
  Directory server DNS name and Port
  Transport security mode: LDAP, LDAPS, ADSI
  Binding credentials used by Oracle Access Manager
  DIT and schema objects used in Oracle Access Manager
  Master/replica configuration details
  For more information, see "Schema and Data Upgrade Planning" on page 1-12

3. **Customization Assessment and Planning**: You must ensure that any custom developed plug-ins, Access Manager API clients, IdentityXML clients, PresentationXML customizations, Portal Inserts, and customized styles are compatible with Oracle Access Manager 10*g* (10.1.4.0.1). This is primarily a manual process. For more information, see:
   - Customization Upgrade Planning
   - Planning Considerations for System Downtime

Planning Considerations for System Downtime

During an upgrade in any environment, system downtime is inevitable. Oracle recommends that you pay special attention to planning and coordinating with any external party that may be directly or indirectly impacted during the upgrade.

Most Oracle Access Manager deployments provide a mission-critical element of the enterprise infrastructure by supporting applications. For example, suppose Oracle Access Manager is protecting access to an employee portal, and providing a registration service for new users. During the upgrade, new users may not be able to register. Moreover, there could be a window of time during which access to the portal and any of the protected applications is not available. This may present significant impact to end users.

This discussion provides information to help you determine the amount of downtime required for the upgrade process and manual upgrade tasks in your environment. There will be some disruption to the services provided by Oracle Access Manager during some portion of the process. However, you can take steps to minimize the overall amount of service downtime. For example, if Oracle Access Manager is deployed in three environments: Development, Test/Demonstration, and Production, then upgrade tasks should be first tried in Development and fine tuned before ultimately being performed in production.

Recommendation: Upgrading each deployment in your environment

1. Perform the entire upgrade task, illustrated next, in your Development environment.
2. When your Development environment is successfully upgraded and confirmed to be running properly, perform the entire upgrade task again in your Test/Demonstration environment.
3. When your Test/Demonstration environment upgrade is successfully completed and running properly, you perform the entire upgrade task in your Production environment.

This approach helps you gauge the time it takes to perform the upgrade in your environment and ensure that all customizations and plug-ins are working properly before you start upgrading in a production environment. This also helps ensure that your production environment upgrade will go smoothly and quickly with fewer service interruptions.

The emphasis is on reducing the impact on availability of Oracle Access Manager (formerly Oblix NetPoint or Oracle COREid) service during the upgrade. One goal of this approach is to identify tactics that can help reduce overall upgrade time and minimize service impact. As you perform upgrade tasks within each environment, you can develop strategies and optimizations that significantly streamline the overall task.

Oracle recommends careful planning to minimize the operational impact of upgrading your earlier environment. Oracle cannot guarantee that a service outage is not required to complete the upgrade.

When planning the upgrade for each environment, it is important to take into account the criticality and number of applications that depend on Oracle Access Manager. This may increase with each environment. Pay special attention to coordinating the change process that the upgrade represents to the environment as a whole. It is important to work with the application owners to ensure that end user impact is properly managed. Standard procedures such as a change control process, scheduled maintenance windows, off hours operation windows, and others should be considered when planning the Oracle Access Manager upgrade.

When assessing the impact of the Oracle Access Manager upgrade, take inventory and categorize the various applications that depend on Oracle Access Manager. This can include applications protected by the Access System, or applications that leverage the Identity System for identity administration functions, as well as the impact on the underlying directory environments. Directory environments are particularly important because the upgrade process requires a directory schema update. In many environments, upgrading the directory schema is a highly privileged operation handled by a directory administration group.

As you take inventory and categorize the various applications, be sure to estimate potential outage windows for each application. This will help set and manage end-user expectations. The estimated duration of outage windows will vary depending on the type of application (whether it is Access System or Identity System dependent) and the estimated duration of the upgrade tasks. For the production environment, estimates can be extrapolated from the experience gained when performing upgrade tasks and fine tuning in your Development and Testing/Demonstration environments.

For more information, see:

- Minimizing Downtime During the Upgrade
- Downtime Assessments
- Downtime Assessment Example

Minimizing Downtime During the Upgrade

The upgrade process will require some downtime of enterprise applications that rely on Oracle Access Manager for identity administration, authentication, and authorization. There are a few upgrade tasks that can occur without impacting these applications. Table 1–1 outlines lists the upgrade tasks, their downtime impact, and planning considerations to minimize downtime where applicable.

Table 1–1 Minimizing Downtime

Upgrade Task	Downtime Impact	Steps to Reduce Downtime
Upgrade Planning	None	N/A. Review the planning chapters, fill in the worksheets as you take inventory of your environment to reduce the probability of human errors, use the checklists to track progress as you complete upgrade tasks.
Preparing for Schema and Data upgrades	None	N/A
Upgrading the Schema and Data	All Oracle Access Manager servers are down and all consumers of Oracle Access Manager are impacted.	Make backups and be prepared with recovery procedures in case of problems. Validate the process in stage environment before trying in production.

Table 1–1 (Cont.) Minimizing Downtime

Upgrade Task	Downtime Impact	Steps to Reduce Downtime
Upgrading Oracle Access Manager components	All Oracle Access Manager servers are down and all consumers of Oracle Access Manager are impacted.	Make backups and be prepared with recovery procedures in case of problems. Validate the process in stage environment before trying in production.Validate the upgrade in a staging environment before upgrading in production.
Upgrading Third-Party Integration Connectors	Only those third-party environments in which the deployment has chosen to upgrade the connectors.	Validate the upgrade in a staging environment before upgrading in production. Consider that Oracle Access Manager is backward compatible with earlier environments, as described in Chapter 4, which may be exploited to minimize downtime.
Upgrading Independently Installed SDKs	Only those environments where an independently installed SDK must be upgraded are impacted.	Validate the upgrade in a staging environment before upgrading in production. Consider that Oracle Access Manager is backward compatible with earlier environments, as described in Chapter 4, which may be exploited to minimize downtime.
Upgrading Customizations	Deployment services that rely on Identity or Access System customizations are impacted.	Apply customizations in a staging environment first, to resolve issues. Then apply them in a production environment.
Validating the Upgrade	None	N/A

Downtime Assessments

Perhaps the greatest amount of time spent in upgrading occurs during the planning process, which occurs offline. Careful planning can help reduce the overall amount of downtime needed to upgrade each environment in your enterprise.

The second greatest amount of time spent in the upgrade task occurs when preparing for the schema and data upgrade.

Less time will be spent actually upgrading components. Depending on your deployment and the amount of customization, some time must be allotted for any manual tasks needed to ensure that your earlier customizations are compatible with 10*g* (10.1.4.0.1) and are successfully redeployed. However, customizations can be handled outside the shared environment and have minimal impact on system downtime.

The following considerations are provided to help you understand the overall upgrade impact and downtime in your environment. Additional information is provided in "Downtime Assessment Example" on page 1-20.

- Planning and taking inventory of your existing environment, as discussed in this chapter and other chapters in Part I, "Introduction", is a zero downtime activity. Careful planning can actually help reduce the amount of system downtime during actual upgrade tasks.
- Schema and data upgrades (introduced in "Schema and Data Upgrade Planning" on page 1-12 and described in Part II, "Upgrading the Schema and Data") will take the greatest amount of time, and includes:
  - Preparing your LDAP directory instances and data
  - Making backup copies of all data, installation directories, and Windows registry entries that include Oracle Access Manager information before the upgrade

  - Preparing a master system to use during the actual schema upgrade
  - Performing the actual schema upgrade
  - Performing the actual data upgrade, which depends upon the number of workflows and workflow steps and the number of access policies, domains, and protected resources
  - Verifying that the schema and data upgrade was successful
- Preparing and upgrading all other Oracle Access Manager components, and Web server configuration upgrades, as described in Part III, "Upgrading Components", for the most part does not require system downtime.
- Manually processing customized stylesheets, plug-ins, forms-based authentication, audit to database implementations, and the like, as described in Part IV, "Upgrading Your Customizations", can be performed outside the shared environment (which greatly reduces the amount of system downtime required)
- Verifying that the upgrade was a success, as discussed in Part V, "Validating the Upgrade" does not result in system downtime.

Downtime Assessment Example

The following estimates are provided to give you an idea of the amount of time it takes to upgrade an earlier deployment that includes approximately 100 workflows, 500 policy domains, 2500 access policies, and 1700 protected resources.

Identity System Downtime Assessment

- **Planning and Taking Inventory (of the currently installed environment)**: Zero downtime. This task is performed outside the environment, before the upgrade. For more information, see "Planning Deliverables" on page 1-14.
- **Preparing for the Schema and Data Upgrade**: ~1 hour and includes:
  - Developing Strategies for Upgrading in a Replicated Environment
  - Configuring the Challenge/Response Phrase at the Object Class Level
  - Configuring Unique Namespaces for Directory Connection Information
  - Preparing Your Directory Instances for the Schema and Data Upgrade
  - Preparing Host Machines for Master Components
  - Adding An Earlier Identity System to Use as a Master
- **Directory Server Backups**: ~15 to 30 minutes For more information, see "Backup and Recovery Strategies" on page 2-3.
- **File system Backups**: ~15 minutes. For more information, see "Backup and Recovery Strategies" on page 2-3.
- **Schema Upgrade**: ~20 minutes
- **Data Upgrade**: ~1.5 hours
- **Identity System Component Upgrades**: ~5 minutes for each component (Identity Server and WebPass) instance, which includes parameter/message upgrades and Web server configuration upgrades.
- **Identity System Customization Upgrades (Zero Downtime)**: The following manual tasks can be performed ahead of the production environment upgrade and

outside the shared environment. As a result, there is no system downtime for these activities:

  - Install and set up a fresh 10*g* (10.1.4.0.1) Identity System to use when testing and upgrading customizations, as described in the *Oracle Access Manager Installation Guide*.
  - **Auditing and Access Reporting**: Create a new database instance for use with 10*g* (10.1.4.0.1), as described in "Upgrading Auditing and Access Reporting for the Identity System" on page 12-2.
  - **Portal Inserts**: "Ensuring Compatibility with Earlier Portal Inserts" on page 12-11.
  - **Custom Identity Event Plug-ins**: Redesign and recompile custom plug-ins as described in "Migrating Custom Identity Event Plug-Ins" on page 12-10.
  - **Styles**: Incorporate stylesheet/javascript/msgcatalog/gif customizations from prior releases, as described in Chapter 12, "Upgrading Your Identity System Customizations".
  - **Validation**: Use procedures in "Validating Identity System Customization Upgrades" on page 12-23.

- **Identity System Customization Redeployment**: ~30 minutes. If you perform the manual customization tasks in the preceding list before upgrading, you need only copy the required files to appropriate directories after upgrading components. This should significantly reduce the amount of downtime during the production environment upgrade.
- **Identity System Customization After Upgrading:** ~1 hour:
  - Finishing Upgrading Auditing and Access Reporting for the Identity System on page 12-2.
  - Combining Challenge and Response Attributes on a Panel on page 12-8.
  - Confirming Identity System Failover and Load Balancing on page 12-9
  - Validating Identity System Customization Upgrades on page 12-23
- **Identity System, Total Downtime**: ~5 hours

Access System Downtime Assessment

- **Planning and Taking Inventory (of the current installed environment)**: Zero downtime. This task is performed outside the environment, before the upgrade. For more information, see "Planning Deliverables" on page 1-14.
- **Preparing for the Access System Schema and Data Upgrade**: ~1 hour and includes:
  - Developing Strategies for Upgrading in a Replicated Environment
  - Configuring the Challenge/Response Phrase at the Object Class Level
  - Configuring Unique Namespaces for Directory Connection Information
  - Preparing Your Directory Instances for the Schema and Data Upgrade
  - Preparing Host Machines for Master Components
  - Adding an Earlier Access Manager to Use as a Master
- **Directory Server Backups**: ~15 to 30 minutes. For more information, see "Backup and Recovery Strategies" on page 2-3.

- **File system Backups**: ~15 minutes. For more information, see "Backup and Recovery Strategies" on page 2-3.
- **Access System Schema Upgrade**: ~20 minutes
- **Access System Data Upgrade**: ~2 hours
- **Access System Component Upgrades**: ~5 minutes for each component instance (Policy Manager, Access Server, WebGate), which includes parameter/message upgrades and Web server configuration upgrades.
- **Access System Customization Upgrades (Zero Downtime):** Several manual tasks can be performed ahead of the production environment upgrade and outside the shared environment. As a result, there is no system downtime for these activities:
  - Install and set up a fresh 10*g* (10.1.4.0.1) Access System to use when testing upgraded customizations, as described in the in the *Oracle Access Manager Installation Guide*.
  - **Auditing and Access Reporting**: Complete this task for the Access Server, as described in "Upgrading Auditing and Reporting for the Access Server" on page 13-2.
  - **Forms-based Authentication**: Perform activities in "Upgrading Forms-based Authentication" on page 13-4.
  - **Custom Authentication and Authorization Plug-ins**: "Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins" is described on page 13-5.
  - Updating the ObAMMasterAuditRule_getEscapeCharacter in Custom C Code is described on page 13-7.
  - **Validation**: Validating Access System Customization Upgrades is described on page 13-7.
- **Access System Customization Redeployment**: ~30 minutes. If you perform the manual customization tasks in the preceding list before upgrading, you need only copy the required files to appropriate directories after upgrading components. This should significantly reduce the amount of downtime during the production environment upgrade.
- **Access System Customization After Upgrading:** ~1 hour:
  - Finishing "Upgrading Auditing and Reporting for the Access Server" on page 13-2.
  - Associating Release 6.1.1 Authorization Rules with Access Policies on page 13-5
  - Assuring Proper Authorization Failure Re-directs After Upgrading from 6.1.1 on page 13-6
  - Validating Access System Customization Upgrades on page 13-7
- **Access System Total**: ~5.5 hours of downtime

Therefore, to upgrade both the Identity and Access Systems in this environment will take about 11 hours for tasks that require system downtime.

Planning Considerations for Extranet and Intranet Deployments

Existing earlier Oracle Access Manager deployments can be classified into two primary categories: Extranet (B2B,G2C, B2C) and Intranet (B2E, G2E) deployments.

These are, of course, generic categories. However, for the purposes of understanding deployment demographics these should provide relevant patterns.

For more information, see the topics:

- Extranet Deployments
- Intranet Deployments

Extranet Deployments

Extranet deployments are those where you have:

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

The most typical characteristics for extranet deployments include:

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

Performing the upgrade to 10*g* (10.1.4.0.1) with minimal service disruption in such a highly complex environment can be challenging.

Intranet Deployments

Intranet deployment environments are typically:

- Internal facing portals with a relatively small user population (less than 20 thousand users)

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

The most typical characteristics for intranet deployments include:

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

Upgrade Paths

This discussion introduces the paths available when upgrading from an earlier release to Oracle Access Manager 10*g* (10.1.4.0.1). The path available to you depends upon your starting release (the release from which you are starting the upgrade) as described in:

- Direct Upgrade Paths
- Indirect Upgrade Paths

Direct Upgrade Paths

There are several direct paths available to bring an earlier release up to Oracle Access Manager 10*g* (10.1.4.0.1), as described in following discussions:

- From Release 6.1.1 to Oracle Access Manager 10g (10.1.4.0.1)
- From Release 6.5 to Oracle Access Manager 10g (10.1.4.0.1)
- From Release 7.x to Oracle Access Manager 10g (10.1.4.0.1)

From Release 6.1.1 to Oracle Access Manager 10*g* (10.1.4.0.1)

Release 6.1.1 deployments are typically large in terms of the number of components deployed at each tier of the architecture, as well as other systems and applications. Oracle Access Manager 6.1.1 is an English only release.

If you are upgrading from release 6.1.1, you may use the Oracle Access Manager 10*g* (10.1.4.0.1) installers to perform a direct upgrade. During the upgrade, each component installer automatically implements product changes for each release between release 6.1.1 and Oracle Access Manager 10*g* (10.1.4.0.1). This includes automatically enabling the multi-language capability available in Oracle Access Manager 10*g* (10.1.4.0.1).

Every environment requires some preparation before starting the upgrade, as discussed in Chapter 5, "Preparing for Schema and Data Upgrades" and Chapter 8, "Preparing Components for the Upgrade". There are no additional caveats and conditions when upgrading directly from release 6.1.1.

From Release 6.5 to Oracle Access Manager 10*g* (10.1.4.0.1)

Release 6.5.0 introduced multi-language support for French and German in addition to providing and enabling English language messages. When you have any Oracle Access Manager 6.5 release, you use the Oracle Access Manager 10*g* (10.1.4.0.1) installers to upgrade directly as described in this manual.

Table 1–2 discusses the various 6.5 releases.During the direct upgrade from any Oracle Access Manager 6.5.x release, each component installer automatically implements product changes for each release between release 6.5.0 and Oracle Access Manager 10*g* (10.1.4.0.1).

To retain earlier multi-language functionality, you must include 10*g* (10.1.4.0.1) Language Packs in the same directory as the 10*g* (10.1.4.0.1) installation package that you use to upgrade the component. Otherwise, only the English language is used. You may install additional supported languages after the upgrade, as described in the *Oracle Access Manager Installation Guide*.

Table 1–2 Upgrade Paths from Oracle Access Manager 6.5 Releases

Starting From	Upgrading To	Caveat
6.5.0.x is an international release (English, German, French)	10*g* (10.1.4.0.1)	Before the upgrade, complete activities in "Adding Packages for Release 6.5.0.x" on page 8-5. See also, "Preparing Multi-Language Installations" on page 8-6.
6.5.1 is an English-only release that introduced support for Active Directory Application Mode (ADAM) as a back end directory.	10*g* (10.1.4.0.1)	No caveats or special requirements. Upgrade as described in this guide.
6.5.2.x is an *English-only* release	10*g* (10.1.4.0.1)	Before upgrading an installation patched to 6.5.2, complete activities in "Adding Packages for Release 6.5.2.x Patch" on page 8-5.
6.5.3.x is an *English-only* WebGate release	Use 10*g* (10.1.4.0.1)	No caveats. Upgrade as described in this guide.

From Release 7.x to Oracle Access Manager 10*g* (10.1.4.0.1)

With the exception of release 7.0.4 (which is an international release that was available as part of Oracle Application Server 10g Release 2 (10.1.2)), all 7.x releases are English only. Typically, NetPoint 7.x environments are newer and less complex than NetPoint 6.5 or 6.1.1 environments.

Table 1–3 provides a brief overview of 7.x releases. During the direct upgrade from any 7.x release, each component installer automatically implements product changes for each release between 7.0 and Oracle Access Manager 10*g* (10.1.4.0.1) and enables multi-language capability. To retain earlier multi-language functionality (or to install new languages), you can include 10*g* (10.1.4.0.1) Language Packs in the same directory as the 10*g* (10.1.4.0.1) installation package. Otherwise, only the English language is used.

Table 1–3 Upgrade Paths from Series 7.x Releases

Starting From	Upgrading To	Caveat
Release 7.0	10*g* (10.1.4.0.1)	No caveats. Upgrade as described in this guide and include Language Packs if desired to upgrade languages.
Release 7.0.1, and later, provide additional platform certifications and parameter and message updates. Going forward, new GIFs, XSL, HTML, images, or similar files can be included in a patch release.	10*g* (10.1.4.0.1)	No caveats. Upgrade as described in this guide and include Language Packs if desired to upgrade languages.

Indirect Upgrade Paths

If you are upgrading from any Oracle Access Manager release earlier than 6.1.1, no direct upgrade path is available to Oracle Access Manager 10*g* (10.1.4.0.1). In this case, an intermediate upgrade from your earlier release to release 6.1.1 is required. From release 6.1.1, you can upgrade directly to Oracle Access Manager 10*g* (10.1.4.0.1).

Table 1–4 lists the various starting point scenarios and associated caveats for an intermediate upgrade.

Table 1–4 Upgrade Paths from Release 5.x through 6.1

Starting From	Upgrading To	Caveats and Conditions
Release 5.2	Release 6.1	To retain Publisher, you can upgrade only to Oracle Access Manager 6.1.
	Release 6.1.1	If you abandon Publisher you may complete an intermediate upgrade to 6.1.1.
	Oracle Access Manager 10*g* (10.1.4.0.1)	From release 6.1.1 you may upgrade directly to Oracle Access Manager 10*g* (10.1.4.0.1). For information on the intermediate upgrade, contact Oracle Support at `http://www.oracle.com/support/contact.html`.
Release 6.0	Release 6.1	To retain Publisher, you can upgrade only to Oracle Access Manager 6.1.
	Release 6.1.1	If you abandon Publisher you may complete an intermediate upgrade to 6.1.1.
	Oracle Access Manager 10*g* (10.1.4.0.1)	From release 6.1.1 you may upgrade directly to Oracle Access Manager 10*g* (10.1.4.0.1). For information on the intermediate upgrade, contact Oracle Support at `http://www.oracle.com/support/contact.html`.

Table 1–4 (Cont.) Upgrade Paths from Release 5.x through 6.1

Starting From	Upgrading To	Caveats and Conditions
Release 6.1	Release 6.1	To retain Publisher, no further upgrade is possible.
	Release 6.1.1	If you abandon Publisher you may complete an intermediate upgrade to 6.1.1.
	Oracle Access Manager 10*g* (10.1.4.0.1)	From release 6.1.1 you may upgrade directly to Oracle Access Manager 10*g* (10.1.4.0.1). For information on the intermediate upgrade, contact Oracle Support at `http://www.oracle.com/support/contact.html`.

Specific details of the intermediate upgrade from earlier releases to release 6.1.1 are outside the scope of this manual. Before you start upgrading from a release *earlier* than 6.1.1, contact Oracle Support at:

`http://www.oracle.com/support/contact.html`

2

Upgrade Concepts and Methods

This chapter introduces upgrade concepts, strategies, and methods. Topics include:

- Upgrade Terms and Concepts
- About Upgrading the Oracle Application Server
- Backup and Recovery Strategies
- Upgrade Start Methods
- Upgrade Event Modes
- Support Deprecated
- Upgrade Strategies When Support is Changed or Deprecated

> **Note:** There are several important name changes that you should know about. Be sure to review "Product and Component Name Changes" on page -xxi. This manual uses the new names, even when referring to earlier releases.

For an introduction to Oracle Access Manager, a road map to related manuals, and a glossary of terms, see the *Oracle Access Manager Introduction*.

Upgrade Terms and Concepts

The latest release provides significant enhancements and regulatory compliance over previous releases. For example, each major release provides new features and additional platform support, and may include changes to the schema, data, parameter, or message files.

The term *upgrade* refers to the process of installing the latest major product release over an earlier product release (whether the earlier release has been patched or not). This is known as an in-place upgrade.

Your existing data and configurations are made available to the new release. For example, suppose you have installed Oracle Access Manager 6.1.1 and added new object classes and panels; assigned or delegated administrative rights to key people; created workflows; protected resources with a policy domain; configured authentication schemes and authorization rules; customized the way the product looks or operates; and modified message files. After upgrading to 10*g* (10.1.4.0.1), you do not need to replicate all the work you had completed on the earlier release. However, certain items must be handled manually.

Separate platform-specific packages are provided for each component. The same package is used to both install 10*g* (10.1.4.0.1) and to upgrade to 10*g* (10.1.4.0.1). For example:

Windows: Oracle_Access_Manager10_1_4_0_1_win32_*Component*.exe
Solaris: Oracle_Access_Manager10_1_4_0_1_sparc-s2_*Component*

Release 10*g* (10.1.4.0.1) uses the Oracle product numbering scheme, where a major release is identified by the *first three numbers* (10.1.4, for example) of a *five segment product number*. The last two digits of an Oracle product number represent the maintenance and patch release numbers (10.1.4.0.0), respectively:

Oracle Release Numbering: 10.1.4.0.1
n . n . n (Major) . Maintenance . Patch

A patch release is one that provides fixes to known problems. The upgrade process handles the differences between release numbering schemes seamlessly. During the upgrade, a minor release (or maintenance release or patch release) is recognized as a major release. Your earlier installation may include patches. However, you do not need to apply patches to the earlier installation before upgrading to 10*g* (10.1.4.0.1).

Earlier release numbers consisted of four elements:

Earlier Release Numbering: 7.0.4.2
Major_release . Minor_release . Maintainence_release . Patch_release

---

Note: You do not need to apply any patch to an earlier installation before upgrading. To upgrade from releases earlier than 6.1.1, contact Oracle Support at `http://www.oracle.com/support/contact.html`.

The program controls the upgrade process, which completes incrementally for each major release. Before 10*g* (10.1.4.0.1), major and minor release numbers were used during incremental upgrade steps. In the sample process overview, the starting release is 6.1.1. While your environment may vary, the incremental upgrade from each major release to 10*g* (10.1.4.0.1) occurs automatically during each component upgrade.

Process overview: Automatic Incremental upgrades

1. The first increment brings your installation from release 6.1.1 to release 6.5.
2. The second increment brings your installation from release 6.5 to release 7.0.
3. Third increment brings your installation from release 7.0 to 10*g* (10.1.4.0.1).

About Upgrading the Oracle Application Server

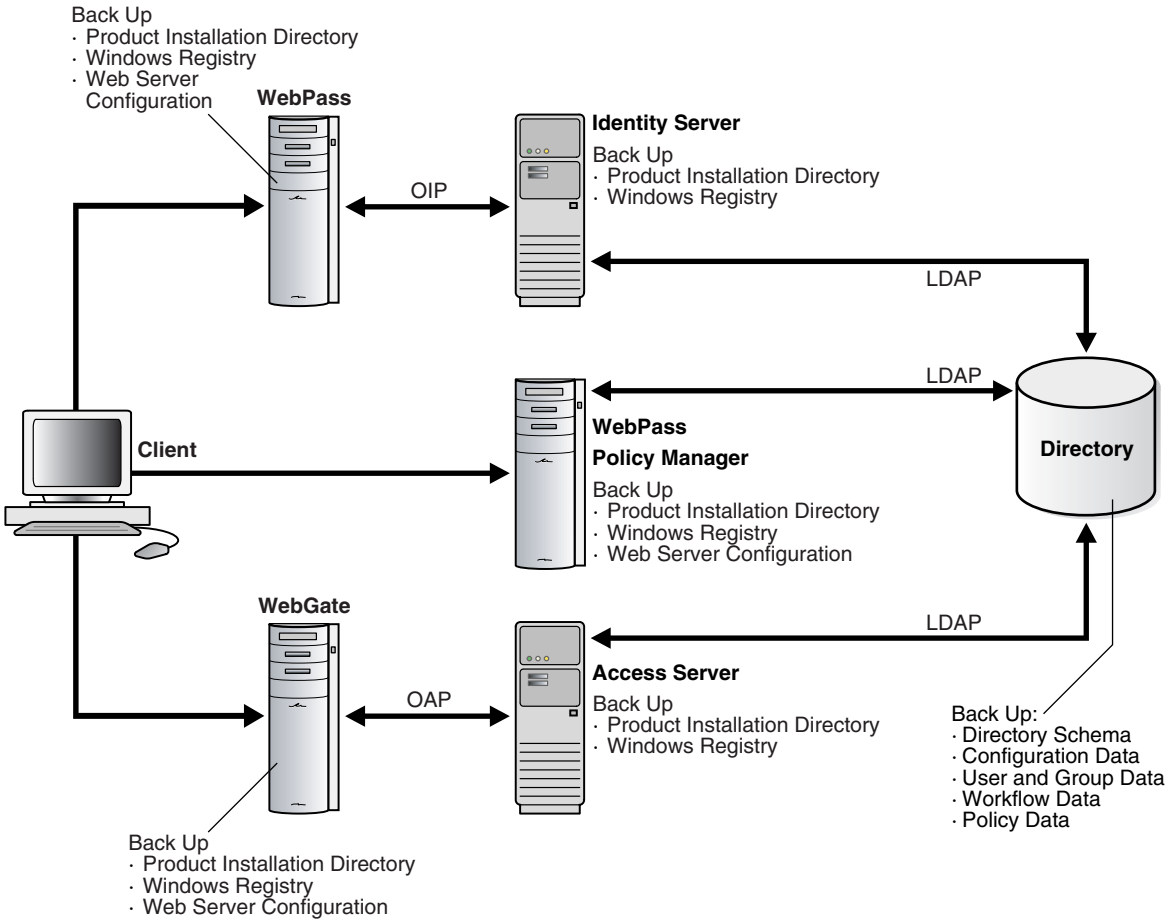
If your system environment includes Oracle Application Server components, you can upgrade these to 10*g* (10.1.4.0.1) by following the instructions in the *Oracle Application Server Upgrade and Compatibility Guide*.

The upgrade procedures for Oracle Access Manager are documented separately from the Oracle Application Server upgrade procedures because the two products are installed separately.

Backup and Recovery Strategies

In any environment, it is important to make back up copies before and after upgrading. Figure 2–1 illustrates the types of information that Oracle recommends you back up. Unless you have the Access System currently installed, ignore details for the Policy Manager, Access Server, and WebGate.

Figure 2–1 Back Up Strategies



As discussed in Chapter 1, the product installation directory for each component includes the following information types:

- Program and library files
- Message and parameter catalogs
- Component-specific configuration files (and in some cases failover configuration files and the software developer kit (SDK) configurations)

A WebPass must also be installed with each Policy Manager on the same Web server instance, at the same directory level.

For more information, see:

- Backup Strategies Before Upgrading
- Backup Strategies After Upgrading
- Recovery Strategies

Backup Strategies Before Upgrading

Oracle recommends that you perform certain back up activities before upgrading to help restore an earlier environment in the unlikely event that you want to do this following an upgrade. Table 2–1 provides more information.

Table 2–1 Back Up Strategies Before Upgrading

Back Up the Following	As Described In
Oracle Access Manager Schema	Backing up the Earlier Oracle Access Manager Schema
Oracle Access Manager Configuration and Policy Data	Backing up Oracle Access Manager Configuration and Policy Data
Oracle Access Manager User and Group Data	Backing Up User and Group Data
Oracle Access Manager Workflow Data	Backing Up Workflow Data
Processed Workflows	Archiving Processed Workflow Instances
Existing Directory Instances	Backing Up Existing Directory Instances
Earlier Installed Component Directory (and any Customization Directories)	Backing Up the Existing Installed Directory
Web Server Configuration Files	Backing Up the Existing Web Server Configuration File
Windows Registry	Backing Up Windows Registry Data

Backup Strategies After Upgrading

After you have completed and verified each component upgrade, Oracle recommends that you back up the upgraded information in Table 2–2. This will enable you to restore an upgraded environment to the newly upgraded status should this be needed.

Table 2–2 Back Up Strategies After Upgrading

Back Up the Following	As Described In
Existing Directory Instances	Backing Up Existing Directory Instances
Earlier Installed Component Directory (and any Customization Directories)	Backing Up the Existing Installed Directory
Web Server Configuration Files	Backing Up the Existing Web Server Configuration File
Windows Registry	Backing Up Windows Registry Data

Recovery Strategies

Should something unlikely occur and you find that a process did not complete successfully, you may use the strategies in Table 2–3 to recover.

Table 2–3 Upgrade Recovery Strategies

Task	What to do If the Task Fails
Backing Up Existing Oracle Access Manager Data	Retry this task using instructions in Chapter 5, "Preparing for Schema and Data Upgrades"
Backing Up Existing Directory Instances	See your directory vendor documentation.
Adding An Earlier Identity System to Use as a Master (against Read/Write master directory instances, not against read-only replicas) Note: You use this additional earlier setup as a master when upgrading the schema and data to ensure that your existing installation is not affected should any issues arise.	Retry this task using instructions in Chapter 5, "Preparing for Schema and Data Upgrades"

Table 2–3 (Cont.) Upgrade Recovery Strategies

Task	What to do If the Task Fails
Adding an Earlier Access Manager to Use as a Master (against Read/Write master directory instances, not against read-only replicas) Note: You use this additional earlier setup as a master when upgrading the schema and data to ensure that your existing installation is not affected should any issues arise.	Retry this task using instructions in Chapter 5, "Preparing for Schema and Data Upgrades"
Upgrading Identity System Schema and Data	Restore the directory instance you backed up before starting this upgrade (see "Backing Up Existing Directory Instances" on page 5-18). Locate your backup copy of the earlier master Identity Server installation directory (made before the upgrade) and make another backup copy. You will retain one to as a backup and use the other when you retry the upgrade. See "Backing Up Directories, Web Server Configurations, and Registry Details" on page 8-8. Retry the upgrade of the master Identity Server using instructions in Chapter 6, "Upgrading Identity System Schema and Data".
Enabling Multi-Language Capability when upgrading the master Identity Server from a starting release of 6.1.1. Note: This process does not occur when your starting release is 6.5 or 7.x because those releases automatically supported multi-language capability.	Restore the directory instance you backed up before starting this upgrade (see "Backing Up Existing Directory Instances" on page 5-18). Locate your backup copy of the earlier master Identity Server installation directory (made before the upgrade) and make another backup copy. You will retain one to as a backup and use the other when you retry the upgrade. See "Backing Up Directories, Web Server Configurations, and Registry Details" on page 8-8. Retry the upgrade of the master Identity Server using instructions in Chapter 6, "Upgrading Identity System Schema and Data".
Upgrading Access System Schema and Data	Restore the directory instance you backed up before starting this upgrade (see "Backing Up Existing Directory Instances" on page 5-18). Locate your backup copy of the earlier master Access Manager installation directory (made before the upgrade) and make another backup copy. You will retain one to as a backup and use the other when you retry the upgrade. See "Backing Up Directories, Web Server Configurations, and Registry Details" on page 8-8. Retry the upgrade of the master Access Manager using instructions in Chapter 7, "Upgrading Access System Schema and Data".
Uploading Directory Server Index Files	Retry this task using instructions in "Uploading Directory Server Index Files" on page 6-17.
Upgrading Components: Upgrading an earlier version of any Oracle Access Manager component (Identity Server, WebPass, Policy Manager (formerly known as the Access Manager component)), Access Server, or WebGate). Note: Schema and data upgrades occur only when upgrading master components added for this purpose.	Locate your backup copy of the earlier component installation directory (made before the upgrade) and make another backup copy. You will retain one to as a backup and use the other when you retry the upgrade. See "Backing Up Directories, Web Server Configurations, and Registry Details" on page 8-8. Retry this step and specify the earlier component installation directory when asked for the installation directory. See Part III, "Upgrading Components".
Upgrading Your Identity System Customizations	Retry this task using instructions in Chapter 12, "Upgrading Your Identity System Customizations"
Upgrading Your Access System Customizations	Retry this task using instructions in Chapter 13, "Upgrading Your Access System Customizations"

Additional information on recovering from an upgrade failure can be found throughout this book and in Appendix F, "Troubleshooting the Upgrade Process".

Upgrade Start Methods

As mentioned earlier, you use the corresponding 10*g* (10.1.4.0.1) component installer to begin each component upgrade. You may launch the installer using either the graphical user interface (GUI method) or the command-line interface (Console method).

Either Method: Regardless of the method you choose, the sequence of events and prompts are nearly identical. In later chapters, minor differences are identified as they occur during a specific sequence. If you see something that does not apply to your environment or installation, you may ignore it. Also, whether you launch the upgrade using GUI or Console method, you will be asked to choose a mode (Automatic versus Confirmed), as described in "Upgrade Event Modes" on page 2-6.

For more information, see:

- GUI Method
- Console Method

GUI Method

This method is the default for Windows systems when you select the installation package from the file system. For example:

Oracle_Access_Manager10_1_4_0_1_win32_Identity_Server.exe

Console Method

The command-line method (also known as Console method) is the default for Unix systems. For example:

```
./ Oracle_Access_Manager10_1_4_0_1_sparc-s2_Identity_Server
```

Upgrade Event Modes

Whether you launched the program using the GUI method or the Console method, after you accept the upgrade you are presented with the following choices and asked to select either Automatic mode or Confirmed mode:

```
-------------------------------------
Please specify the mode for migration:
'1' - Automatic (recommended)
          Each step is performed automatically.
          No interaction from the user is required.
'2' - Confirmed
       Each step needs confirmation from the user.
Enter choice ( '1' or '2' ) :
-----------------------------------------
```

For more information, see:

- Automatic Mode
- Confirmed Mode

Automatic Mode

Oracle recommends that you choose the Automatic mode. This mode provides declarative messages to keep you informed as the upgrade progresses. For example:

```
Creating original folders ...
-----------------------------------------------------
Copying general configuration files
OK.
-----------------------------------------------------
Updating parameter catalogs ...
OK.
-----------------------------------------------------
```

From time to time in Automatic mode, you are asked to respond to specific queries that require your acceptance before being initiated (or simply acknowledging that you are ready to continue). For example when upgrading the master Identity Server and master Access Manager, you are asked to accept an automatic schema and data upgrade as indicated in this example:

```
Oracle Access Manager schema migration ...

Retrieving Oracle configuration parameters ...
OK.
   The following directory server's schema will be updated:
      Host:DNShostname.domain.com
      Port: port#
      Type:ns
   NOTE:  If you do not want to migrate schema at this time,
          type 'SKIP'.
   Please type 'Yes" to proceed:
```

For more information about the sequence of events, see Chapter 3, "About Automated Processes and Manual Tasks".

> **Note:** In both Automatic and Confirmed mode, you are informed as the program completes each step of the upgrade process. This guide provides information using Automatic mode, both for brevity and because this is the recommended method.

Confirmed Mode

If you select Confirmed mode during a component upgrade, you are presented with a question before each and every event (not just those that require acceptance during Automatic mode). The types of messages you see in Confirmed mode are shown here:

```
Copy general configurations files?
    '1' - Yes
    '2' - No
Enter choice ( '1' or '2' ) : 1
OK.
```

In Confirmed mode, each event is performed only after you accept by entering the number 1. If you enter the number 2, the event is skipped and you are then asked to accept or decline the next event.

Confirmed mode is recommended for use in only the following situations when you need to conditionally run, skip, and re-run certain event in a component upgrade. For example:

- **Upgrade Strategies When Support is Changed or Deprecated**: Suppose a release 6.1.1 WebPass resides on a machine with a Web server version that is not supported by 10*g* (10.1.4.0.1). In this case, you must upgrade the 6.1.1 WebPass incrementally, as follows.

  - Retain the Web server version that is supported for the release 6.1.1 WebPass.
  - After initiating the WebPass upgrade and selecting Console mode, you accept activities to upgrade the WebPass from the release 6.1.1 to the next Oracle Access Manager release that supports the existing Web server version (to release 6.5 for example). You decline activities to upgrade WebPass further and accept updating the Web server configuration with 6.1.1 to 6.5 information.
  - After the incremental WebPass upgrade, you must upgrade the Web server to a version that is supported by the next WebPass release using your vendor documentation as a guide.
  - After upgrading the Web server, you complete another incremental WebPass upgrade in Console mode by skipping the release 6.1.1 to 6.5 events that were already completed and continuing to the next component release that supports the existing Web server.

    For more information, see "Upgrade Strategies When Support is Changed or Deprecated" on page 2-9.

- **Correcting Information**: Suppose you provide incorrect information during an component upgrade (or another problem arises). In this case, you may also use Confirmed mode to conditionally re-run a step. For example, suppose you entered incorrect information while upgrading the Identity Server. When the upgrade finishes, you can re-run it in Confirmed mode to skip events that completed successfully the first time (and enter correct information for unsuccessful events the second time). For instance, if you forgot to change the schema domain, you can re-run the upgrade using Confirmed mode and fix the problem.
  - Continue the component upgrade as far as you can despite entering incorrect information.
  - Restart the component upgrade and choose Confirmed mode.
  - Skip any events that completed successfully during the initial component upgrade.
  - Accept and perform any events that were not successful (and restate any incorrect information).
  - Confirm that the component upgrade was successful as described in Part III, "Upgrading Components".
  - Perform all other tasks in sequence, as described in "Upgrade Task Overview" on page 1-6.
  - When all upgrade tasks are performed, validate the complete system upgrade as described in Part V, "Validating the Upgrade".

Support Deprecated

Table 2–4 describes the items for which support is not longer officially available.

Table 2–4 Deprecated in 10g (10.1.4.0.1)

Component	Comments
IDLink	Support was deprecated in release 6.1. If your earlier installation includes Oblix IDLink, you are notified while upgrading the Identity Server. To continue using Oblix IDLink, you must retain the earlier release.

Table 2–4 (Cont.) Deprecated in 10g (10.1.4.0.1)

Component	Comments
Publisher	Support was deprecated in release 6.0. Publisher cannot operate at the same time as release 6.1 or later. Oracle Access Manager 10*g* (10.1.4.0.1) provides reporting, auditing, and logging enhancements. You can create, view, and configure reports within the User, Group, and Organization Manager applications. For more information, see the *Oracle Application Server Release Notes*.
NetPoint Certificate Process Server (CPS)	Support was deprecated in release 7.0. If your earlier installation includes the CPS, following the upgrade you will have to request and install any new certificates through a third-party vendor.
NetPoint Associate Portal Services (APS)	Support was deprecated in release 6.5 when NetPoint SAML Services (now Oracle COREid Federation) became the preferred method to provide access privileges across multiple associated portals and DNS domains. APS remains deprecated.
NetPoint SAML Services	There is no migration path from NetPoint SAML Services to any Oracle Federation product available with 10*g* (10.1.4.0.1). NetPoint SAML Services was replaced with Oblix SHAREid.
Oblix SHAREid	Renamed to Oracle COREid Federation. This functionality is now accomplished with Oracle Identity Federation. There is no migration path. However, you can install Oracle Identity Federation after upgrading to Oracle Access Manager10*g* (10.1.4.0.1).
Oracle COREid Federation	This functionality is now accomplished with Oracle Identity Federation. There is no migration path. However, you can install Oracle Identity Federation after upgrading to Oracle Access Manager10*g* (10.1.4.0.1).
Oracle COREid Provisioning	Support for this feature is deprecated in 10*g* (10.1.4.0.1). There is no migration path.
MIIS Provisioning	Provisioning external applications from Oracle Access Manager by integrating with Microsoft Identity Integration Server (MIIS) is deprecated in 10*g* (10.1.4.0.1). This functionality is now accomplished with Oracle Identity Manager (Oracle Xellerate Identity Provisioning), and is no longer available in Oracle Access Manager. There is no migration path.
Microsoft .NET Passport	Support for this feature is deprecated in 10*g* (10.1.4.0.1).
Valicert Authentication plug-in	Support was deprecated in release 7.0.4 (also available as part of Oracle Application Server 10g Release 2 (10.1.2)). This is no longer distributed with the Access Server (including Authn_valicert authentication plug-in, authn_valicert.dll, and authn_valicert_d.dll).
Siemens DirX Directory	This directory is not supported in 10*g* (10.1.4.0.1). Although the installation screen may still display DirX as a possible option.
NetPoint Connector for BEA Ready Realm	Support was deprecated in release 7.0.4.2. However, the Security Provider for WebLogic SSPI is still supported. To upgrade an earlier Security Provider for WebLogic SSPI to the latest release, see"Upgrading Third-Party Integration Connectors" on page 11-1. To integrate a new Security Provider for WebLogic SSPI, see the *Oracle Access Manager Integration Guide*.

Upgrade Strategies When Support is Changed or Deprecated

This discussion provides strategies to help you proceed with a component upgrade when support for a directory server or Web server version has changed or been deprecated.

The strategies presented here focus on a single component upgrade in a specific situation:

- Upgrading When Third-Party Support Has Changed
- Upgrading When Third-Party Support Has Been Deprecated

Note: Before upgrading an Oracle Access Manager installation earlier than release 6.1.1, contact Oracle Support at `http://www.oracle.com/support/contact.html`

Upgrading When Third-Party Support Has Changed

When 10*g* (10.1.4.0.1) supports a different Web server or directory server release than those in your earlier installation, you must complete the upgrade a little differently to accommodate upgrading third-party components.

The following overview outlines the sequence you need to complete when you must upgrade an Oracle Access Manager component in addition to upgrading a Web server (or directory server) instance to meet 10*g* (10.1.4.0.1) requirements. This is provided to give you an idea of how to proceed in this situation and is not meant to provide all steps needed to accomplish the task. See your vendor documentation for information about third-party components and other chapters in this guide for details about Oracle Access Manager components and validation steps.

For example, when 10*g* (10.1.4.0.1) supports the same Web server or directory server versions as your earlier installed Oracle Access Manager release, you simply upgrade each component once and accept changes to third-party configuration files. However, during an upgrade, third-party configuration files are not updated in their entirety. Instead, only the delta is applied (the difference between changes for the old release and changes for 10*g* (10.1.4.0.1). For this reason, you may not simply install a new Web server instance and specify the path to it during an upgrade.

Note: The strategies outlined here presume that you have completed all appropriate preparation tasks, and that you are following steps provided elsewhere in this guide. Preparation, verification, and recovery steps are *not* repeated here. Steps to upgrade the specific Oracle Access Manager component are *not* repeated here.

Task overview: Upgrading Oracle Access Manager together with third-party product versions

1. Compare support requirements under the Certify tab at `https://metalink.oracle.com`
   - Log in as directed.
   - Click the Certify tab.
   - Click View Certifications by Product.
   - Select the Application Server option and click Submit.
   - Choose Oracle Application Server and click Submit.
2. **Directory Server Upgrade**: If this applies to your environment, perform the activities in the following list:
   - Use instructions in Chapter 5, "Preparing for Schema and Data Upgrades" to backup current directory instances and data (and to create and prepare master instances of the earlier Identity Server, WebPass, and Policy Manager against the existing directory server).

- Stop all Identity Server services and follow instructions in your vendor documentation to upgrade a third-party directory server to the new level supported by 10*g* (10.1.4.0.1).
- Perform and validate the schema and data on the upgraded directory instance upgrade using the master Identity Server and Policy Manager as described in:
  - Chapter 6, "Upgrading Identity System Schema and Data"
  - Chapter 7, "Upgrading Access System Schema and Data"

3. **Web Server Upgrades amid Oracle Access Manager Upgrades**: If your environment includes an earlier Web server version than is supported by10*g* (10.1.4.0.1), prepare components and perform upgrade activities as prescribed in this manual with the following differences:
   - **Oracle Access Manager Web Component Upgrades**: When you upgrade Web components, accept the automatic Web server configuration file update for the currently installed Web server.
   - **Web Server Upgrade**: Use your vendor documentation to backup an older third-party Web server then upgrade it to the new level supported by 10*g* (10.1.4.0.1).
   - Manually update the Web server configuration file for 10*g* (10.1.4.0.1) following the Web server upgrade. For more information, see the *Oracle Access Manager Installation Guide*.

   Note: You may not apply Oracle Access Manager-related Web server configuration changes to a new Web server instance.

4. Complete other activities as described in this manual, then validate the upgrade as described in Chapter 14, "Validating the Entire System Upgrade".

Upgrading When Third-Party Support Has Been Deprecated

In some cases, you may discover that Oracle Access Manager 10*g* (10.1.4.0.1) does not support an earlier Web Server or directory server release. For example, the release 6.1 Policy Manager supports Sun (formerly iPlanet) 4.x Web server. However, from Oracle Access Manager 6.5 onward this Web server release is not supported.

When 10*g* (10.1.4.0.1) does not support an earlier Web Server or directory release, you must complete the upgrade as outlined in:

- Upgrading with Manual Web Server Configuration When Support is Deprecated
- Upgrading Oracle Access Manager Incrementally When Third-Party Support is Deprecated

Note: Before upgrading an installation earlier than release 6.1.1, contact Oracle Support at
`http://www.oracle.com/support/contact.html`

Upgrading with Manual Web Server Configuration When Support is Deprecated

When 10*g* (10.1.4.0.1) support does not include your earlier release Web Server, you can use the strategy here to upgrade to 10*g* (10.1.4.0.1). For example, from Oracle Access Manager 6.5 onward the Sun (formerly iPlanet) 4.x Web server is not

supported. As a result, during the upgrade from Oracle Access Manager release 6.1.1 to release 6.5 the Web server configuration files are not automatically updated. Instead, you must install the Sun 6.x Web server and run EditObjConf and ManageObjConf manually to update the Web server configuration files for Oracle Access Manager release 6.5, 7.x, and 10*g* (10.1.4.0.1).

The following task overview is provided to give you an idea of how to proceed in this situation and is not meant to provide all steps needed to accomplish the task.

> **Note:** The strategies outlined here presume that you have completed appropriate preparation tasks, and that you are following steps provided elsewhere in this guide. Preparation, verification, and recovery steps are *not* repeated here. Steps to upgrade the specific Oracle Access Manager component are *not* repeated here. Release numbers in examples are provided for illustration only.

Task overview: Upgrading when Web server support was deprecated

1. Upgrade your earlier Oracle Access Manager installation to 10*g* (10.1.4.0.1), including all Web components (WebPass, Policy Manager, and WebGate).

   > **Note:** Web server configuration files are not automatically updated.

2. Create an instance of the Web server that is supported by 10*g* (10.1.4.0.1) using your vendor documentation as a guide.
3. Run the EditObjConf tool for WebPass, Policy Manager (formerly the Access Manager component), then WebGate, as needed.

   WebComponent_install_dir\identity|access\oblix\apps\common\bin\EditObjConf.exe

4. Run the ManageObjConf tool for WebPass, Policy Manager, then WebGate, as needed.

   WebComponent_install_dir\identity|access\oblix\apps\common\bin\ManageObjConf.exe

5. Perform the component validation step to ensure that it upgraded properly as described in Part III, "Upgrading Components".

Upgrading Oracle Access Manager Incrementally When Third-Party Support is Deprecated

The following method describes an incremental upgrade that you can use when the latest Oracle Access Manager release is *not* compatible with both the currently *installed* Web server (or directory server) release, and the currently *supported* release.

In this case, the goal is to use Confirmed mode to upgrade the Oracle Access Manager component incrementally to a release that supports both the earlier Web server (or directory server) release and a later interim Web server (or directory server) release. You repeat the process to incrementally upgrade the Oracle Access Manager component and the third-party component until both are in sync with 10*g* (10.1.4.0.1).

As you complete this task in confirmed mode, you accept appropriate processes while skipping those that take the Oracle Access Manager component too far. Then, you migrate your earlier Web server (or directory server) to the newer supported release.

This may involve a sequence of manual stpes to *true up* the configuration files for the new instance. You may need to repeat this sequence until you have upgraded both the third-party component to a release supported by 10*g* (10.1.4.0.1), and the Oracle Access Manager is upgraded to 10*g* (10.1.4.0.1).

In the following task overview, a WebPass upgrade is interspersed with a Web server upgrade. The strategies outlined here presume that you have completed appropriate preparation tasks, and that you are following steps provided elsewhere in this guide. Preparation, verification, and recovery steps are *not* repeated here. Steps to upgrade the specific Oracle Access Manager component are *not* repeated here. See also "Console Method" on page 2-6.

> **Note:** Release numbers in examples are provided for illustration only. Before upgrading an installation earlier than release 6.1.1, contact Oracle Support at
> `http://www.oracle.com/support/contact.html`

Task overview: Upgrading incrementally when Web server support is deprecated

1. On the machine hosting the earlier Oracle Access Manager Web component (WebPass, Policy Manager, or WebGate), start the upgrade using 10*g* (10.1.4.0.1) installers. For example:

   Start Upgrading: WebPass 5.2 on Sun ONE Web Server 4.1

2. In Confirmed mode, accept processes that upgrade the Oracle Access Manager Web component only to the next major release that supports the current Web server. For example, in this case you upgrade only to 6.0:

   From: WebPass 5.2

   To: WebPass 6.0

   > **Note:** Skip any processes that would upgrade this component to a release that does not support the current environment.

3. Accept the automatic Web server configuration file update, and finish the component upgrade for this incremental release.

4. Migrate the current Web server to the latest level supported by the interim WebPass release, using your vendor documentation as a guide. For example:

   From: Sun ONE Web Server 4.1 (supported by Oracle Access Manager 5.2)

   To: Sun ONE Web Server 6.0 (supported by Oracle Access Manager 6.0)

5. Restart the WebPass upgrade, use Confirmed mode to skip processes already completed and accept upgrade processes that upgrade WebPass to a later release that supports the upgraded Web server. For example:

   From: WebPass 6.0

   To: WebPass 6.1.1

6. Accept the automatic Web server configuration file update, and finish this incremental component upgrade.

7. Repeat steps in this list as needed until you reach and meet 10*g* (10.1.4.0.1) support requirements for the third-party component and upgrade the Oracle Access Manager component to 10*g* (10.1.4.0.1).
8. Validate the WebPass upgrade as described in "Finishing and Verifying the WebPass Upgrade" on page 9-11.

For more information, see Appendix D, "Upgrading Sun Web Server Version 4 to Version 6 on Windows 2000". If needed, see "Preparing a Directory Server when Its Release is Deprecated" on page 5-9.

3

About Automated Processes and Manual Tasks

This chapter introduces both the automated processes that are initiated when you start a component upgrade and manual tasks that you must perform. Topics include:

- Supported Components and Applications
- About the Automated In-Place Component Upgrade Process and Events
- Upgraded Items
- Preserved Items
- Items that You Must Manually Upgrade

Supported Components and Applications

Both earlier releases and 10*g* (10.1.4.0.1) support the following components:

- Identity Server (formerly known as the COREid Server), WebPass, Policy Manager (formerly known as the Access Manager component), Access Server, WebGate, and Software Developer Kit

  Note: The Simple Network Management Protocol (SNMP) has not changed and does not require an upgrade.

- Oracle Access Manager applications, which are integral to components, include the Identity System Console, User Manager, Group Manager, Organization Manager, the Selector, the Access System Console, Policy Manager (formerly known as the Access Manager), and other applications
- Integration components such as the Security Provider for WebLogic SSPI and Connector for WebSphere, as well as single sign-on (SSO), provisioning, portal and application server integrations are supported. Complete implementation details are described in the *Oracle Access Manager Integration Guide*.

For information about system requirements and changes, see "Platform Support" on page 4-1.

About the Automated In-Place Component Upgrade Process and Events

As mentioned earlier, you upgrade each component in place. This means that the newest product release is installed over an earlier product release in the same location.

This discussion introduces the program-driven processes that occur during component upgrades.

You initiate a component upgrade using the corresponding Oracle Access Manager 10*g* (10.1.4.0.1) installer. During each component upgrade, the program controls the sequence of events and messages automatically. The component upgrade process requires very little input from you.

After you start an upgrade and specify the same (target) installation directory where the earlier (source) component resides, you are asked if you want to upgrade the earlier version of the component.

Note: If the target installation directory for a 10*g* (10.1.4.0.1) component does not match the installed directory of the same earlier component, the component is installed (not upgraded).

When you accept the upgrade option, the source directory is renamed with the addition of a time stamp (*yearmonthday_hourminutesecond*). This time-stamped source directory contains earlier original files that are sometimes accessed to compare content or extract customized information.

Sample Time-Stamped Directory:

IdentityServer_install_dir_20060422_141440\identity

After the source directory is renamed with a time stamp, the target directory is created and new files are extracted to the target. For example:

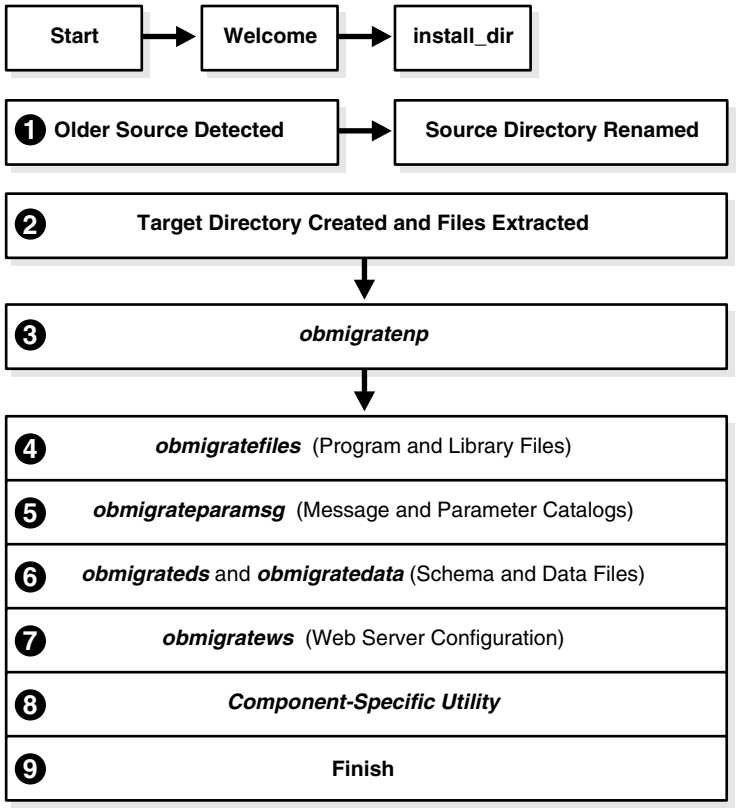
Sample Target Directory: *IdentityServer_install_dir*\identity

New folders are created in the target directory as needed. While the directory structure for Oracle Access Manager release 6.5, release 7.0, and 10*g* (10.1.4.0.1) are the same, they differ from earlier releases. For details, see Appendix A, "Oracle Access Manager Directory Structure Changes".

You may choose to have individual events within the upgrade process performed automatically or with your confirmation, as described earlier in "Upgrade Event Modes" on page 2-6.

Figure 3–1 and the process overview that follows it describe a typical component upgrade that is driven by each program (and the utilities that are called automatically during the process). At several points during the process, you are asked to provide specific input (such as a path name) or accept an automatic process, or simply acknowledge that you are ready to continue.

Figure 3–1 Automated Program-Driven Events During a Component Upgrade



During each component upgrade, additions and changes (from each major release to the next major release to release 10*g* (10.1.4.0.1)) are implemented As a result, the sequence of events and messages may repeat automatically until all changes between your starting release and 10*g* (10.1.4.0.1) are incorporated.

> **Note:** Process overviews, such as the one here, identify automated processes. After you initiate the process, you may be asked to accept or acknowledge certain events. Skipping or declining an event is sometimes an option.

Process overview: During an in-place component upgrade

1. After you launch the 10*g* (10.1.4.0.1) component installer, and an earlier source directory is detected in the same location, you are asked if you want to upgrade. When you accept the upgrade, the source directory is renamed with a time stamp.
2. The target directory is created and 10*g* (10.1.4.0.1) files are extracted into it. The English language is upgraded automatically. Other installed languages are upgraded when you include appropriate 10*g* (10.1.4.0.1) Language Packs for each installed language in the same directory as the 10*g* (10.1.4.0.1) component installer.

> **Note:** If you upgrade an existing multi-language implementation without 10*g* (10.1.4.0.1) Language Packs, you will lose multi-language functionality.

3. A utility (obmigratenp) is called by the component installer to determine the release you are upgrading *from* as well as the release you are upgrading *to*. obmigratenp internally detects which features need to be upgraded for this particular component and which other utilities to use for those upgrades. When your installation includes multiple languages, obmigratenp migrates message catalogs.
4. A utility (obmigratefiles) is called to upgrade earlier program and library files.
5. A utility (obmigrateparamsg) is called to upgrade earlier message and parameter catalog files.
6. **Schema and Data Upgrades Only**: Two utilities (obmigrateds and obmigratedata) are called automatically to initiate Oracle Access Manager schema and data upgrades for the master Identity Server (and master Policy Manager when your installation includes the Access System). During subsequent Identity Server (and Policy Manager) upgrades, the initial schema and data upgrade is detected and this portion of the process is skipped.

   Oracle recommends that you upgrade the Oracle Access Manager schema and data automatically, as described in Part II, "Upgrading the Schema and Data". Such upgrades use ldif files specific to your directory. Each ldif file includes only changes from one release of Oracle Access Manager to the next. As a result, the schema and data upgrade will repeat one time for each release from your starting release to 10*g* (10.1.4.0.1). For example, if you are upgrading from release 6.1.1, the schema and data upgrade occurs as follows:

   - From release 6.1.1. to release 6.5
   - From release 6.5 to release 7.0
   - From release 7.0 to 10*g* (10.1.4.0.1)
7. **Web Server Configuration Updates Only**: A utility (obmigratews) is called to perform a selective Web server configuration file and filter upgrade, to accommodate changes for newer releases of Policy Manager, WebPass, and WebGate.
8. A component-specific utility is selected and run to make changes to related registry entries for Windows, plug-ins, and other files. The component's configuration files are updated. For more information, see "Component-Specific Upgrades" on page B-16.

   See Also: "Upgraded Items" next, "Preserved Items" on page 3-6, and "Items that You Must Manually Upgrade" on page 3-8

During each component upgrade, one or more log files may be produced to inform you if any problem should arise. If a log file is created, a message during the upgrade process indicates the name and location of the file. In general, you can find upgrade log files in:

Log File Path:

Component_install_dir\identity|access\oblix\tools\migration_tools*toolname*.log

where *Component_install_dir* is the directory where the specific component is installed; identity|access represents the system to which the component belongs (Identity System or Access System, respectively); and *toolname* represents the name of the utility that produced the log.

Each log file contains information about a particular activity that occurs during the component upgrade. For example, a separate log file may be generated for file

upgrades, or message and parameter upgrades, or the Oracle Access Manager schema upgrade to name a few. For information about specific log files and their content, see Appendix B, "Upgrade Process and Utilities".

In addition, the log files here are created to inform you of any ldap specific errors:

- During Identity Server data migration, error_output_*fromversion*_to_*toversion*_ osd.ldif file is created in the *IdentityServer_install_ dir*\identity\oblix\tools\migration_tools\obmigratedata directory.
- During Policy Manager data migration, error_output_*fromversion*_to_*toversion*_ psc.ldif file is created in the *PolicyManager_install_ dir*\access\oblix\tools\migration_tools\obmigratedata directory

For details about using log files, see Appendix F, "Troubleshooting the Upgrade Process".

Upgraded Items

The following items are upgraded during component upgrades:

- Program and library files
- Message and parameter catalogs
- The Oracle Access Manager schema and configuration and policy data are upgraded only with the master Identity Server and Policy Manager installed for this purpose
- Web server files and filters are upgraded only for Web components (WebPass, Policy Manager, WebGate)
- Component-specific information, including component configuration files and system environment settings such as registry entries for Win32
- Product and component names are changed as described in "Product and Component Name Changes" on page -xxi
- Certain configuration files, including those for failover and the software developer kit (SDK) are upgraded as indicated in the following list:
  - **Identity Server:** sample_failover.xml
  - **Access Server:** failover files

    See Also: "Directory Server Failover" on page 3-6 and Chapter 4, "System Behavior and Backward Compatibility"

  - **SDK Configuration**: The upgrade is invoked automatically as the last step when upgrading the Identity Server (and integration components that rely on the Software Developer Kit libraries). Oracle recommends that you accept the automatic upgrade to preserve current configuration settings. Otherwise, you must reconfigure the SDK later using the configureAccessGate tool, as described in Chapter 11, "Upgrading Integration Components and an Independently Installed SDK".

    Note: Only supported components are upgraded. For more information, see "Support Deprecated" on page 2-8.

Preserved Items

Any names assigned by an administrator during product installation and configuration are retained during an upgrade (not changed). Therefore if you have named a service "COREid Identity Server" or "NetPoint Identity Server" these names will be the same in the upgraded environment.

Earlier authentication schemes and policy domains assigned by administrators are also retained during the upgrade. After the upgrade, these names are still available.

The items in the following list are preserved in the time-stamped directory and copied into the new target directory:

- certificate files (simple/cert mode)
- password.xml and .lst (.lst is converted to .xml for the Access System)
- configuration files (.oblix/config, .oblix/data)
- obnavigation.xml
- oblixpppcatalog.lst—not converted to .xml
- cert7.db

  Starting with release 7.0 and continuing with 10*g* (10.1.4.0.1), the default certificate store format and name has changed to cert8.db from cert7.db. After the upgrade, the old certificate store is used. 10*g* (10.1.4.0.1) (and release 7.x) work with both the cert7.db (upgraded environments) and cert8.db (new installations) certificate store. See "Certificate Store and Localized Certificates" on page 4-6.

For additional information, see the topics:

- Directory Server Failover
- Connection Pool Details
- Encryption Schemes and the Shared Secret

> **See Also:** Chapter 4, "System Behavior and Backward Compatibility"

Directory Server Failover

Starting with the Access System release 6.5, directory profiles are created during Policy Manager setup and are used by the Access System to access user directory data. These profiles replace the UserDB.lst and GroupDB.lst files and the UserDBFailover.lst and GroupDBFailover.lst configuration files that were used in earlier releases of the Access System.

During the Access System upgrade from release 6.1.1, new directory profiles are created based on the UserDB.lst and GroupDB.lst files and the UserDBFailover.lst and GroupDBFailover.lst configuration files.

Your earlier implementation may also include failover between an Identity Server and the directory server. The Identity Server failover configuration has resided in the directory server profiles since release 5.2. As a result, during the Identity Server upgrade there is no migration of parameters from failover configuration files to directory profiles. Although the schema itself has changed, migration of these changes is performed automatically during the upgrade.

Impact of the Upgrade on Directory Server Failover

Starting with release 6.5, the Access System began partially using directory profiles and database instances for accessing user data. Directory profiles replace the UserDB.lst, GroupDB.lst, UserDBFailover.lst, and GroupDBFailover.lst configuration files that were used in earlier Access System releases. During the incremental upgrade to release 6.5, directory profiles for the Access Server are automatically created and replace certain earlier configuration files where:

- Primary directory server information was stored in: *AccessServer_install_dir*/access/oblix/config/UserDB.lst and GroupDB.lst
- Information for all the failover and load-balancing directory servers (primary and secondary) was stored in: *AccessServer_install_dir*/access/oblix/config/UserDBFailover.lst and GroupDBFailover.lst

When creating the Directory Server Profile for the Policy Manager, directory server credentials are read from *PolicyManager_install_dir*/access/oblix/config/userDB.lst.

> **Note:** If the configuration tree is in the user directory server and under the user node, then the configuration directory profile is not created. Otherwise, a configuration directory profile is created using directory server information from *PolicyManager_install_dir*/oblix/config/ldap/configdb.lst and marked for use only by the Policy Manager.

Profiles are *not* created for Policy Manager failover servers. In the case of release 6.1, if the policy tree was on a separate directory server a profile for policy data existed.

After upgrading the Identity System and Access System, it is a good idea to validate your failover and load balancing configurations and to test that these are still operating as expected. For details, see discussions in:

- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"

See also "Connection Pool Details" next.

Connection Pool Details

As described in the *Oracle Access Manager Deployment Guide*, the number of connections opened to the directory is specified by the `Initial Connections` parameter in the Database Instance Profile. More connections are opened, as needed, until they equal the number specified by `Maximum Connections` parameter in the database instance profile. Connections remain open until the Identity or Access Server shuts down or the directory server stops responding. Those connections are then pooled and used by the Identity and Access Server.

> **Note:** Starting with Oracle Access Manager release 7.0, connection pooling was consolidated to support failover across the entire system. The directory connection pool does not depend on directory type.

There may be some impact when upgrading the Access System, depending on the earlier configuration of Oracle Access Manager to each directory server used. For details, see "Confirming Access System Failover and Load Balancing" on page 13-3.

See also the previous discussion on "Directory Server Failover".

Impact of the Upgrade on Connection Pools

There may be some impact when upgrading, depending on the earlier configuration of Oracle Access Manager to each directory server used. For instance:

- **Identity System Connection Pools**: There is no impact. `Initial Connections` and `Maximum Connections` specified in the database instance profile are retained and will operate as they did previously.
- **Access System Connection Pools Before release 6.5**: Values for the `Initial Connections` and `Maximum Connections` in the UserDB.lst and UserDBFailover.lst may **not** be retained. After upgrading Access System components, it is a good idea to verify the values in the database instance profile of the newly created directory server profile.
- **On NDS**: For concurrent authentication requests on NDS directory servers, Oracle recommends that you increase the connection pool size to something higher than the default (1) for the user directory profile using the System Console.

Encryption Schemes and the Shared Secret

The shared secret encryption algorithm is an Oracle Access Manager-wide setting. It affects all encrypted cookies. For example, the ObSSOCookie cookie is encrypted using a configurable encryption scheme known as a *shared secret*. During the upgrade, the earlier encryption scheme is retained.

In release 7.0.4 and earlier, WebGates/AccessGates handled encryption and decryption using the shared secret value. Starting in 10*g* (10.1.4.0.1), however, the Access Server handles encryption/decryption. As a result, the shared secret is no longer needed on WebGate.

Oracle recommends that you upgrade earlier WebGates. However, earlier WebGates may coexist with 10*g* (10.1.4.0.1) Access Servers when specific conditions are met. For more information on encryption schemes, the shared secret, Access Servers, and WebGates in Chapter 4, "System Behavior and Backward Compatibility".

Items that You Must Manually Upgrade

The following discussions outline items that require you to perform manual upgrade tasks to ensure compatibility and proper operation with release 10*g* (10.1.4.0.1):

- Auditing and Access Reporting
- C++ Programs
- Customized Styles
- Plug-ins

In addition to the topics in the preceding list, see details in:

- Preparing Earlier Customizations in Chapter 8, "Preparing Components for the Upgrade"
- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"

Auditing and Access Reporting

Oracle Access Manager 10*g* (10.1.4.0.1) supports the Unicode standard. To support all the languages available with Oracle Access Manager 10*g* (10.1.4.0.1), the definitions of auditing and reporting tables have changed. Simply upgrading or altering existing database instances and tables is not supported and could result in permanent truncation and loss of existing data.

After upgrading the Identity System (and Access System), you need to create a new database instance to operate with 10*g* (10.1.4.0.1). To upload the new Audit table schema to support the auditing of 10*g* (10.1.4.0.1) UTF-8 data and writing this data to the new SQL Server instance, you must create a new oblix_audit_events table. This schema upgrade includes datatype changes within Audit table columns. Next you need to create tables for the reporting application (oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users) in 10*g* (10.1.4.0.1).

To query or generate any report that requires data from both the old and new database, you need to import data from the original instance into the new instance *before* you start auditing with 10*g* (10.1.4.0.1). This is an optional step that may or may not be needed in your environment.

> **Note:** Retain the earlier database to preserve the original data. Importing earlier data may result in some data loss through truncation. However, if you do not import old data before you start auditing, you cannot generate any report that requires the data from both the old and new database.

The steps you need to perform, even when you have an English only environment, depend on the type of database you are using. For details, see:

- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"
- See also "Auditing and Access Reporting" on page 4-5

C++ Programs

You will need to recompile C++ programs created with the Software Developer Kit and Oracle Access Manager APIs following that upgrade, for the reasons stated in "Plug-ins" on page 3-10.

> **Note:** Oracle recommends that you begin migrating your earlier customizations in a test environment well before you begin upgrading components. This will help reduce system downtime when upgrading your production environment and redeploying customizations.

For more information, see Chapter 4, "System Behavior and Backward Compatibility", Chapter 12, "Upgrading Your Identity System Customizations", and Chapter 13, "Upgrading Your Access System Customizations".

Challenge and Response Attributes Must Appear on a Panel

In earlier releases, the challenge phrase and response attributes were allowed on different panels of Profile pages. In 10*g* (10.1.4.0.1), however, both the challenge phrase and response attributes must be on the same panel. In 10*g* (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.

For details about combining challenge and response attributes on a single panel, see "Combining Challenge and Response Attributes on a Panel" on page 12-8.

Customized Styles

Default product stylesheets are periodically updated by Oracle to instantiate improvements. For example, to support multiple languages, the location of java scripts, stylesheets, and images changed starting with Oracle Access Manager release 6.5. The directory structure introduced with release 6.5 continues with 10*g* (10.1.4.0.1).

Upgraded functionality depends, in part, on stylesheet files in the new release \style0 and \shared directories.

> **Note:** If files in your earlier Oracle Access Manager \style0 directory were customized, you must manually edit the newer version files in \style0 and \shared directories after the upgrade.

It is important to understand the new file hierarchy and stylesheet structure before you can successfully migrate custom images and stylesheets to 10*g* (10.1.4.0.1). If you simply copy the old stylesheets, images, JavaScript files, and related items to the new release, Oracle Access Manager may experience problems.

The following files must be processed manually after the upgrade:

- XSL stylesheets
- Images (.gifs for Oracle Access Manager)
- JavaScript Files

> **Note:** Oracle recommends that you begin migrating your earlier customizations in a test environment well before you begin upgrading components. This will help reduce system downtime when upgrading your production environment and redeploying customizations.

For details, see Chapter 12, "Upgrading Your Identity System Customizations". For details about the Oracle Access Manager directory structure, see Appendix A, "Oracle Access Manager Directory Structure Changes".

Plug-ins

Plug-in behavior has changed in recent releases. Following are important details that you need to be aware of with regard to custom plug-ins.

All earlier plug-ins send and receive data using Latin-1 encoding. Starting with 10*g* (10.1.4.0.1), Oracle Access Manager components and plug-ins send and receive data in UTF-8 format. Identity and Access Servers that are upgraded to 10*g* (10.1.4.0.1) provide backward compatibility with earlier plug-ins using Latin-1 encoding automatically. However, to send and receive internationalized data, earlier plug-ins should be redesigned to communicate using UTF-8 encoding. This includes Identity Event plug-ins and custom authentication and authorization plug-ins. For more information about globalization, see Chapter 4, "System Behavior and Backward Compatibility".

Starting with Oracle Access Manager release 7.0, components on Solaris and Linux are compiled using the GCC v3.3.2 C++ compiler to address multi-threading issues encountered with earlier compiler releases. This means that after the upgrade, you must recompile custom plug-ins from release 5.x or 6.x using GCC v3.3.2 C++ compiler. This includes Identity Event plug-ins and custom authentication and authorization plug-ins. You must use the GCC v3.3.2 compiler, regardless of the one that may be provided by your Operating System.

Note: Release 7.0 plug-ins as well as earlier plug-ins implemented as executables or those using a scripting language (such as perl) do not require recompiling after the upgrade. However, to send and receive internationalized data, earlier plug-ins should be redesigned to communicate using UTF-8 encoding.

Identity Event API Plug-Ins: Some plug-ins are copied during the upgrade; however, Identity Event API plug-ins are not. After the upgrade you must move earlier Identity Event plug-ins. These plug-ins may also need to be re-compiled or re-designed. For more information, see "Migrating Custom Identity Event Plug-Ins" on page 12-10.

Authentication and Authorization Plug-Ins: After the Access System upgrade, see "Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins" on page 13-5. Also, the *Oracle Access Manager Access Administration Guide* provides more information about:

- Adding customized plug-ins and parameters to an authentication scheme to be used for any of the scheme's steps
- Installing a custom authorization plug-in on application servers that you want to protect and creating custom schemes that include custom plug-ins to perform different (or additional) tasks from those of the default scheme

---

Note: Oracle recommends that you begin migrating your earlier customizations in a test environment well before you begin upgrading components. This will help reduce system downtime when upgrading your production environment and redeploying customizations. For more information, see "Customization Upgrade Planning" on page 1-13.

4

System Behavior and Backward Compatibility

This chapter provides a centralized summary of expected system behaviors and changes between Oracle Access Manager 10*g* (10.1.4.0.1) and earlier releases. Behaviors that have not changed are, for the most part, not included in this chapter. Information is organized into the categories in the following list:

- Platform Support
- About Upgrading and Backward Compatibility
- General Behavior Changes
- Identity System Behavior Changes
- Access System Behavior Changes

> **Note:** For a quick reference table of Oracle Access Manager 10*g* (10.1.4.0.1) behaviors (as well as an overview of new functions and features), see the *Oracle Access Manager Introduction*. See also "Product and Component Name Changes" on page -xxi.

Platform Support

There are no significant changes in platform support between releases 7.0.4 (also available as part of Oracle Application Server 10g Release 2 (10.1.2)) and 10*g* (10.1.4.0.1). However, there may significant differences between releases prior to 7.0.4 and 10*g* (10.1.4.0.1).

To ensure that you always have the most up to date information, support details are not presented in manuals. For the latest support information, see details under the Certify tab on:

```
https://metalink.oracle.com
```

To use MetaLink

1. Navigate to `https://metalink.oracle.com`.
2. Log in to MetaLink as directed.
3. Click the Certify tab.
4. Click View Certifications by Product.
5. Select the Application Server option and click Submit.

6. Choose Oracle Application Server and click Submit.

For a quick reference table of components and third-party products that are no longer supported, see "Support Deprecated" on page 2-8.

About Upgrading and Backward Compatibility

Backward compatibility with earlier plug-ins and the like is enabled automatically when you upgrade components. However, not all components are backward compatible with earlier components. Table 4–1 provides an overview with pointers to additional information.

Table 4–1 Backward Compatibility for Oracle Access Manager Components

Component	Backward Compatibility Enabled Automatically	For More Information See
Identity Servers (formerly known as the NetPoint or COREid Server)	When you upgrade Identity Servers, backward compatibility with earlier custom plug-ins is enabled automatically. If you add a 10*g* (10.1.4.0.1) Identity Server to an upgraded environment, you must set a flag manually to enable backward compatibility with earlier custom plug-ins. Upgraded Identity Servers are not backward compatible with earlier WebPass instances.	Identity Server Backward Compatability on page 4-18
WebPass	After upgrading all earlier Identity Servers, you must upgrade all earlier WebPass instances. Earlier WebPass instances are not compatible with 10*g* (10.1.4.0.1) Identity Servers (or Policy Managers). You may install 10*g* (10.1.4.0.1) WebPass instances in your upgraded environment. However, 10*g* (10.1.4.0.1) WebPass instances are not compatible with earlier Identity Servers (or Policy Managers).	"Web Components and Backward Compatibility" on page 4-16
Policy Managers (formerly known as the Access Manager component)	After upgrading the schema and data, and all Identity System components, you must upgrade all earlier Policy Managers.	Policy Manager on page 4-28
Access Servers	When you upgrade earlier Access Servers, backward compatibility with earlier custom plug-ins and earlier WebGates is enabled automatically. However, if you add a 10*g* (10.1.4.0.1) Access Server to an upgraded environment, you must set a flag to enable backward compatibility. Earlier Access Servers are not compatible with 10*g* (10.1.4.0.1) WebGates.	Access Server Backward Compatibility on page 4-24
WebGates	Release 6.1.1, 6.5, and 7.x WebGates may coexist with upgraded 10*g* (10.1.4.0.1) Access Servers. You may install 10*g* (10.1.4.0.1) WebGates in your upgraded environment. However, 10*g* (10.1.4.0.1) WebGates are not compatible with earlier Access Servers. If you add a 10*g* (10.1.4.0.1) Access Server to the upgraded environment, you must set a flag to enable backward compatibility with earlier WebGates.	WebGates on page 4-29 Access Server Backward Compatibility on page 4-24

Oracle Access Manager 10*g* (10.1.4.0.1) retains customizations made in your earlier installation as described in "Preserved Items" on page 3-6. However, in certain cases, you need to perform manual tasks to upgrade or integrate customized items from your earlier environment into the upgraded environment, as outlined in Table 4–2. For more information, see "Customization Upgrade Planning" on page 1-13.

Table 4–2 Manual Tasks You Must Perform to Upgrade Customizations

Manual Tasks for Customizations	Details
Upgrading Auditing and Access Reporting for the Identity System	on page 12-2
Combining Challenge and Response Attributes on a Panel	on page 12-8
Confirming Identity System Failover and Load Balancing	on page 12-9
Migrating Custom Identity Event Plug-Ins	on page 12-10
Ensuring Compatibility with Earlier Portal Inserts	on page 12-11
Incorporating Customizations from Release 6.5 and 7.x	on page 12-12
Incorporating Customizations from Releases Earlier than 6.5	on page 12-13
Validating Identity System Customization Upgrades	on page 12-23
Upgrading Auditing and Reporting for the Access Server	on page 13-2
Confirming Access System Failover and Load Balancing	on page 13-3
Upgrading Forms-based Authentication	on page 13-4
Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins	on page 13-5
Associating Release 6.1.1 Authorization Rules with Access Policies	on page 13-5
Assuring Proper Authorization Failure Re-directs After Upgrading from 6.1.1	on page 13-6
Updating the ObAMMasterAuditRule_getEscapeCharacter in Custom C Code	on page 13-7
Validating Access System Customization Upgrades	on page 13-7

In addition to the preceding tables, this chapter provides a consolidated and centralized summary of expected behaviors in Oracle Access Manager 10*g* (10.1.4.0.1) so that you do not need to look through all the manuals to locate this information.

General Behavior Changes

This discussion provides information about previous behaviors that apply equally to the Identity and Access Systems. The focus is on changes to previous behaviors and what to expect after upgrading to 10*g* (10.1.4.0.1). Topics include:

- Acquiring and Using Multiple Languages
- Auditing and Access Reporting
- Automatic Schema Update Support for ADAM
- C++ Programs
- Cache Flush
- Certificate Store and Localized Certificates
- Compilers for Plug-ins
- Configuration Files
- Connection Pool Details
- Console-based Command-line Interfaces
- Customized Styles, Images, and JavaScript
- Database Input and Output
- Date and Time Formats

- Directory Profiles and Database Instance Profiles
- Directory Server Connection Details
- Directory Server Failover
- Directory Server Interface
- Directory Structure
- Domain Names, URIs, and URLs
- Encryption Schemes
- Failover and Failback
- File and Path Names
- Graphical User Interface
- HTML Pages
- Message and Parameter Files
- Names Assigned by Administrators and Product Names
- Namespaces for Policy Data and User Data Stored Separately
- Reconfiguring the Logging Framework without a Restart
- Support Changes
- Transport Security for the Directory Server
- XML Catalogs and XSL Stylesheet Encoding
- Web Server Configuration Files

Acquiring and Using Multiple Languages

Early product releases provided messages for end users and administrators in only the English language. Starting with release 6.5, support for translatable messages was provided through Language Packs for certain Latin-1 languages (French and German). Oracle Access Manager 10*g* (10.1.4.0.1) provides support for nearly a dozen Administrator languages and over two dozen end-user languages, as described in the *Oracle Access Manager Introduction*.

After installing Oracle-provided Language Packs, you must enable all languages to be used, then configure Oracle Access Manager to use the installed languages by entering display names for attributes, tabs, and panels. See the *Oracle Access Manager Installation Guide* and *Oracle Access Manager Identity and Common Administration Guide* for details about installing and enabling multiple-languages after the upgrade.

When upgrading to (or installing) 10*g* (10.1.4.0.1), you choose the language (locale) to be used as the default for Administrative tasks. Administrative information for the Identity System Console, Access System Console, and Policy Manager can be displayed in only installed Administrator languages. In earlier releases, a drop-down list of languages appeared in the top-right corner of the System Console. However, this is not available in 10*g* (10.1.4.0.1). The only way to select the language is by changing the browser setting on the user's or administrator's machine. If administrative pages are requested in any user language (based on the language selected for the browser), the language that was selected as the default Administrator language during product installation (or upgrades) is used to display administrative pages.

Messages in Oracle Access Manager stylesheets depend upon a language. Beginning with release 6.5 multiple-language capability, messages have been brought out of the

stylesheets and defined separately as variables in msgctlg.xsl (and msgctlg.js for JavaScript files). In addition, each stylesheet has a corresponding language-specific thin wrapper stored in *IdentityServer_install_dir*\identity\oblix\lang*langTag*\style0. Each wrapper in \style0 includes the main language-neutral stylesheet stored in *IdentityServer_install_dir*\identity\oblix\lang\shared. The purpose of this new thin wrapper is to segregate the main functionality of the stylesheet template, which is language independent, from language-specific messages in the stylesheets. For more information, see the *Oracle Access Manager Customization Guide*.

For more information, see "Console-based Command-line Interfaces" on page 4-7.

Auditing and Access Reporting

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

Oracle Access Manager 10*g* (10.1.4.0.1) supports the Unicode standard. To support all the languages available with Oracle Access Manager 10*g* (10.1.4.0.1), the definitions of auditing and reporting tables have changed. Simply upgrading or altering existing database instances and tables is not supported and could result in permanent truncation and loss of existing data. For more information, see "Auditing and Access Reporting" on page 3-9.

For the steps you need to take to ensure a properly working auditing and reporting environment after upgrading Oracle Access Manager components to 10*g* (10.1.4.0.1), see auditing and access reporting topics in:

- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"

Also, when configuring Audit Policies in the Identity System Console, you can specify a list of profile attributes for every audit record. 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.

Automatic Schema Update Support for ADAM

This has been removed due to an ldifde.exe tool licensing issue. For ADAM, the schema must be updated manually. For details, see the *Oracle Access Manager Installation Guide*.

C++ Programs

When upgrading from releases earlier than 7.0, you may need to recompile C++ programs created with the Access Manager SDK and Oracle Access Manager APIs following the upgrade. For more information, see:

- Identity System Event Plug-ins
- Access Manager SDK, Access Manager API, and Custom AccessGates

- Custom Authentication and Authorization Plug-ins and Interfaces

Cache Flush

A 10*g* (10.1.4.0.1) Identity Server cannot flush the cache of an earlier Access Server. To eliminate any problems, be sure to upgrade your Access Servers to 10*g* (10.1.4.0.1).

For more information, see "Access Server Backward Compatibility" on page 4-24.

Certificate Store and Localized Certificates

Communication between a directory server and Oracle Access Manager Servers, and the Policy Manager may be either open (no security) or use the Secure Sockets Layer (SSL). SSL-enabled requires a signer's certificate (root certification Authority (CA) certificate) in Base64 format from a third-party Certificate Authority.

Three transport security modes are provided for communication between Web clients (WebPass and Identity Server and between Policy Manager and WebPass and between Access Server and WebGate. These security modes are Open, Simple (Oracle-provided), and Cert (third-party CA).

In both Simple and Cert mode, Oracle Access Manager components use X.509 digital certificates only. This includes Cert Authentication between WebGates and the Access Server where the standard cert-decode plug-in decodes the certificate and passes certificate information to the standard credential_mapping authentication plug-in.

Both Oracle and third-party vendors provide localized certificates for LDAP SSL communication between components and the directory server and for Oracle Access Manager components installed in Cert mode. With localization and UTF-8 support in 10*g* (10.1.4.0.1), you can request and add localized certificates containing non-ASCII text in all fields except Email and Country (according to x509 standards). After receiving a localized certificate, you must install it using one of the Oracle Access Manager command-line tools as described in the *Oracle Access Manager Identity and Common Administration Guide*. If the server is running a non-English operating system, you *may* want to set the Oracle National Language Support NLS_LANG or COREID_NLS_LANG environment variables (or both) to override the automatic server locale detection as described in the *Oracle Access Manager Installation Guide*. Setting these variables is optional because Oracle Access Manager automatically detects and uses the server locale.

Starting with Oracle Access Manager 7.0 (also available as as part of Oracle Application Server 10g Release 2 (10.1.2)), the default certificate store format and name has changed from cert7.db to cert8.db for LDAP SSL certificates. When you upgrade to 10*g* (10.1.4.0.1), the old certificate store (cert7.db) is used. Oracle Access Manager 10*g* (10.1.4.0.1) works with both the cert7.db (upgraded environments) and cert8.db (new installations) certificate stores. You are not required to manually generate a new certificate store after upgrading. However, this will happen transparently whenever you add, modify, or delete certificates using configureAAAServer, setup_ois, or setup_accessmanager utilities that automatically modify the certificate store format and name to cert8.db.

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

Compilers for Plug-ins

Starting with Oracle Access Manager release 7.0, components on Solaris and Linux are compiled using the GCC v3.3.2 C++ compiler to address multi-threading issues

encountered with earlier compiler releases. This means that after the upgrade, you must recompile custom plug-ins from release 5.x or 6.x using GCC v3.3.2 C++ compiler. This includes Identity Event plug-ins and custom authentication and authorization plug-ins.

For more information, see:

- Migrating Custom Identity Event Plug-Ins
- Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins

WARNING: You must use the GCC v3.3.2 compiler, regardless of the compiler that may be provided with the Operating System.

Configuration Files

Previous versions of Oracle Access Manager managed certain information (including but not limited to directory connection information and WebGate parameters) solely through XML and LST configuration files. In 10*g* (10.1.4.0.1), Oracle Access Manager provides the ability to manage this information through the graphical user interface (GUI). See also "Directory Server Connection Details" on page 4-10 and "WebGates" on page 4-29.

Connection Pool Details

Starting with Oracle Access Manager release 7.0, connection pooling was consolidated to support failover across the entire system. The directory connection pool does not depend on directory type.

There may be some impact when upgrading, depending on the earlier configuration of Oracle Access Manager to each directory server used:

- **Identity System Connection Pools**: There is no impact. Initial Connections and Maximum Connections specified in the database instance profile are retained and will operate as they did previously.
- **Access System Connection Pools Before Release 6.5:** Values for `Initial Connections` and `Maximum Connections` in the UserDB.lst and UserDBFailover.lst may not be retained. After upgrading Access System Components, it is a good idea to verify the values in the database instance profile of the newly created directory server profile.
- **On NDS:** For concurrent authentication requests on NDS directory servers, Oracle recommends that you increase the connection pool size to something higher than the default (1) for the user directory profile using the System Console.

See also "Directory Server Failover" on page 4-11.

Console-based Command-line Interfaces

Oracle Access Manager provides console-based command-line tools that can be used by administrators to configure Access and Identity components. 10*g* (10.1.4.0.1) command-line tools automatically detect the server locale and use it for processing. You may optionally set either the COREID_NLS_LANG or NLS_LANG environment variables (or both), which toggle auto-detection off and take precedence over the server locale.

To ensure correct behavior of command-line interfaces with a non-English operating system, the Master Administrator must complete several tasks as described in the *Oracle Access Manager Installation Guide*.

Customized Styles, Images, and JavaScript

Default product stylesheets are periodically updated by Oracle to instantiate improvements. Upgraded functionality depends, in part, on stylesheet files in the latest \style0 and \shared directories.

Customized .XSL style files, images, and JavaScript files are not migrated during the upgrade. If your earlier Oracle Access Manager installation includes customized images, JavaScript files, and stylesheets, you need to complete manual processing to use these with 10*g* (10.1.4.0.1). If you use a style other than the Oracle Access Manager default Classic Style, you need to manually include those changes in 10*g* (10.1.4.0.1) stylesheets, images, and JavaScript files.

Starting in Oracle Access Manager 6.5, you need to reference images using the two variables ($gifPathName and jsPathName) to make your customization language and style independent.

Note: Any style directories created in the earlier installation are **saved**, not migrated, and are stored in the renamed (backup) source directory during the upgrade. After upgrading the Identity System, you must complete manual processing to use customized styles with 10*g* (10.1.4.0.1). See "Incorporating Customizations from Releases Earlier than 6.5" on page 12-13.

To support multiple languages, the location of java scripts, stylesheets, and images changed starting with Oracle Access Manager release 6.5. The directory structure introduced with release 6.5 continues with 10*g* (10.1.4.0.1).

Database Input and Output

Earlier releases used the Latin-1 character set. With 10*g* (10.1.4.0.1), Oracle Access Manager supports the Unicode character set and internationalized characters (Chinese, Japanese, Arabic, and the like).

In new installations, Oracle recommends that you choose a Unicode character set for your database. If you upgrade an earlier installation to 10*g* (10.1.4.0.1), be sure to change your database character set to Unicode.

In earlier releases with the Latin-1 character set, the varchar type for columns of audit and reporting related tables was sufficient. 10*g* (10.1.4.0.1), the audit record may contain data with non Latin-1 characters. For more information, see "Auditing and Access Reporting" on page 4-5.

Date and Time Formats

Formats may differ between the Identity System and Access System, as follows:

Identity System: In the 10*g* (10.1.4.0.1) Identity System, the date format remains the same as in the last release and is not internationalized (on the Diagnostics page and Ticket Information page for example). However, month names taken from Identity System message catalogs will be displayed in the locale specified by the browser.

As in earlier releases, date order formats (MM/DD/YYYY versus /MM/YYYY and the like) can be configured by modifying object class attributes in the Identity System Console as described in the *Oracle Access Manager Identity and Common Administration Guide*. On the Ticket Information page, the date is displayed in the format specified in the `obDateType` parameter in the globalparams.xml file. Weekday names do not appear anywhere within the Identity System.

Access System: In the 10*g* (10.1.4.0.1) Access System, month names, the date-order format (MM/DD/YYYY versus DD/MM/YYYY), and weekday names are displayed according to the locale specified for the browser. In the Access System, month and weekday names are not taken from message catalog files. The following information may vary from one locale to another:

Access System Date Format: In the Access System only, the date format is internationalized and will appear in the locale specified for the browser. In India, for example, the date format is typically expressed as DD/MM/YYYY. In the United States the date format is typically expressed as MM/DD/YYYY.

Access System Month Names: Earlier releases presented the names of months from language-specific message catalogs on the server. However, this meant that the user would see the month name in the server's locale. In the 10*g* (10.1.4.0.1) Access System, the name of the month will reflect the user's browser locale.

> **Note:** Month names, and weekday names, (both full and abbreviated) have been removed from the globalmsg.xml file. Time zone locations have been removed from the oblixadminmsg.xml and policyservcenmsg.xml. These files are located in *AccessServer_install_dir*\access\oblix\lang\en-us.

Access System Weekday Names: In earlier releases, the names of weekdays (like the names of the month) were taken from language-specific message catalogs in the server's locale. In the Access System 10*g* (10.1.4.0.1), the name of the day will reflect the locale specified by the user's browser.

Access System Time Zone List: In earlier releases, the names of the location/city appeared with the Greenwich Mean Time (GMT) offset. However, the location/offset pair was not static because of the daylight savings time rule.

In 10*g* (10.1.4.0.1), the Time zone list shows only the offset expressed as Universal Time Coordinated (UTC) plus or minus from 00:00 to 12:00 hours. For example, UTC-00:00 or UTC+01:00 or UTC-03:30, and so on.

> **Note:** Universal Time Coordinated (UTC) is also known as Coordinated Universal Time and sometimes as Universal Coordinated Time. All are abbreviated as UTC and refer to the standard time common to every place in the world (formerly and still widely referred to as Greenwich Mean Time (GMT) or World Time. UTC reflects the mean solar time along the Earth's prime meridian. The Time format remains the same as it was in the last release (7.0, also available as part of Oracle Application Server 10g Release 2 (10.1.2).

You will see examples of these behaviors on the Access Server Diagnostics page; the Timing Conditions page under the Authorization Rules in access policies created in the Policy Manager; on the Manage Reports page under the System Management tab

in the Access System Console; and the Manage Sync Records page under the System Management tab of the Access System Console.

Default Product Pages

With Oracle Access Manager 10*g* (10.1.4.0.1), there can be only one static HTML page at the address /identity/oblix/index.html and one static HTML page at the address /access/oblix/index.html.

These static product pages always use the default Administrator language selected during Identity Server and Access Server installation at this location. Starting with release 6.5, the product supported multiple Latin-1 languages (French, German). The default product page behavior remains the same as in previous releases.

See also "HTML Pages" on page 4-13.

Directory Profiles and Database Instance Profiles

Starting with release 5.2, the Identity System included directory profiles and database instances. Starting with release 6.5, the Access System began partially using directory profiles and database instances for accessing user data. Directory profiles replace the UserDB.lst, GroupDB.lst, UserDBFailover.lst, and GroupDBFailover.lst configuration files that were used in earlier Access System releases.

A directory profile (also known as 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.

Each directory profile can contain multiple primary/secondary "database instances". Each database instance profile represents connection information to and for a single directory server, including connection pool information.

A directory profile is created automatically each time you install an Identity Server, Policy Manager, or Access Server and specify new directory server connection information. You can create additional directory server profiles for load balancing and failover.

When you upgrade an earlier Policy Manager or Access Server, a message appears during the interval to release 6.5 informing you that a new directory profile was created. The message "DB Profiles created" refers to the directory server profile. During the creation of new Access System directory profiles, connection pool values from earlier configuration files may not be retained. After the upgrade, you need to verify these values in the Database Instance profile of the newly created directory profile. See also "Connection Pool Details" on page 4-7.

Directory Server Connection Details

Previous versions of Oracle Access Manager managed directory connection information solely through XML configuration files. Recently, Oracle Access Manager provided the ability to manage this information through the interface using the Directory Profile page in the Identity System Console and the Access System Console. However, some configuration and policy data are still managed through the XML files. See also "Directory Profiles and Database Instance Profiles" on page 4-10.

Directory Server Failover

The Identity Server failover configuration has resided in the directory server profile in the System Console since release 5.2. Starting with release 6.5:

- A directory server profile is created for the master directory server as well as any failover directory servers.
- Directory server information from certain configuration files is used to create one Database Instance Profile each for all configured primary (and secondary) directory servers.

  For example, during the incremental upgrade to release 6.5, directory profiles for the Access Server are automatically created and replace certain earlier configuration files where:

  – Primary directory server information was stored in:

    AccessServer_install_dir/access/oblix/config/UserDB.lst and GroupDB.lst

  – Information for all the failover and load-balancing directory servers (primary and secondary) was stored in:

    AccessServer_install_dir/access/oblix/config/UserDBFailover.lst

    and GroupDBFailover.lst

- When creating the directory server profile for the Policy Manager, directory server credentials are read from *PolicyManager_install_dir*/access/oblix/config/userDB.lst.

  Note: If the configuration tree is in the user directory server *and* under the user node, then the configuration directory profile is **not** created. Otherwise, a configuration directory profile is created using directory server information from *PolicyManager_install_dir*/oblix/config/ldap/configdb.lst and marked for use only by the Policy Manager.

- Profiles are not created for Policy Manager failover servers. In the case of release 6.1, if the policy tree was on a separate directory server a profile for policy data existed.
- The Access Server will handle multiple directory servers following data upgrades.

After upgrading, to verify that the failover configuration you had in the previous release operates as expected see:

- Confirming Identity System Failover and Load Balancing
- Confirming Access System Failover and Load Balancing

Former .lst files are transformed into .xml files, as described in "Message and Parameter Files" on page 4-14. See also "Connection Pool Details" on page 4-7.

Directory Server Interface

The 10*g* (10.1.4.0.1) directory server interface reads, processes, and stores data in UTF-8 encoding. Earlier releases behaved in this same way. Therefore, there is no impact in upgraded environments because Oracle Access Manager used UTF-8 encoding for directory server communications even in earlier releases

Directory Structure

Product releases before release 6.5 did not include any language directories, because English was the only language. When you install 10*g* (10.1.4.0.1) components, you can name the top-level directory as you like. During installation, Oracle Access Manager appends an identifier is appended to the directory name you assign to identify the type of components installed therein. For example, the top-level structure is:

- *OracleAccessManager*\access: Created with the installation of the Access Server
- *OracleAccessManager*\identity: Created with the installation of the Identity Server
- *OracleAccessManager*\webcomponent: Created with the installation of Oracle Access Manager Web components (WebPass, Access Manager, WebGate)

Release 6.5 through 10*g* (10.1.4.0.1) installations provide a language directory containing a named subdirectory for each installed language (which contains .XML message catalog files for various applications that you may customize):

IdentityServer_install_dir\identity\oblix\lang\en-us: English messages
IdentityServer_install_dir\identity\oblix\lang\fr-fr: French messages
IdentityServer_install_dir\identity\oblix\lang\shared: default global stylesheets in all languages

For more information, see Appendix A, "Oracle Access Manager Directory Structure Changes".

Domain Names, URIs, and URLs

As in earlier releases of the product, 10*g* (10.1.4.0.1) supports ASCII characters for domain names and Uniform Resource Identifiers (URIs). The most common form of a URI is a Web page address (a subset of the URI is known as a Uniform Resource Locator or URL).

With Oracle Access Manager10*g* (10.1.4.0.1), there is no support for international characters in domain names (internationalized domain names) nor in the Uniform Resource Identifiers (internationalized resource identifiers) nor, by extension, in the URL.

Encryption Schemes

Starting in Oracle Access Manager release 7, AES became the encryption scheme used by Access System components. The Identity System continues to use RC6 encryption for Lost Password Management responses.

The shared secret encryption algorithm is an Oracle Access Manager-wide setting that affects all encrypted cookies. For example, the ObSSOCookie cookie is encrypted using a configurable encryption key known as a *shared secret*.

- For shared secret keys used in release 5.x, the RC4 encryption scheme was recommended.
- For shared secret keys used in release 6.x, the RC6 encryption scheme was recommended. (RC6 encryption is deprecated in Oracle Access Manager 10*g* (10.1.4.0.1), and its support will be deprecated in future releases.)
- AES is a new encryption scheme introduced in release 7.0 which continues in to 10*g* (10.1.4.0.1). AES is the default encryption scheme.

In environments that include earlier WebGates, the earliest encryption algorithm should be used.

For more information, see "Shared Secret" on page 4-29 and details about setting encryption schemes in the *Oracle Access Manager Access Administration Guide*.

Failover and Failback

Oracle Access Manager release 7 introduced a heartbeat polling mechanism to facilitate immediate failover to a secondary directory server when the number of connections in the connection pool is less than the specified threshold level. Additionally, a failback mechanism facilitates switching from the secondary directory server back to the primary server as soon as the preferred connection has been recovered.

The heartbeat feature polls all the primary directory server connections periodically to verify the availability of the directory service (and by implication, the network). You configure the polling interval by setting the `Sleep For (Seconds)` parameter for each Directory Profile in the System Console as described in the *Oracle Access Manager Identity and Common Administration Guide*.

When the host cannot be reached, further attempts to connect to that host are blocked for the specified the `Sleep For` interval, rather than for the TCP timeout used previously.

A new `heartbeat_ldap_connection_timeout_in_millis` parameter in globalparams.xml determines the timeout interval for establishing a connection. The default value for this parameter is 4000 (4 seconds). See the *Oracle Access Manager Deployment Guide*.

If the directory service is not available, the heartbeat mechanism immediately initiates failover to the secondary directory server. Thus, failover can take place without being triggered by an incoming directory service request and a subsequent TCP timeout.

In situations where the enterprise network performance is poor, the heartbeat feature can trigger false alarms and tear down already-established connections. Therefore, the `heartbeat_enabled` parameter in the globalparams.xml files enables you to activate or deactivate the heartbeat mechanism in response to current network conditions. By default the heartbeat feature is activated.

File and Path Names

As with earlier releases, only ASCII characters are supported in file and path names.

> **Note:** Be sure that all file and path names include only English language characters. In file and path names, no international characters are allowed.

Graphical User Interface

A number of changes have been made to improve and clarify the Web-based graphical user interface. These changes are introduced in the *Oracle Access Manager Introduction* and are illustrated throughout the suite of manuals.

HTML Pages

Each Identity System application, such as the User Manager, Group Manager, and Org Manager, generates HTML for each page within the application. Access System components, such as the Policy Manager and WebGate, generate HTML pages. In

earlier releases, HTML pages were generated and displayed using a superset of the Latin-1 character set.

In 10*g* (10.1.4.0.1), all HTML pages generated by Oracle Access Manager use UTF-8 encoding. This encoding is communicated to Web browsers using the Content-Type HTTP header and META tags.

UTF-8 encoding is rendered correctly on all supported browsers with the Unicode version of the font to be used. For browser support, see the *Oracle Access Manager Installation Guide*.

Certain Web servers (Apache, for example) allow administrators to specify the default encoding using the Content-Type HTTP header. However if the Web server setting specifies a different character encoding, Oracle Access Manager HTML pages are displayed incorrectly.

Note: To prevent incorrect behavior, Oracle recommends disabling such Web server settings.

See also, "Default Product Pages" on page 4-10.

Message and Parameter Files

Prior to release 6.5, Oracle Access Manager messages were controlled by an XML file for a specific application and stored in application specific directories. For example:

IdentityServer_install_dir/identity/oblix/apps/*appname*/bin/*appnamemsg*.xml

where *IdentityServer_install_dir* is the directory where the Identity Server is installed and *appname* matches a specific application, as follows:

groupservcenter--Group Manager

objservcenter--Organization Manager

userservcenter--User Manager

In 10*g* (10.1.4.0.1), these message files now reside in language-specific directories. For example: *IdentityServer_install_dir*/identity/oblix/lang/*langTag*/oblixbasemsg.xml.

Earlier release also provided a mix of parameter and message catalogs in .xml and .lst (proprietary) format. For example, Access System and SNMP Agent messages and parameters were stored in .LST files. To accommodate translation, .lst files were converted to .xml.

During an upgrade to Oracle Access Manager10*g* (10.1.4.0.1), any customizing in earlier message and parameter catalogs is preserved automatically. Also, .lst files are converted to XML format.

Note: During the upgrade, messages are displayed in English.

New installations of Oracle Access Manager10*g* (10.1.4.0.1) include .xml parameter and message catalog files. The exception to this rule includes files that are used during an upgrade to Oracle Access Manager10*g* (10.1.4.0.1), such as ois_520_to_600_msg.lst, which remain in the proprietary .lst format. 10*g* (10.1.4.0.1) upgrade tools use .lst message and parameter catalogs.

Names Assigned by Administrators and Product Names

Some product and component names have changed as you will see after an installation or upgrade. During an upgrade, earlier product, component, and functions names are changed to the new name. For example, in 10*g* (10.1.4.0.1), the default Policy domains are Identity Domain and Access Domain and the default authentication schemes are Oracle Access and Identity, Oracle Access and Identity for AD Forest, and Anonymous. These new names replace earlier names during the upgrade.

Certain function names have revised to noun phrases the Access and Identity Systems as noun phrases. The AM Service State name has changed to Policy Manager API Support Mode. For more information, see "Product and Component Name Changes" on page -xxi.

Any names assigned by an administrator during installation and configuration are retained during an upgrade (not changed). Therefore if you have named a service "COREid Identity Server" or "NetPoint Identity Server" these names will be the same in the upgraded environment. Your earlier authentication schemes and policy domains are also retained as is during the upgrade. See also "Preserved Items" on page 3-6.

Namespaces for Policy Data and User Data Stored Separately

Before release 6.5, the namespaces for Policy data and user data stored in two separate directories had to be unique. During an upgrade to 10*g* (10.1.4.0.1) you need to confirm this uniqueness to ensure that multi-language capability can be enabled.

For more information, see "Configuring Unique Namespaces for Directory Connection Information" on page 5-7.

Reconfiguring the Logging Framework without a Restart

In 10*g* (10.1.4.0.1), you may reconfigure the logging framework without restarting the servers. To do this an administrator must manually update the log configuration for each component:

Identity Server
WebPass
Policy Manager
Access Server
WebGate

Changes to logging parameters take effect within one minute, rather than requiring you to restart the server where the changes were made. For more information, see the *Oracle Access Manager Identity and Common Administration Guide*.

Support Changes

There have been a number of changes in supported platforms and third-party versions. You can now locate complete platform support details under the Certify tab at `https://metalink.oracle.com`.

Transport Security for the Directory Server

When you configure SSL mode for the directory server, only server authentication is supported. Client certificates are not supported. Oracle Access Manager verifies the server certificate against the Root CA certificate that you imported during product setup. For more information, see the *Oracle Access Manager Identity and Common Administration Guide*.

Web Components and Backward Compatibility

Earlier WebPass instances are not compatible with 10*g* (10.1.4.0.1) Identity Servers (or Policy Managers). After upgrading all earlier Identity Servers, you must upgrade all earlier WebPass instances. The exception to this rule is when you upgrade the schema and data against the master Identity System that you add for this purpose. For more information, see Part II, "Upgrading the Schema and Data".

You may install 10*g* (10.1.4.0.1) WebPass instances in your upgraded environment. However, 10*g* (10.1.4.0.1) WebPass instances are not compatible with earlier Identity Servers (or Policy Managers).

Release 6.1.1, 6.5, and 7.x WebGates may coexist with upgraded Access Servers. You may install 10*g* (10.1.4.0.1) WebGates in your upgraded environment. However, 10*g* (10.1.4.0.1) WebGates are not compatible with earlier Access Servers. Fore more information, see "WebGates" on page 4-29.

If you add a 10*g* (10.1.4.0.1) Access Server to the upgraded environment, you must set a flag to enable backward compatibility with earlier WebGates. For more information, see "Access Server Backward Compatibility" on page 4-24.

XML Catalogs and XSL Stylesheet Encoding

This discussion outlines the encoding schemes you will see in XML message and parameter catalog files and XSL stylesheet files, and what to specify if you customize these files. See also "Acquiring and Using Multiple Languages" on page 4-4.

ISO-8859-1 Encoding: For pure English text, there is no difference between ISO-8859-1 encoding and UTF-8 encoding. For this reason, the encoding scheme for English language XML message and XSL files remains ISO-8859-1. The following example shows an XML message file (auditmsg.xml), from an English directory (\lang\en-us):

IdentityServer_install_dir\identity\oblix\lang\en-us\auditmsg.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <MessageCtlg xmlns="http://www.oblix.com" CtlgName="auditmsg">
...
```

> **Note:** XML files in earlier product releases may continue to specify `encoding="ISO-8859-1"`, while earlier LST files that have been converted to XML use UTF-8 encoding. See also "Message and Parameter Files" on page 4-14.

The next example illustrates an XSL stylesheet wrapper (style.xsl), which is the same in all language directories: English \lang\en-us, or German \lang\de-de, or Japanese \lang\ja-jp, and so on). The only difference in these files is the language designation specified by the *langtag* item in the last line of this example, which will differ from language to language:

IdentityServer_install_dir\identity\oblix\lang*langtag*\style0\style.xsl

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
 - <!--  Copyright (c) 1996-2005, Oracle All Rights Reserved.  -->
 - <xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:oblix="http://www.oblix.com/">
  <xsl:variable name="styleName">style0/</xsl:variable>
  <xsl:variable name="localeName">langtag</xsl:variable>
```

```
...
```

UTF-8 Encoding: For non-English languages, XML message files have encoding set as UTF-8, because ISO-8859-1 encoding cannot represent all characters in all languages. The sample file shown next is from the German language directory *IdentityServer_install_dir*\identity\oblix\lang\de-de\auditmsg.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <MessageCtlg xmlns:oblix="http://www.oblix.com" CtlgName="">
<Message MsgTag="ExAuditInitHandler">ExçêpäìÖÑExç Öççürrêdöçç ìÑì ähêä AüdìäAü
MÖdülêMÖ ìÑìäìàlìzàäìÖÑìÑìäì. ThêT êxçêpäìÖÑêxç säàçksä ìsì: %1.</Message>
...
```

It is worth mentioning that even within the English language directory (\lang\en-us) some files state UTF-8 encoding because this encoding scheme is universal. For example, the English version of data_types.xml is:

```
<?xml version="1.0" encoding="UTF-8" ?>  <?xml version="1.0" encoding="UTF-8" ?>
- <MessageCtlg xmlns="http://www.oblix.com" CtlgName="data_types.xml">
<Message MsgTag="OB_BIN">Binary</Message>
<Message MsgTag="OB_DN">Distinguished Name</Message>
<Message MsgTag="OB_TEL">Telephone</Message>
...
```

In other language directories, German for example, the same file appears as:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <MessageCtlg xmlns:oblix="http://www.oblix.com" CtlgName="">
<Message MsgTag="OB_BIN">BìÑàryBì</Message>
<Message MsgTag="OB_DN">DìsäìÑgüìshêdDìsä NàmêN</Message>
<Message MsgTag="OB_TEL">TêlêphÖÑêTêl</Message>
...
```

> **Note:** When customizing XML and XSL files, you can choose either `encoding="ISO-8859-1"` or `encoding="UTF-8"`. In either case, the Oracle Access Manager XML parser reads the encoding tag in the file for correct processing.

For more information about XSL stylesheets and wrapper files, see the *Oracle Access Manager Customization Guide*.

Web Server Configuration Files

Security-related changes have been implemented to ensure that sensitive data in the following directories cannot be viewed directly through a browser:

- Configuration files from /access or identity/oblix/config/*.*
- Log files from /access or identity/oblix/log/*.* directory

The importantnotes.txt file has been removed and the information that was in this file is now documented in an appendix in the *Oracle Access Manager Installation Guide*.

There have been no changes for globalization and UTF-8 support in any Web server configuration files.

Identity System Behavior Changes

This discussion provides information about previous Identity System behaviors with a focus on changes and what to expect after upgrading to 10*g* (10.1.4.0.1). Topics include:

- Challenge and Response Attributes
- Identity Server Backward Compatability
- Identity System Event Plug-ins
- IdentityXML and SOAP Requests
- Java Applets
- Mail Notification Enhancements
- Minimum Number of Search Characters
- Multi-Step Identity Workflow Engine
- Oracle Identity Protocol (OIP)
- Portal Inserts and the URI Query String
- Password Policies and Password Management Runtime Changes
- PresentationXML Directories
- Sorting User Search Results

Challenge and Response Attributes

In earlier releases, the challenge phrase and response attributes were allowed on different panels of the User Profile page. In 10*g* (10.1.4.0.1), however, both the challenge phrase and response attributes must be on the same panel. In 10*g* (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 at all. For details about combining these on a single panel, see "Combining Challenge and Response Attributes on a Panel" on page 12-8.

IdentityXML changes have also been made for this feature. For details, see the *Oracle Access Manager Developer Guide*.

Identity Server Backward Compatability

Starting with 10*g* (10.1.4.0.1), the Identity Server uses UTF-8 encoding and plug-in data will contain UTF-8 data. Earlier plug-ins send and receive data in Latin-1 encoding.

When you upgrade earlier Identity Servers, backward compatibility with earlier custom plug-ins is enabled automatically. In this case, a new flag (`encoding`) is added to the oblixpppcatalog.lst file automatically to ensure backward compatibility with earlier plug-ins. A backward-compatible Identity Server continues to send data to earlier plug-ins in Latin-1 encoding.

Caution: When you add a new 10*g* (10.1.4.0.1) Identity Server to an upgraded environment, you must manually edit *IdentityServer_install_dir*\identity\oblix\apps\common\bin\oblixpppcatalog.lst to enable communication with earlier plug-ins and interfaces that need backward compatibility for Latin-1 data. For details, see the *Oracle Access Manager Installation Guide*.

For more information, see the discussion on backward compatibility in "Identity System Event Plug-ins", next. See also "Cache Flush" on page 4-6. Upgraded Identity Servers are not backward compatible with earlier WebPass instances.

Identity System Event Plug-ins

The Identity Event Plug-in API is a standard component installed with the Identity Server that enables you to extend base Identity System functionality by developing your own small applications (called actions) to perform custom business logic and integrate with external systems. The Identity System makes certain data available to the actions, which are then allowed to modify the data and influence the outcome of the event.

Starting with 10*g* (10.1.4.0.1), the Identity Server uses UTF-8 encoding; plug-in data will contain UTF-8 data. Also, on Solaris and Linux, plug-ins earlier than release 7.x must be re-compiled using the GCC v3.3.2 C++ compiler as described in "Plug-ins" on page 3-10.

Identity Event Plug-in Backward Compatibility

In earlier releases, data was sent to Identity Event plug-ins using Latin-1 encoding. In an upgraded environment, any earlier Identity Event plug-in still uses Latin-1 encoding. You may need to redesign earlier custom plug-ins to use UTF-8 encoding. In some cases, however, you may want 10*g* (10.1.4.0.1) Identity Servers to communicate with earlier plug-ins.

Backward compatibility with earlier Identity Event plug-ins is automatic when you upgrade an earlier Identity Server to 10*g* (10.1.4.0.1). During the upgrade, a new flag is added to the oblixpppcatalog.lst file (`encoding`). A backward-compatible Identity Server continues to send data to earlier plug-ins in Latin-1 encoding; earlier plug-ins receive and send data in Latin-1 encoding. There is no change in plug-in data encoding.

When you add a new 10*g* (10.1.4.0.1) Identity Server to an upgraded environment, you need manually set the `encoding` flag in the Identity Server oblixpppcatalog.lst to enable communication with earlier plug-ins and interfaces.

The catalog is stored in *IdentityServer_install_dir*\oblix\apps\common\bin\oblixpppcatalog.lst. It contains event handler entries and their mapping to the various events. The format of the entries is:

```
actionName;exectype;netpointparam1,...;path;execparam,...;apiVersion;encoding;
```

The next sample line shown here illustrates how to use the encoding flag to enable backward compatibility with Latin-1 plug-ins:

```
userservcenter_view_post;lib;;..\..\..\unsupported\ppp\ppp_dll
\ppp_dll.dll;PostProcessingTest;Latin-1;
```

Note: In the catalog file, the `encoding` flag is similar to the `apiVersion` flag, which sets the version of the Event API to be used by the event handler. As described in the catalog file, `apiVersion` can be used to set backward compatibility for the Event API. For example, if `apiVersion` is set to preNP60 then the API format for versions prior to Oracle Access Manager v60 and Latin-1 encoding is used by default. In this case, setting the encoding flag is redundant.

Common Uses of the Identity Event Plug-in API

Common uses of the Identity Event Plug-in API include password validation, integration, and provisioning. For example, you can develop an event handler for password management events that use the Identity Event API and add this event handler to the Oracle Access Manager password policy function. Or, you can develop an event handler for the Enable step of each registration workflow instance to either update the remote database using the RDBMS vendor's API or to generate a unique string in the required format and pass it back to the Identity System to use as the uid attribute value. For details, see the *Oracle Access Manager Developer Guide*

Identity Event Plug-in Action Types

An action is a unit of external logic (also known as an event handler) written by a developer and configured by an Oracle Access Manager administrator to execute in response to a particular event. Actions may perform their tasks without accessing external components, or use any available mechanism to access third-party applications and resources such as Web services, RDBMS services, and ERP applications. You connect actions to the event using the oblixpppcatalog.lst file. At startup the Identity Server reads the catalog, which identifies the events that have actions. When an event occurs, the server executes the associated action.

There are three types of actions with the Identity Event Plug-in API:

- **LIB Action**: A function within a shared library (DLL on Windows systems) that the Identity Server calls. Once dynamically loaded, the action executes in the same process space as the Identity Server and has direct access through API functions to data objects held by the server. The Identity Server sends a C++ object containing plug-in data to the library.
- **MANAGEDLIB Action**: A function for Windows systems only, written in any .NET language for which a Microsoft Intermediate Language (MIL) compiler exists. MIL instructions are compiled once into native machine instructions and stored in dynamic memory, then executed by the Microsoft .NET Common Language Runtime (CLR). MANAGEDLIB actions are similar to LIB actions with the benefits of managed code. The Identity Server sends a C++ object containing plug-in data to the library.
- **EXEC Action**: A standalone executable program that run in their own process space. Communication with the Identity Server is limited to startup parameters and an XML stream for input, and an XML stream plus an exit status code for output. Actions can also use any other APIs, such as an LDAP Identity Event Plug-in.

Identity Event Plug-in Event Types

An event is a state change within the Identity System. Examples of events include when a request is received and is about to be passed to an application (such as the User Manager view program), or results have been generated by an application (such

as the Group Manager search program), or a user has entered a challenge response while attempting a password reset, or an attribute on a profile page for an application (such as the Organization Manager) has been modified, or a workflow ticket awaiting approval the by corporate IT group has been approved.

The most frequently used type of events are pre-processing and post-processing events, which are generated in pairs. Each application (User, Group, or Org Manager) contains a number of programs (view, search, and so on) that generate HTML for each page within the application. Each program recognizes the event pair. Pre-processing events are generated before the program begins to create the page and allows an event handler to work with a request before it reaches a program. The Post-processing event is generated after the program has created the page and before responding to the user with an HTML page. The post-processing event allows an event handler to work with the results of processing a request.

IdentityXML and SOAP Requests

Rather than interacting with the application through a browser, you can write a program. IdentityXML provides a programmatic interface for carrying out the actions that a user can perform when accessing an Identity System application from a browser. IdentityXML enables you to process simple actions and multi-step workflows to change user, group, and organization object profiles. IdentityXML allows external applications to access Identity System functions.

Starting with release 6.5, certain syntax changes were made for IdentityXML requests. Earlier syntax should still operate without problem. For new syntax descriptions, see the *Oracle Access Manager Developer Guide*.

In 10*g* (10.1.4.0.1), UTF-8 encoding is used for XML pages, for SOAP/IdentityXML requests, and for Identity Event Plug-in data sent to executables. Earlier releases used ISO-8859-1 encoding (also known as Latin-1).

To provide backward compatibility, 10*g* (10.1.4.0.1) supports IdentityXML requests in both ISO-8859-1 encoding and UTF-8. For XML documents written to disk, both ISO-8859-1 and UTF-8 encoding are supported. However, IdentityXML responses are emitted in only UTF-8 encoding.

IdentityXML changes have also been made in 10*g* (10.1.4.0.1) to accommodate challenge and response phrase changes. For details, see the *Oracle Access Manager Developer Guide*.

Java Applets

An applet is a small program that is sent to a user along with a Web page. Java applets perform interactive animations, immediate calculations, or other simple tasks without having to send a user request back to the server.

In earlier releases, the Identity System Console included a drop-down list that enumerated the languages that were installed and configured in the product. When the user changed the language in this list, applets and other pages would be rendered in the selected language. For example, a user working in an English locale could work with applets displayed in a European language simply by selecting the language in the drop-down list. This model worked well for only European languages.

With the introduction of multibyte languages, such as Japanese, the model has changed to ensure that multibyte characters are rendered correctly. The language list has been eliminated from the Identity System Console. A user working in an English locale cannot view applets in multibyte languages. To work with applets in a multibyte language, the locale on the user's machine must be set to the same language.

Note: There is a known limitation of Java applets in JDK1.1.7. Oracle Access Manager 10*g* (10.1.4.0.1), applets with non-ASCII data can only be displayed properly on machines running with a native encoded operating system. Setting browser encoding will not work.

There are no JavaScript changes that impact the user experience.

Mail Notification Enhancements

Identity Server sends notification mails for various functions, such as attribute modification, workflows, containment limit, and others. The formats that are available for mail include text only, rich HTML, and MHTML (MIME encapsulation of aggregate documents, such as HTML. Both asynchronous and synchronous modes are supported when sending mail. The Identity Server communicates directly with the mail server using the SMTP protocol.

Earlier releases used ISO-8859-1 (Latin-1) "Q" encoding for the header messages, which is a recommended standard when most of the characters to be encoded are in the ASCII character set. In 10*g* (10.1.4.0.1) uses UTF-8 "B" (Base64 encoding) encoding is used.

MIME headers for all non-MHTML mail message are set as follows:

```
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8;
Content-Transfer-Encoding: 8bit
```

Minimum Number of Search Characters

In previous releases, you needed to enter at least three characters when performing a search in Identity System applications (User Manager, Group Manager, and Organization Manager). In 10*g* (10.1.4.0.1) there is no minimum number of characters required. By default, you can enter no characters. As in previous releases, 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.

Multi-Step Identity Workflow Engine

You can model your business processes in the Identity System using workflows. In earlier releases, you could use workflows to issue, revoke, and renew certificates. However, this is no longer supported.

Oracle Identity Protocol (OIP)

The Oracle Identity Protocol (formerly known as the NetPoint or COREid Identity Protocol) facilitates communication between Identity Servers and associated WebPass instances. There are no changes in the protocol for globalization. See also "Oracle Access Protocol (OAP) Updates" on page 4-28.

Password Policies and Password Management Runtime Changes

You can use the Identity System to define policies to constrain passwords. These policies are enforced at runtime and include such items as:

- Minimum password length
- Minimum number of uppercase characters
- Minimum number of lowercase characters
- Minimum number of non-alphanumeric characters
- and the like

In 10*g* (10.1.4.0.1), internationalized characters are supported in password policies.

In earlier releases, password policies worked only with Latin1 characters when enforcing policy constraints. There are no Password Management runtime changes.

Portal Inserts and the URI Query String

A Web page address is commonly known as a Uniform Resource Locator (URL), which is a subset of the Uniform Resource Identifier (URI). The encoding of data in the query string of the URI has changed from Latin-1 (in earlier releases) to UTF-8 encoding in 10*g* (10.1.4.0.1).

In new 10*g* (10.1.4.0.1) installations, the change is transparent. However, earlier Portal Inserts in installations that have been upgraded to 10*g* (10.1.4.0.1) require modification. After upgrading the environment to 10*g* (10.1.4.0.1), you must change the encoding of the query string in earlier Portal Inserts from Latin-1 to UTF-8.

The HTTP standard does not provide any mechanism for a browser to specify the encoding of the query string. Oracle Access Manager 10*g* (10.1.4.0.1) cannot detect query string character encoding and assumes it to be UTF-8. The 10*g* (10.1.4.0.1) Identity Server cannot process Latin-1 data from earlier Portal Inserts.

Note: After upgrading the environment to 10*g* (10.1.4.0.1), you must change the encoding of the query string in earlier Portal Inserts from Latin-1 to UTF-8.

PresentationXML Directories

Before release 6.5, the PresentationXML library was provided under two directories and distributed depending upon how the files were likely to be used. For example, stylesheets that define the default Oracle Access Manager Classic Style were maintained in flat files in the file system directory *IdentityServer_install_dir*\identity\oblix\apps*AppName*. Starting with release 6.5, and continuing through 10*g* (10.1.4.0.1), the PresentationXML library are now stored in different directories:

IdentityServer_install_dir\identity\oblix\apps*AppName*\bin
IdentityServer_install_dir\identity\oblix\lang*langTag*
IdentityServer_install_dir\identity\oblix\lang*langTag*\style0
IdentityServer_install_dir\identity\oblix\lang\shared

WebPass_install_dir\identity\oblix\lang*langTag*
WebPass_install_dir\identity\oblix\lang*langTag*\style0
WebPass_install_dir\identity\oblix\lang\shared
WebPass_install_dir\identity\oblix\WebServices\XMLSchema

For more information, see "About Custom Items and Upgrades" on page 12-11.

Sorting User Search Results

In the User Manager, Group Manager and Org. Manager, search results are sorted using a locale-based case insensitive method when you click the column heading (Full Name, for example) in the search results table.

Access System Behavior Changes

This discussion provides information about previous behaviors of the Access System. The focus is on what to expect after upgrading to 10*g* (10.1.4.0.1). Topics include:

- Access Server Backward Compatibility
- Access Manager SDK, Access Manager API, and Custom AccessGates
- Authorization Rules and Access Policies
- Custom Authentication and Authorization Plug-ins and Interfaces
- Directory Profiles
- Forms-based Authentication
- Maximum Elements in Session Token Cache
- Oracle Access Protocol (OAP) Updates
- Policy Manager
- Policy Manager API
- Preferred HTTP Host
- Shared Secret
- Triggering Authentication Actions After the ObSSOCookie Is Set
- WebGates

Access Server Backward Compatibility

In releases before 10*g* (10.1.4.0.1), cookie encryption and decryption was handled by WebGate/AccessGate. However, cookie encryption and decryption is now handled by the Access Server. For this reason, earlier Access Servers are not compatible with 10*g* (10.1.4.0.1) WebGates. See also "Encryption Schemes" on page 4-12.

Starting with Oracle Access Manager 10*g* (10.1.4.0.1), the Access Server uses UTF-8 encoding and plug-in data will contain UTF-8 data. Earlier plug-ins send and receive data in Latin-1 encoding.

When you upgrade earlier Access Servers, backward compatibility with earlier custom plug-ins and earlier WebGates is enabled automatically. In this case, a new parameter `"IsBackwardCompatible" Value="true"` is set in the Access Server globalparams.xml file automatically. This provides backward compatibility that enables the Access Server to continue to send (and receive) data to earlier custom authentication and authorization plug-ins in Latin-1 encoding (and earlier custom plug-ins will set data in Latin-1 encoding). In addition, the Access Server maintains backward compatibility with earlier WebGates and custom AccessGates that continue to encrypt/decrypt cookies

Caution: When you add a new 10*g* (10.1.4.0.1) Access Server to an upgraded environment, you must manually set "`IsBackwardCompatible" Value="true`" in the new Access Server globalparams.xml to enable communication with earlier plug-ins and interfaces, as well as earlier WebGates and custom AccessGates. For details, see the *Oracle Access Manager Installation Guide*.

For more information, see "Custom Authentication and Authorization Plug-ins and Interfaces" on page 4-26 and "Oracle Access Protocol (OAP) Updates" on page 4-28.

Access Manager SDK, Access Manager API, and Custom AccessGates

The Access Manager SDK (formerly known as the Access Server SDK) is an optional component that provides all the documentation, resources, and code samples that you need to construct simple custom AccessGate servlets or applications for each of the supported development platforms. AccessGates are Access Server clients (or agents) that process user requests for access to resources within the LDAP domain protected by Oracle Access Manager. The code for processing user requests can be embedded in a plug-in or written as a standalone application.

After installing the Access Manager SDK, you can use the Access Manager API (formerly known as the Access Server API) to write custom AccessGate code in any of the four supported development languages: Java, C and C++, and C# (.NET). The four implementations are functionally equivalent even though each takes advantage of platform-specific features to implement the API.

While you can select any of the four implementations as the development language interface you use to write your custom AccessGate code, your code will interact with underlying C++ binaries in the API, as described in the *Oracle Access Manager Developer Guide*.

When you develop custom AccessGates using the 10*g* (10.1.4.0.1) C and C++ Access Manager APIs, data is sent and received in UTF-8 encoding automatically. In earlier releases, data was sent and received in Latin-1 encoding.

For the C# (.NET) Managed Code implementation of the Access Manager API, there have been no external changes for 10*g* (10.1.4.0.1). The C# .NET implementation internally uses UTF-16 encoding, which was converted to Latin-1 in earlier Oracle Access Manager releases. 10*g* (10.1.4.0.1) Access Servers and C# AccessGates use UTF-8 encoding automatically.

For Java interfaces and the Java implementation of the Access Manager API, there have been no external changes for 10*g* (10.1.4.0.1). JNI calls use UTF-16 encoded Java string objects. Earlier Oracle Access Manager releases converted this data to Latin-1. 10*g* (10.1.4.0.1) Access Servers and AccessGates use UTF-8 encoding automatically.

Note: The 10*g* (10.1.4.0.1) Access Manager SDK and custom 10*g* (10.1.4.0.1) AccessGates are **not** backward compatible with earlier Access Servers, nor with the earlier Access Manager SDK and AccessGates. However, you can use earlier AccessGates with 10*g* (10.1.4.0.1) Access Servers that are enabled to be backward compatible. See also "Oracle Access Protocol (OAP) Updates" on page 4-28.

Custom AccessGates (and WebGates) no longer perform cookie encryption and decryption. As a result, these components no longer need the shared secret key.

Authentication Scheme Updates

In 10*g* (10.1.4.0.1) it is no longer necessary to disable an authentication scheme before you modify it. Also, in 10*g* (10.1.4.0.1) you can configure an authentication scheme that allows the user to log in for a period of time rather than a single session.

Authorization Rules and Access Policies

In release 6.1.1, Authorization Rules were attached to particular access policies. Starting with release 6.5 (and later), Authorization rules are grouped under a different tab (named "Authorization Rules").

During an upgrade, the name of an Authorization Rule is shifted to the Authorization Rules tab. In addition, the name becomes a combination of the Policy name to which the rule belongs, followed by the Authorization Rule name: *PolicyName_AuthorizationRuleName*. For more information about recognizing and handling Authorization Rules after the upgrade, see "Associating Release 6.1.1 Authorization Rules with Access Policies" on page 13-5.

Also, a new authorization inconclusive state was introduced in release 7.x (apart from authorization success and failure states). When your earlier installation included authorization failure redirects, you need to complete a procedure after the upgrade to specify an explicit Deny rule and to change `Allow takes precedence` to `Yes` under the General tab of the authorization rule. For more information, see "Assuring Proper Authorization Failure Re-directs After Upgrading from 6.1.1" on page 13-6.

Custom Authentication and Authorization Plug-ins and Interfaces

With 10*g* (10.1.4.0.1) there are some changes and backward compatibility, considerations as described here.

Authentication is the process of determining that a user trying to access a protected resource is who they say they are. Authorization is the process of determining that an authenticated user has access rights for the protected resource. The Access Server uses both authentication and authorization controls to limit access to resources that it protects, and provides defined interfaces that interact with authentication and authorization plug-ins.

You can either use standard authentication and authorization plug-ins or create your own custom plug-ins using the Oracle Access Manager Authentication Plug-In API and Authorization Plug-In API. Each custom plug-in implements the appropriate interface (authentication or authorization). Depending on the plug-in, the interface is activated to pass relevant information between the Access Server and the plug-in. Methods within the interface parse the data.

Before 10*g* (10.1.4.0.1), the Authentication Plug-In API and Authorization Plug-In API for C used Latin-1 encoding for data exchanged between the Access Server and the custom plug-ins. However, 10*g* (10.1.4.0.1) the Authentication Plug-In API and Authorization Plug-In API for C use UTF-8 encoding for plug-in processing.

There is no change for .NET (managed code) plug-ins, which continue to use the same API interface as in earlier releases of Oracle Access Manager.

Access Server Backward Compatibility

You may need to redesign earlier custom plug-ins to use UTF-8 encoding. In some cases, however, you may want 10*g* (10.1.4.0.1) Access Servers to communicate with earlier plug-ins.

An earlier Access Server that is upgraded to 10*g* (10.1.4.0.1) provides backward compatibility automatically. However, when you add a new 10*g* (10.1.4.0.1) Access Server to an upgraded environment, you need manually set backward compatibility. For more information, see "Access Server Backward Compatibility" on page 4-24.

Authentication and Authorization Plug-ins Background

This discussion provides an overview of authentication and authorization plug-ins in Oracle Access Manager.

Authentication is governed by authentication rules. Authentication rules use authenticating schemes; the schemes use one or more plug-ins to test the credentials provided by a user. Standard authentication plug-ins are provided as part of the Access Server installation or you can create your own custom plug-ins using the Authentication Plug-In API.

Authorization is governed by a policy domain that includes an authorization expression among a set of default rules that specify how resources for this domain are protected. Authorization rules are combined to create authorization expressions. When you create a rule, you include an authorization scheme in it. You can use the authorization scheme provided by the Access System or configure one or more custom ones schemes that include custom plug-ins created using the Authorization Plug-In API.

Directory Profiles

Release 6.5 introduced support for directory server profiles for the Access Server and Policy Manager. During a Policy Manager upgrade from any release before 7.x, a new directory server profile is added automatically. However, the values for `Initial Connections` and `Maximum Connections` are not retained during the Policy Manager upgrade

After upgrading, Oracle recommends that you verify and validate that new directory server profiles were properly created and that load-balancing and failover settings in Access System directory server profiles are configured as expected.

For more information, see "Directory Profiles and Database Instance Profiles" on page 4-10.

Forms-based Authentication

10*g* (10.1.4.0.1) WebGates accept input data only in UTF-8 encoding. As a result, in 10*g* (10.1.4.0.1), form-based authentication supports non-ASCII login credentials (username/password). When you use form-based authentication with 10*g* (10.1.4.0.1) WebGates, you must ensure that character set encoding for the login form is set to UTF-8.

To set the login form encoding to UTF-8 after an upgrade, see "Upgrading Forms-based Authentication" on page 13-4.

Note: Basic Authentication fails with non-ASCII login credentials. Use form-based authentication for non-ASCII login credentials. Use Basic Authentication with ASCII login credentials.

Maximum Elements in Session Token Cache

In earlier releases, the default value for this parameter was 100000. However, in Oracle Access Manager 10*g* (10.1.4.0.1), the default value has changed to 10000. You can find this parameter by navigating to the Access System Console, Access System Configuration tab, Access Server Configuration function. Look on the Details for Access Server page.

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

Oracle Access Protocol (OAP) Updates

The Oracle Access Protocol (formerly known as the NetPoint or COREid Access Protocol) enables communication between Access System components during user authentication and authorization. WebGates and AccessGates store the user information required for authentication and authorization for example, (login name, password, headers, and the like). The data is serialized and sent to the Access Server where it is deserialized. The Access Server sends results back to the Access clients.

In earlier product releases, Latin-1 encoding was used for data as it was sent and received. In 10*g* (10.1.4.0.1), UTF-8 encoding is used. An updated Oracle Access Protocol is provided to accommodate both globalization and shared secret generation for 10*g* (10.1.4.0.1) Access Servers.

In new 10*g* (10.1.4.0.1) installations, you do not need to take any action. The latest version of Oracle Access Protocol is used for all communication between Access Servers and associated WebGates/AccessGates, as well as between Access Servers and new standard and custom authentication and authorization plug-ins.

In upgraded environments, Access Server backward compatibility is provided as discussed in "Access Server Backward Compatibility" on page 4-24.

See also "Oracle Identity Protocol (OIP)" on page 4-22.

Policy Manager

After upgrading all Identity System components, you must upgrade all earlier Policy Managers (formerly known as the Access Manager component).

Policy Manager API

The Access System provides programmatic access to most of the functions provided by the Policy Manager graphical user interface (GUI). However, you can use the Policy Manager API (formerly known as the Access Management API) to create and manage policy domains and their contents or to allow custom applications to access the authentication, authorization, and auditing services of the Access Server. As in earlier releases, the 10*g* (10.1.4.0.1) Policy Manager API provides Java, C, and C# (.NET managed code) bindings for classes.

In earlier releases, `ObAMMasterAuditRule_getEscapeCharacter` returned the audit escape character.

In Oracle Access Manager 10*g* (10.1.4.0.1):

- In the C language API, the `ObAMMasterAuditRule_getEscapeCharacter` remains and you may continue using this. However, the audit escape character must be an ASCII character; otherwise the return value is incorrect. In this case, you must modify your C code to use the new API.
- On Java clients, the `ObAMMasterAuditRule_getEscapeCharacter` works correctly and you can continue using this even when the audit escape character is not an ASCII character.
- In the C language API, a new `ObAMMasterAuditRule_getUTF8EscapeCharacter` has been added, which returns a pointer to the UTF-8 encoded audit escape character.

Preferred HTTP Host

This WebGate configuration parameter is now mandatory and must be configured with an appropriate value whenever a WebGate is added (from the Access System Console, select Access System Configuration, Add New AccessGate). This must be done before WebGate installation.

The Preferred HTTP Host parameter defines how the hostname appears in all HTTP requests as users attempt to access the protected Web server. The hostname within the HTTP request is translated into the value entered into this field (regardless of the way the hostname was defined in an HTTP request from a user). This safeguard prevents a hacker from constructing a malicious HTTP request that could bypass the WebGate. For more information, see the *Oracle Access Manager Access Administration Guide*.

Shared Secret

In earlier releases, the shared secret was stored in the directory server and cookie encryption and decryption was accomplished by WebGates and custom AccessGates. In 10*g* (10.1.4.0.1), the shared secret remains in the directory server; however, cookie encryption and decryption is accomplished by the Access Server. As a result, WebGates and AccessGates no longer need the shared secret key.

If you change the shared secret during a user session, the user does not need to re-authenticate. If a cookie is being decrypted with the old shared secret and the cookie is refreshed, it is encrypted with the new shared secret. For more information, see the *Oracle Access Manager Access Administration Guide*.

For details about Access Servers, see "Access Server Backward Compatibility" on page 4-24. For details about WebGates, see "WebGates" on page 4-29. See also "Encryption Schemes" on page 4-12.

Triggering Authentication Actions After the ObSSOCookie Is Set

You can cause authentication actions to be executed after the ObSSOCookie is set. Typically, authentication actions are triggered after authentication has been processed and before the ObSSOCookie is set. However, in a complex environment, the ObSSOCookie may be set before a user is redirected to a page containing a resource. In this case, you can configure an authentication scheme to trigger these events. See also *Oracle Access Manager Access Administration Guide*.

WebGates

Release 6.1.1, 6.5, and 7.x WebGates may coexist with upgraded Access Servers. You may install 10*g* (10.1.4.0.1) WebGates in your upgraded environment. However, 10*g* (10.1.4.0.1) WebGates are not compatible with earlier Access Servers.

The WebGateStatic.lst file available in earlier releases no longer exists. Instead, with 10*g* (10.1.4.0.1) WebGates you configure such parameters as IPValidation and IPValidationExceptions from the Access System Console, as described in the *Oracle Access Manager Access Administration Guide*.

In releases before 10*g* (10.1.4.0.1), cookie encryption and decryption was handled by WebGate/AccessGate. However starting with 10*g* (10.1.4.0.1), cookie encryption and decryption is now handled by the Access Server.

The code for WebGates has been rewritten so that 10*g* (10.1.4.0.1) WebGates and AccessGates share the same code base. For more information, see the *Oracle Access Manager Developer Guide*.

Oracle recommends that you upgrade all earlier WebGate even though these may coexist with 10*g* (10.1.4.0.1) Access Servers. In environments that include a mix of WebGate releases, use the encryption scheme that corresponds to the earliest WebGate. For example:

- Use RC4 as the encryption scheme if you have release 5.x and 10*g* (10.1.4.0.1) WebGates co-existing in the same system.
- Use RC6 as the encryption scheme if you have release 6.x and 10*g* (10.1.4.0.1) WebGates co-existing in the same system.
- Use the AES encryption scheme if you have only release 7.0 or 10*g* (10.1.4.0.1) WebGates co-existing in the same system.

As discussed earlier, if you install a 10*g* (10.1.4.0.1) Access Server in an upgraded environment that includes earlier WebGates/AccessGates, you must manually configure the Access Server for backward compatibility. For more information, see "Access Server Backward Compatibility" on page 4-24.

Part II

Upgrading the Schema and Data

This part of the book explains how to prepare your earlier environment for the schema and data upgrade, then how to upgrade the schema and data with master components installed for this purpose.

Part II contains the following chapters:

- Chapter 5, "Preparing for Schema and Data Upgrades"
- Chapter 6, "Upgrading Identity System Schema and Data"
- Chapter 7, "Upgrading Access System Schema and Data"

5

Preparing for Schema and Data Upgrades

This chapter is intended for directory server administrators who are responsible for maintaining and updating directory schemas and data. Here you will find information on preparing the environment for the Oracle Access Manager (formerly known as Oblix NetPoint or Oracle COREid) schema and data upgrade. The following topics provide information to help you prepare your environment:

- About Schema and Data Upgrades
- Strategies for Upgrading in a Replicated Environment
- Configuring the Challenge/Response Phrase at the Object Class Level
- Configuring Unique Namespaces for Directory Connection Information
- Preparing Your Directory Instances for the Schema and Data Upgrade
- Backing Up Existing Oracle Access Manager Data
- Backing Up Existing Directory Instances
- Preparing Host Machines for Master Components
- Adding An Earlier Identity System to Use as a Master
- Adding an Earlier Access Manager to Use as a Master
- Finishing Preparation

Note: If your starting Oracle Access Manager release is earlier than 6.1.1, contact Oracle Support before upgrading:
`http://www.oracle.com/support/contact.html`

About Schema and Data Upgrades

There are several types of data used by Oracle Access Manager: user data, configuration data, and policy data. User data refers to the enterprise identity store (LDAP) that Oracle Access Manager is configured to work against. Configuration is metadata pertaining to Oracle Access Manager configuration that is stored in directory. Policy data is metadata pertaining to access policies that is stored in the directory.

A schema upgrade occurs when you upgrade the master Identity Server. Configuration data is also upgraded during the master Identity Server (formerly known as the COREid Server) upgrade. When you upgrade additional Identity Server instances, the initial schema and data upgrade is detected automatically. Further Identity System schema and data upgrades are not requested.

Policy data is upgraded with the master Policy Manager (formerly known as the Access Manager component). When the configuration tree and policy node are in the same directory server, the master Identity Server upgrade touches only configuration data. Policy data is upgraded only when the master Policy Manager is upgraded. If you have a large number of entries in the configuration tree, data migration may take a while to complete.

Additional schema updates are not typically required with policy data unless you have several directory instances configured as shown here:

Directory_1 communicates with the Identity System

Directory_2 communicates with the Identity System *and* Access System

Directory_3 communicates with the Access System

In such cases, the Oracle Access Manager schema and configuration data are upgraded on Directory_1 and Directory_2 during the master Identity Server upgrade. The schema and policy data are upgraded on Directory_2 and Directory_3 during the master Policy Manager upgrade.

No schema upgrade occurs during Access Server, WebPass, or WebGate upgrades. A data upgrade occurs automatically during each Access Server upgrade and a directory server profile is created for each Access Server.

For more information, see:

- Considerations for Workflows in Multiple Directories
- About Preparing For and Performing the Schema and Data Upgrade
- Error Logging for All Directory Servers

Considerations for Workflows in Multiple Directories

Oracle recommends that you keep all workflows on one directory server. When workflows are stored on multiple directory servers, you cannot automatically upgrade the schema and data.

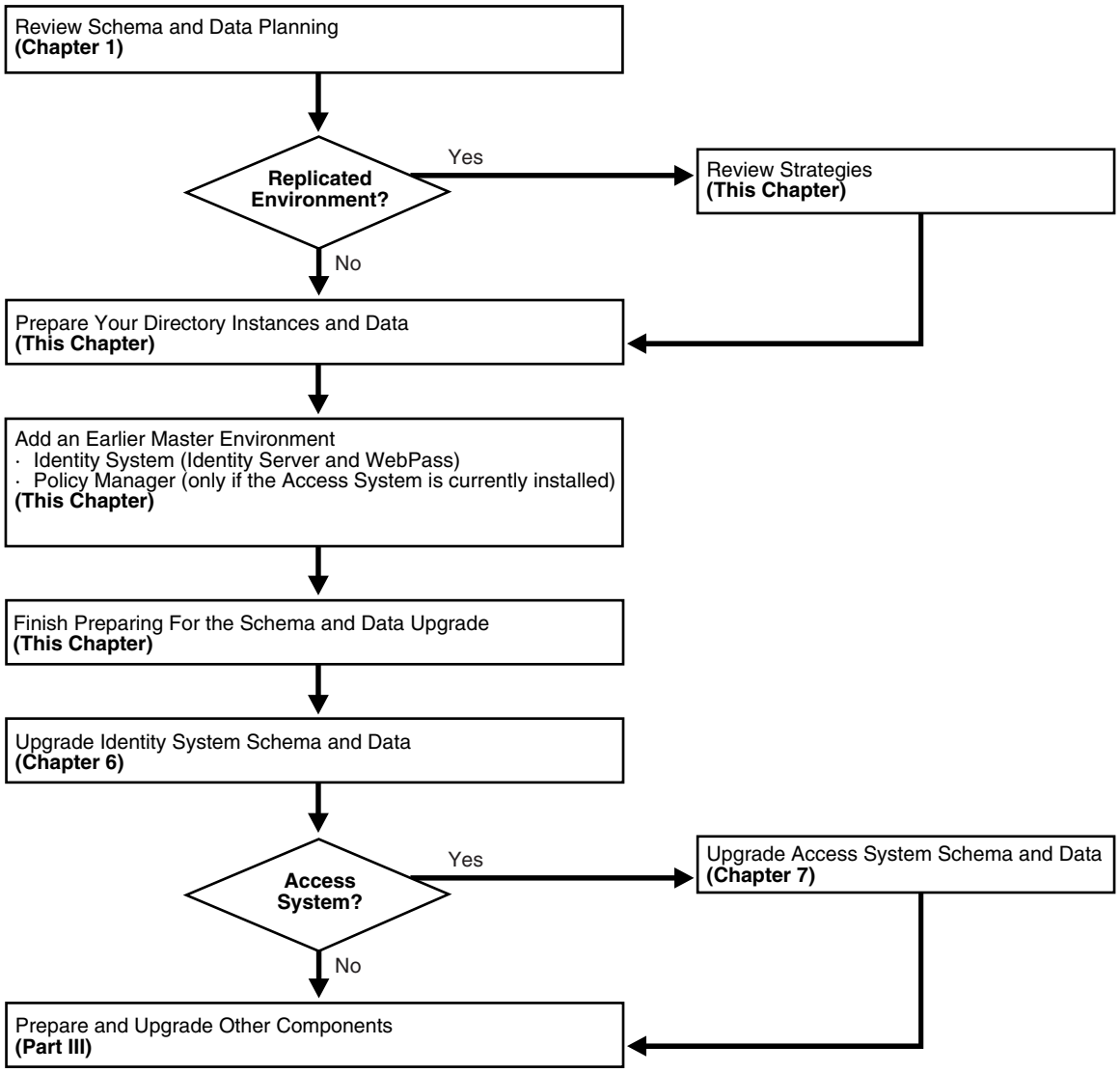
If your installation includes workflows on separate directory servers, you must manually upgrade the schema and data. In this case, see Appendix C, "Manual Schema and Data Upgrades".

About Preparing For and Performing the Schema and Data Upgrade

Worksheets that you can use to record details about your existing deployment are provided in Appendix E, "Planning Worksheets and Tracking Checklists". Checklists that can help you track the completion of upgrade tasks in your environment are also provided in Appendix E, "Planning Worksheets and Tracking Checklists".

Figure 5–1 illustrates the tasks involved in preparing for and performing the schema and data upgrade. Additional information follows the figure.

Figure 5–1 Schema and Data Upgrade Task Overview



Task overview: Preparing for and performing schema and data upgrades

1. Review the following topics to gain an understanding of the conditions that may govern the sequence and tasks you must perform:
   - Schema and Data Upgrade Planning on page 1-12
   - Strategies for Upgrading in a Replicated Environment
2. **Prepare Directory Instances and Data**: Perform the tasks in the following list to prepare your directory instances and data for the upgrade, as described in:
   - Configuring Unique Namespaces for Directory Connection Information
   - Configuring the Challenge/Response Phrase at the Object Class Level
   - Preparing Your Directory Instances for the Schema and Data Upgrade
   - Backing Up Existing Oracle Access Manager Data
   - Backing Up Existing Directory Instances

3. Perform the next set of tasks to create a master environment of secondary servers for your original master Read/Write directory server instances:
   - Preparing Host Machines for Master Components
   - Adding An Earlier Identity System to Use as a Master
   - Adding an Earlier Access Manager to Use as a Master
4. Finish preparing for the schema and data upgrade as described in "Finishing Preparation" on page 5-29.
5. Upgrade the schema and data in sequence, as described in:
   - Chapter 6, "Upgrading Identity System Schema and Data"
   - Chapter 7, "Upgrading Access System Schema and Data"
6. When the schema and data upgrade is successful, prepare for and upgrade the remaining components as described in Part III, "Upgrading Components".

Error Logging for All Directory Servers

During data migration, the following two files are created, regardless of your directory server type. Both contain new Oracle Access Manager data, generated after applying the upgrade for the specific incremental release. The older Oracle Access Manager tree is deleted and the appropriate file is uploaded to generate the upgraded tree:

- During Identity Server data migration, output_*fromversion*_to_*toversion*_osd.ldif is created in the *IdentityServer_install_dir*\identity\oblix\tools\migration_tools\obmigratedata directory.
- During Policy Manager data migration, output_*fromversion*_to_*toversion*_psc.ldif file is created in the *PolicyManager_install_dir*\access\oblix\tools\migration_tools\obmigratedata directory

Additionally, the files listed next are created to log any ldap specific errors as follows:

- During Identity Server data migration, error_output_*fromversion*_to_*toversion*_osd.ldif file is created in the *IdentityServer_install_dir*\identity\oblix\tools\migration_tools\obmigratedata directory.
- During Policy Manager data migration, error_output_*fromversion*_to_*toversion*_psc.ldif file is created in the *PolicyManager_install_dir*\access\oblix\tools\migration_tools\obmigratedata directory

For more information, see "Accessing Log Files" on page F-1.

Strategies for Upgrading in a Replicated Environment

Your installation may employ replicas to increase system availability and improve performance. Using replicas in a failover configuration helps increase system availability, which is important for enterprise-class applications. Replicas can be used in load-balancing configurations to enhance the performance and throughput of the application.

This discussion introduces additional tasks that you must perform when upgrading in a replicated environment. When performing the next tasks, use your directory vendor documentation as a guide unless otherwise indicated. Additional information follows the task overview.

Task overview: Upgrading in a replicated environment

1. Disable the replication agreement.
2. Prepare for and perform the schema and data upgrade as outlined in the previous task overview.
3. Stop the Identity Server, Policy Manager, and Access Server from the original deployment.
4. Re-establish the replication agreement.
5. Push changes to the replicas.
6. After changes have been pushed to the replicas, then upgrade the components configured against these replicas as described in Part III, "Upgrading Components".

> **Note:** The upgrade tools automatically detect that the schema and data have been upgraded and these steps are suppressed when upgrading remaining components.

For more information, see:

- About User Data Replication
- About Configuration Data Replication

About User Data Replication

Your deployment may be architected to leverage user data replicas in various ways. For example, to achieve failover or load-balancing and the like, as described in the following discussions:

- Failover Configuration
- Load Balancing Configuration
- Load Balancing and Failover Configuration
- Operation-based Load Balancing Configuration

Failover Configuration

Suppose that you have one master directory server (named M) and one replica (named R). To setup a failover configuration, you need one DB Profile configured with the primary server named M and a secondary server named R. In this case, Oracle Access Manager will failover to R when M is not reachable.

Upgrade Consideration: This directory server configuration with replica will be a multi-master deployment. Hence, user schema should be uploaded against the master instance M.

Load Balancing Configuration

This scenario is quite similar to the failover configuration in the preceding discussion. Except that in this situation there are multiple primary LDAP servers to help balance the load. For example, suppose that you have one master directory server (named M) and one replica (named R). To setup a load-balancing configuration, you will create one DB Profile with two primary servers configured. In this case, both server M and R will be configured as primary.

Upgrade Consideration: This is the same consideration as in the failover configuration in the preceding discussion. The directory server configuration with replica will be a multi-master deployment. Therefore, the user schema should be uploaded against the master instance (M).

Load Balancing and Failover Configuration

In this configuration you will have multiple primary servers along with secondary servers configured. Let us consider that you have one master directory server (say M) and two replicas (say R1 and R2). There will be one DB Profile configured with two primary servers as M and R1, and there will be one secondary server, R2.

You can configure the 'Failover threshold' for this DB Profile as 1 to indicate that after one primary server goes down start using secondaries along with remaining primaries.

Upgrade Consideration: This too is the same consideration as in the failover configuration in the preceding discussion. The directory server configuration with replica will be a multi-master deployment. Therefore, the user schema should be uploaded against the master instance (M).

Operation-based Load Balancing Configuration

This deployment configuration is typically employed when you have one Read/Write master and one read-only replica. For example, suppose that the Read/Write master is M and the read-only replica is R. In this case, you must configure two DB Profiles for the same namespace. One DB profile will allow only write operations to occur against M. The other DB Profile will allow only read & bind operations against R. Oracle Access Manager will use the appropriate profile based on the requested operation.

Upgrade Consideration: This is not a multi-master deployment of directory servers. During the upgrade, the schema should be upgraded only against the master (M), because the replica (R) is read-only.

About Configuration Data Replication

Replicated configuration data can be leveraged by Oracle Access Manager in failover configurations. In this scenario there is one master directory server (named M) containing the configuration data and another Read/Write replica (named R). There is one DB Profile configured with primary instance (M) and secondary instance (R).

Note: Oracle Access Manager does not allow load-balancing for configuration data. The same is true for policy data replication.

Upgrade Consideration: This is a multi-master deployment. Schema and data upgrades should be done only against one instance (M). This applies to both Identity System and Access System (policy) data.

Configuring the Challenge/Response Phrase at the Object Class Level

If Challenge and Response attributes are configured at the Employees tab level (rather than at the object class level), then the configuration data upgrade may not complete correctly. Oracle recommends that before starting the upgrade you ensure that the Challenge and Response attributes are configured at the object class level.

To configure the challenge/response phrase as the object class level

1. Login into COREid System Console, as usual.
2. Navigate to the Common Configuration tab, then click Object Classes in the left pane.
3. If attributes P and Q are configured as Challenge and Response attributes, then recollect the object class to which P and Q belongs.
4. On the Configure Object Classes page, click the object class to which the P and Q attributes belong.
5. On the View Object Class page, click the Modify Attributes button.
6. In the attribute configuration applet, ensure that when attribute P is selected in the Attributes list, the Challenge semantic type is highlighted in the Semantic Types list.
7. If the Challenge semantic type is not highlighted, select the Challenge semantic type and save this.
8. Repeat with attribute R for the Response semantic type.

For more information, see your earlier version of the Oblix NetPoint or Oracle COREid *Administration Guide* (Volume 1 if you have a two volume set).

Configuring Unique Namespaces for Directory Connection Information

Each directory server profile contains connection information for a directory that includes the profile name, a domain or namespace to which it applies, a directory type, and a set of operational requirements for Read, Write, Search, and so on. A default directory server profile is created automatically each time you install the Identity Server and specify new directory server connection information.

Before release 6.5, the directory namespaces for policy data and user data had to be unique when the data was stored in two separate directories. During the upgrade to 10*g* (10.1.4.0.1), you must confirm this uniqueness to ensure that multi-language capability can be enabled.

When your environment includes one of the following situations, you need to complete the following procedure before upgrading to ensure that namespaces are unique and do not overlap with other directory server profile namespaces:

- Earlier Oracle Access Manager installation with configuration data or policy data stored in a different directory server than user data.

  Note: Exceptions to overlapping namespaces include a directory server profile for a Microsoft Active Directory subdomain, and the directory server profile containing the configuration DN.

- If the namespace for the configuration DN or policy base assigned during Identity System setup matches the searchbase *and* you upgrade to 10*g* (10.1.4.0.1) without ensuring unique configuration and policy data namespaces, the automated process to enable multi-language capability during the master Identity Server upgrade may fail.

To ensure namespace uniqueness and reconfigure if needed

1. On the directory server, ensure that the namespace for configuration data is unique and does not overlap any other namespace. For details, see the documentation for your directory server.
2. On the directory server, ensure that the namespace for policy data is unique on the directory server and does not overlap any other namespace. For details, see the documentation for your directory server.
3. Using the COREid System Console, configure LDAP Directory Server profiles to include unique namespaces for configuration data and policy data. For example:

   COREid System Console, System Admin, System Configuration

   Configure Directory Options, *Profile_Link*

   Namespace: Enter a unique namespace

   For details about configuring LDAP directory server profiles, see your earlier Oblix NetPoint or Oracle COREid *Administration Guide* (Volume 1 if you have a two volume set).
4. Restart Identity Servers.
5. Re-run Identity System setup, as described in your earlier Oblix NetPoint or Oracle COREid *Administration Guide* (Volume 2 if you have a two volume set).
6. Re-run Policy Manager setup, as described in your earlier Oblix NetPoint or Oracle COREid *Administration Guide* (Volume 2 if you have a two volume set).
7. Re-run Access Server setup, as described in your earlier Oblix NetPoint or Oracle COREid *Administration Guide* (Volume 1 if you have a two volume set).

Preparing Your Directory Instances for the Schema and Data Upgrade

Before starting an upgrade, Oracle recommends that you review the following considerations and perform all tasks outlined here.

Task overview: Preparing directory instances for the schema and data upgrade

1. Review the supported directory servers and releases under the Certify tab at `https://metalink.oracle.com`, then:
   - Log in as directed.
   - Click the Certify tab.
   - Click View Certifications by Product.
   - Select the Application Server option and click Submit.
   - Choose Oracle Application Server and click Submit.
2. **Directory Release Deprecated**: Perform activities in "Preparing a Directory Server when Its Release is Deprecated" on page 5-9.
3. Perform activities in "Changing the Directory Server Search Size Limit Parameter".
4. Review considerations for your directory server and ensure that your environment meets all requirements as described in:
   - Active Directory Considerations and Preparation
   - Active Directory Application Mode Considerations and Preparation

- IBM Directory Server Considerations and Preparation
- Oracle Internet Directory
- Siemens DirX Directory Deprecation
- Sun Directory Server Considerations and Preparation

5. Back up all directory instances containing Oracle Access Manager data, using instructions from your directory vendor.
6. Proceed to "Backing Up Existing Oracle Access Manager Data" on page 5-15

Preparing a Directory Server when Its Release is Deprecated

If your directory server release is no longer supported, you may upgrade earlier directory server profiles in Oracle Access Manager and the directory server as outlined next, then upgrade to 10*g* (10.1.4.0.1).

> **Note:** Use the next sequence as a guide and see your vendor documentation for specific details about administering the directory server. See also "Upgrade Strategies When Support is Changed or Deprecated" on page 2-9.

Task overview: Installing a new directory server when its release is deprecated

1. Check the latest support information for Oracle Access Manager 10*g* (10.1.4.0.1) under the Certify tab on the site:

   ```
   https://metalink.oracle.com
   ```

2. Before starting the Oracle Access Manager upgrade, install a 10*g* (10.1.4.0.1)-supported directory server.
3. In your earlier Oracle Access Manager installation, reconfigure directory server profiles (and database instance profiles contained within) before you start the upgrade to 10*g* (10.1.4.0.1).
4. In your earlier Oracle Access Manager installation, change the directory server as follows:

   From the COREid System Console select System Configuration, Configure Directory Server Options, click Directory Server, change any settings as needed.

5. Re-run Identity System setup, as described in your earlier Oblix NetPoint or Oracle COREid *Administration Guide* (Volume 1 if you have a two volume set), to ensure that configuration files are properly updated.
6. Before starting the upgrade to 10*g* (10.1.4.0.1), complete all activities in this chapter, including the addition of a master Identity Server (formerly known as the COREid Server), WebPass, and Policy Manager (formerly known as the Access Manager component) configured against the new directory.
7. Upgrade to 10*g* (10.1.4.0.1) as described in this manual.

Changing the Directory Server Search Size Limit Parameter

Before starting the upgrade you need to verify that the value of the directory server's search size limit parameter is greater than the number of entries in your configuration

tree. The default value for this parameter varies from directory to directory. See your vendor documentation for complete details.

> **Note:** If the number of entries in your configuration tree is greater than the value of the directory server's size limit parameter, then the Oracle Access Manager data upgrade process may fail.

There are no specific rules to determine a suitable value for this parameter. As a result, the process of defining and verifying the correct value is an iterative one, as described in the next procedure.

To set an appropriate value for the directory server's size limit parameter

1. Check the suitability of the existing value of the directory server's size limit parameter using the `ldapsearch` command to retrieve all nodes in your configuration tree to retrieve all entries. For example:

   ```
   ldapsearch.exe -h  host -p  port-D bindDN -w password -b config_root
   -s sub (objectclass=*) Dn
   ```

   In the preceding command, the *bindDN* is the one that was specified during your earlier Identity System setup (formerly known as COREid).

   - If the result of the `ldapsearch` command is successful, there should be no problem during data migration and you may skip the rest of this procedure.
   - If the `ldapsearch` results in a message about exceeding the size limit, complete step 2.

2. Increment the value of the directory server's size limit parameter using information available in your vendor documentation, then repeat step 1.

   For example, try setting the value to 10000 (or a promising value for your environment) then complete another ldapsearch to see if this is successful. If this also exceeds the size limit, you must repeat step 2 until the ldapsearch command executes successfully.

3. After a successful `ldapsearch`, retain the successful search size limit value until you finish upgrading.

4. After a successful upgrade, you may set the size limit parameter to its original value.

Active Directory Considerations and Preparation

If you have Active Directory as a backend directory server, be sure to review the following information and perform any tasks needed for your environment before upgrading:

- Changing the MaxPageSize Parameter
- Confirming You Are Using a Schema Master

Changing the MaxPageSize Parameter

Before starting the upgrade you need to verify that the value of the search size limit parameter (`MaxPageSize`) is greater than the number of entries in your configuration tree. The `MaxPageSize` parameter specifies the maximum number of entries to return in a search operation. The default value for the `MaxPageSize` parameter is 1000. If the

number of entries in your configuration tree is greater than the value set for the `MaxPageSize` parameter, the Oracle Access Manager data migration process may fail.

> **Note:** The example shown here is based on Active Directory running on Microsoft Windows 2000 Advanced Server. These details may vary for other versions. See your Active Directory documentation for specific details for your version.

To view the existing value of the MaxPageSize parameter and set a new value

1. Use the `ntdsutil` tool at the command prompt to display the current value of the `MaxPageSize` parameter, as shown in the following transcript. For example:

```
C:\Documents and Settings\Administrator ntdsutil
ntdsutil: ldap policies
ldap policy: connections
server connections: connect to server <machine_name>
Binding to <machine_name> ...
Connected to <machine_name> using credentials of locally logged on user
server connections: q
ldap policy: show values
```

2. Use the `ntdsutil` tool at the command prompt to set a new `MaxPageSize` value and view the changes, as shown in the following transcript. For example:

```
C:\Documents and Settings\Administrator ntdsutil
ntdsutil: ldap policies
ldap policy: connections
server connections: connect to server <machine_name>
Binding to <machine_name> ...
Connected to <machine_name> using credentials of locally logged on user
server connections: q
ldap policy: set MaxPageSize to <new_value>
ldap policy: commit changes
ldap policy: show values
```

To choose an appropriate value for this parameter, see "Changing the Directory Server Search Size Limit Parameter" on page 5-9.

Confirming You Are Using a Schema Master

Active Directory, schema modifications may only be completed against a schema master. You may skip this discussion if your earlier environment is configured to use an Active Directory schema master.

If you are not using a schema master, the following procedure may be completed on Windows 2000 platforms. Otherwise, see the Microsoft knowledge base article 285172 "To Enable Schema Updates by Means of the Registry" (previously published under Q285172P) on the Microsoft support Web site.

To enable the schema to be modified

1. Open your Active Directory schema plug-in, which is often located in Administrative Tools.
2. Right-click the top node for the schema and select Operations Master to display the Change Schema Master dialog.
3. Check the box beside "The Schema may be modified on this Domain Controller", then click OK.

Active Directory Application Mode Considerations and Preparation

Before starting the upgrade you must verify that the value of the size limit parameter is greater than the number of entries in your configuration tree. For details, see "Changing the Directory Server Search Size Limit Parameter" on page 5-9.

With ADAM as the directory server, there is no support for obsolete schema cleanup during the upgrade. With ADAM, you must update the schema manually during the upgrade process. However, after manually upgrading the schema, you can accept the automatic data upgrade and complete the component upgrade process.

Support for ADAM started with release 6.5. When upgrading ADAM, Oracle Access Manager provides the following schema files for manual upgrades to configuration and user directories:

IdentityServer_install_dir\identity\oblix\tools\migration_tools\
osd_650_to_700_schema_adam.ldif
user_650_to_700_schema_adam.ldif
policy_650_to_700_schema_adam *(only when user and configuration data are stored separately)*

osd_700_to_1014_schema_adam.ldif
user_700_to_1014_schema_adam.ldif
policy_700_to_1014_schema_adam (*only when user and configuration data are stored separately)*

Each file contains only the specific schema modifications between the named releases. This means:

- If you are upgrading from release 6.5, you must upload the 650_to_700 files during the incremental upgrade from release 6.5 to 7.0. In this case, you must also upload the 700_to_1014 files during the incremental upgrade from release 7.0 to 10*g* (10.1.4.0.1).
- If you are upgrading directly from release 7.0 to 10*g* (10.1.4.0.1), you need only upload the 700_to_1014 files during the incremental upgrade from 7.0.

> **Note:** There are no specific files needed during the upgrade when you are using statically-linked auxiliary classes.

A sample ldifde command to manually update the ADAM schema is shown here and described in Table 5–1. For more information, see your Microsoft documentation:

```
ldifde -k -b
"<user_distinguished_name>""<domain_name>""<user_password>"
-c"<GUID>"<ADAM_instance_ID> -i -f ADAM_oblix_schema_add -s
<ADAM_server_name> -t <port>
```

Table 5–1 Idifde Command Description for ADAM

Option	Description
-k	This option ignores errors.
-b "<*user_distinguished_name*>" "<*domain_name*>" "<*user_password*>" For example: cn=administrator,o=oblix.com,c=us password	To extend the schema, the values represent: ■ *user_distinguished_name*: a Windows security principal user name ■ *domain_name*: domain name of the machine where ADAM is installed ■ *user_password*: password
-c "<GUID>" <*ADAM_instance_ID*>	In this option, "<GUID>" should be retained as is, not replaced by any value; do include the quotes. <*ADAM_instance_ID*> should be substituted by the ADAM root DSE using tools like ldp.exe. When the initial connection is made, the root DSE is shown. For example, an ADAM root DSE value may be EC31B31B-19FC-4FD4-8590-3BD57D6A3E77.
-i -f <filename>	The -i option specifies the import option. The -f option identifies a file name; the value identifies the file you are importing. For example: ADAM_oblix_schema_add.ldif ADAMAuxSchema.ldif
-s <*ADAM_server_name*>	This value is the name of the machine where ADAM is installed.
-t <*port* >	This value is the port number on which this instance listens for the schema update (an open port is needed).

IBM Directory Server Considerations and Preparation

Before starting the upgrade you must verify that the value of the size limit parameter is greater than the number of entries in your configuration tree. For details, see "Changing the Directory Server Search Size Limit Parameter" on page 5-9.

During the first Identity Server and Policy Manager upgrade, the user under whom the IBM SecureWay directory server runs must have read and write access to Oracle Access Manager schema files and to the directory containing the schema files. During the upgrade, you may be prompted to copy the schema files. The upgrade program provides instructions on where to copy them.

The next task overview is provided as a guide in the event that the IBM directory server in your earlier installation is not supported in 10*g* (10.1.4.0.1). Any references to a specific product release is provided for illustration only.

> **Note:** See your vendor documentation for explicit information about administering your directory server. See also "Upgrade Strategies When Support is Changed or Deprecated" on page 2-9.

Task overview: Upgrading with an unsupported IBM Directory Server

1. Check the latest support information for Oracle Access Manager 10*g* (10.1.4.0.1) under the Certify tab on the site:

   `https://metalink.oracle.com`

   - Log in as directed.
   - Click the Certify tab.

- Click View Certifications by Product.
- Select the Application Server option and click Submit.
- Choose Oracle Application Server and click Submit.

2. Before starting the Oracle Access Manager upgrade, you must upgrade your earlier IBM Directory Server (for example, v4.x) data and schema to IBM Directory Server version 5.1 using information available in your IBM documentation.
3. Before starting the upgrade to 10*g* (10.1.4.0.1), complete activities in "Changing the Directory Server Search Size Limit Parameter" on page 5-9.
4. Use 10*g* (10.1.4.0.1) installation packages to upgrade Oracle Access Manager components as described in this guide.

Oracle Internet Directory

Before starting the upgrade you must verify that the value of the `orclsizelimit` parameter is greater than the number of entries in your configuration tree. This specifies the maximum number of entries to return in a search operation. For Oracle Internet Directory the size limit parameter is `orclsizelimit`. The default value for this parameter is 1000. To choose an appropriate value for this parameter, see "Changing the Directory Server Search Size Limit Parameter" on page 5-9.

Note: The example in the following procedure is based on Oracle Internet Directory version 10.1.2 running on Microsoft Windows 2000 Advanced Server. These details may vary for other versions. See your Oracle Internet Directory documentation for specific details for your version.

To view the existing value of the orclsizelimit parameter or set a new value

1. Open Oracle Directory Manager (ODM).
2. In the left navigator pane, expand the Oracle Internet Directory Servers.
3. Select the Directory Server instance to which you have configured your earlier release of Oracle Access Manager.
4. In this instance page (right pane), select the System Operational Attributes tab.

   The "Query Entry Return Limit" parameter on this page refers to the orclsizelimit.

   The value in the text box against this Query Entry Return Limit parameter, shows the existing value for the orclsizelimit parameter.
5. Modify the value in the text box against the Query Entry Return Limit parameter, then apply the changes.

Siemens DirX Directory Deprecation

Oracle Access Manager 10*g* (10.1.4.0.1) does not support Siemens DirX directory. There is no migration path with this directory.

Sun Directory Server Considerations and Preparation

Before starting the upgrade you must verify that the currently directory server release is supported. If not, see "Preparing a Directory Server when Its Release is Deprecated" on page 5-9. Also, you need to ensure that the value of the size limit parameter is

greater than the number of entries in your configuration tree. On a Sun (formerly iPlanet) directory server is `nsslapd-sizelimit` (Size Limit). This specifies the maximum number of entries to return in a search operation. The default value for this parameter is 2000. To choose an appropriate value for this parameter, see "Changing the Directory Server Search Size Limit Parameter" on page 5-9.

> **Note:** The example is based on iPlanet 5.1 running on Microsoft Windows 2000 Professional. These details may vary for other versions. See your Sun directory documentation for specific details for your version.

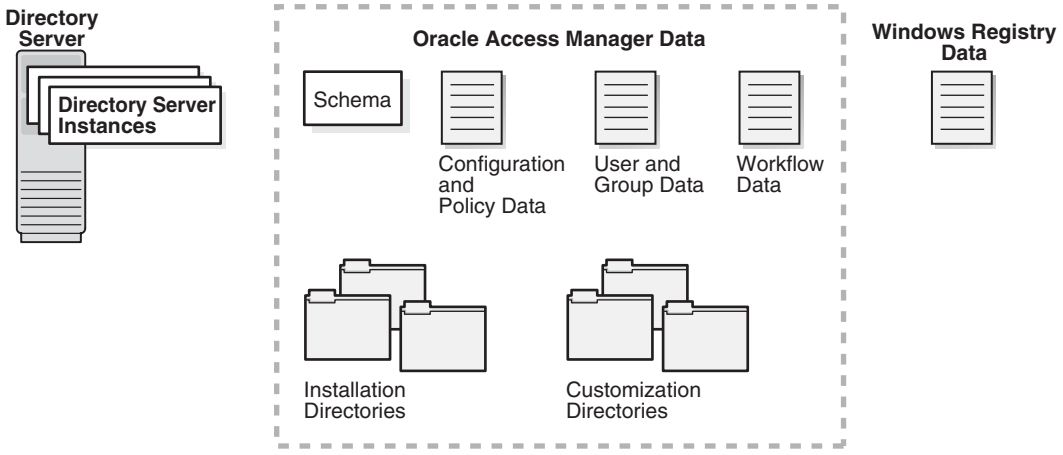
To view the existing value of the nsslapd-sizelimit parameter and set a new value

1. Open the iPlanet console.
2. In Servers and Applications tab, expand the tree in the left pane to show a list of all existing Directory server instances.
3. Select the Directory Server instance to which you have configured COREid.
4. Open the management window of the selected instance.
5. In the opened window, select the Configuration tab.
6. Select the Performance tab to display the existing value of this size limit parameter.
7. Modify the value in the size limit parameter text box, then save the changes.

Backing Up Existing Oracle Access Manager Data

Figure 5–2 shows the types of data to back up before starting the upgrade.

Figure 5–2 Data to Back Up



For more information, see topics in this chapter on:

- Backing up the Earlier Oracle Access Manager Schema
- Backing up Oracle Access Manager Configuration and Policy Data
- Backing Up User and Group Data

- Backing Up Workflow Data
- Archiving Processed Workflow Instances
- Backing Up Existing Directory Instances
- Chapter 8, "Preparing Components for the Upgrade" includes details about:
  - Backing Up the Existing Installed Directory
  - Backing Up the Existing Web Server Configuration File
  - Backing Up Windows Registry Data

Backing up the Earlier Oracle Access Manager Schema

Before starting the upgrade, Oracle recommends that you use tools provided by your directory vendor to backup the schema for your existing directory server instances. For example, Sun ONE directory server stores the schema in the slapd-*instance-name*config/schema/99user.ldif file.

For more information, see your directory vendor documentation.

Backing up Oracle Access Manager Configuration and Policy Data

Before starting the upgrade Oracle recommends that you manually export the Oracle Access Manager configuration and policy data to an ldif file. This file can be used to restore the setup in the unfortunate event of an upgrade failure.

Most vendors provide a directory server console application that can be used to export the directory data into an ldif file. Alternatively you can execute an ldapsearch for the configuration- and policy base at the sub-tree level. In this case, you can use a filter such as (objectclass=*) and re-direct the output of the search to an ldif file.

To back up configuration and policy data

1. Perform the `Ldapsearch` command.
2. In the command, specify the configuration/policy base to back up user and group data.

```
Ldapsearch -h hostname -p port> D bind_dn -w password -s sub
-b  config/policy base dn (objectclass=*) > backup_cp_data.ldif
```

Backing Up User and Group Data

The steps to backup the user and group data are similar to those in "Backing up Oracle Access Manager Configuration and Policy Data" on page 5-16. However, in this case, you specify the searchbase.

For components installed on Windows platform, Oracle recommends that you backup the Windows registry entries for the component in addition to the user and group data.

To back up user and group data

1. Perform the `Ldapsearch` command.
2. In the command, specify the searchbase to back up user and group data.

```
Ldapsearch -h hostname -p port> D bind_dn -w password -s sub
-b searchbase dn (objectclass=*) > backup_ug_data.ldif
```

3. **Windows**: Complete activities in "Backing Up Windows Registry Data" on page 8-9.

Backing Up Workflow Data

Unless you chose to store workflows separately by configuring appropriate DB Profiles, workflow data is automatically backed up as part of the configuration data. When workflows are stored separately, you must perform similar steps to back up workflow data as you did when "Backing up Oracle Access Manager Configuration and Policy Data" on page 5-16.

The only difference when backing up workflow data separately, is that you specify the namespace of the workflow database instance profile in the `Ldapsearch` command, as shown in the procedure here. For example, if the database instance profile is named `workflow_namespace`, that is what you include in the command.

For components installed on Windows platform, Oracle recommends that you backup the Windows registry entries for the component in addition to the user and group data.

To back up workflow data

1. Perform the `Ldapsearch` command.
2. In the command, specify the workflow to back up user and group data.

```
Ldapsearch -h hostname -p port> D bind_dn -w password -s sub
-b workflow_namespace (objectclass=*) > backup_wf_data.ldif
```

3. **Windows**: Complete activities in "Backing Up Windows Registry Data" on page 8-9.

To speed up searching for tickets, you also need to archive processed workflow instances, as described next.

Archiving Processed Workflow Instances

Workflow instances, including those that have been completed and processed by the workflow participants, are stored in the directory server. During the upgrade, workflow data is not disturbed or deleted.

Before starting the upgrade, Oracle recommends that you archive all processed workflow instances. Archived instances are stored in an ldif file in *IdentityServer_install_dir*/identity/oblix/data/common/wfinstance.ldif. This provides a record of the processed workflows and may help speed up the search for workflow tickets.

To archive your processed workflow instances to speed up searching for tickets

1. Navigate to the COREid System Console, as usual.
2. From the User Manager, Group Manager, or Organization Manager application, select Requests.
3. On the Monitor Requests page, fill in the search criteria and search for tickets.
4. If any results are returned, select these and click the Archive button.

   A file is created named wfinstance.ldif and stored in the *IdentityServer_install_dir*/identity/oblix/data/common directory.

Backing Up Existing Directory Instances

To help with a recovery strategy, Oracle recommends that you back up any directory instances containing Oracle Access Manager data before you start the upgrade.

Use instructions from your directory vendor to accomplish this task.

Preparing Host Machines for Master Components

Your next activity is to prepare host machines for the master components you will add and use when upgrading the schema and data. The master components include an earlier Identity Server, WebPass (and Policy Manager (formerly the Access Manager component) if you have the Access System installed).

Details about preparing host machines *before* installing master components, see the following topics in Chapter 8, "Preparing Components for the Upgrade":

- Preparing Host Machines
- Logging in with Appropriate Administrative Rights

After preparing host machines for the master components, proceed to "Adding An Earlier Identity System to Use as a Master". If you also have the Access System installed, you perform tasks in "Adding an Earlier Access Manager to Use as a Master" on page 5-24 after adding the master Identity System. If you do not have the Access System installed, skip Access System-related activities. Whether your installation includes the Access System or not, you will upgrade the Identity System schema and data after completing activities in this chapter.

Adding An Earlier Identity System to Use as a Master

You complete activities here to add one (earlier) Identity Server instance (formerly known as the COREid Server) and WebPass to your existing installation. This additional Identity System will be used as a secondary server for your original master Read/Write directory server instances. Upgrading the schema and data against these master components helps ensure that the schema and data upgrade is successful before you upgrade the rest of your earlier installation.

The master instance that you add here may be installed on any machine you choose that meets 10*g* (10.1.4.0.1) requirements. However, the master instance that you add need *not* be configured for things like auditing and access reporting (even if this is configured for other Identity Servers in your environment). The master instance that you add here has a single purpose and that is to be used during the schema and data upgrade. After upgrading your entire Identity System environment, you may retain this additional instance or remove it without impacting the rest of the upgraded environment.

> **Note:** When your earlier installation includes languages other the English, this additional instance should be installed with the same Language Packs.

Setting up master Identity System for the schema and data upgrade is described next.

Task overview: Adding a master Identity System for the schema and data upgrade includes

1. Defining Additional Instances in the Existing System Console

2. Installing the Master COREid Server Instance
3. Installing the Master WebPass
4. Setting Up the Master Identity System for the Schema and Data Upgrade

Before you begin, confirm that you have completed tasks in Table 5–2. Failure to complete prerequisites may adversely affect your upgrade.

Table 5–2 Master Identity Server Installation Prerequisites Checklist

Checklist	Master Identity Server Installation Prerequisites
	Perform all preparation activities in this chapter. ▪ If you have a multi-language environment, move 10*g* (10.1.4.0.1) Identity System Language Packs for currently installed languages into the same directory as the earlier COREid Server installer that you will use here. ▪ Check host compatibility in your earlier version of the Oblix NetPoint or Oracle COREid *Installation Guide* and complete any installation prerequisites needed for this COREid Server instance.
	Review information in Part I, "Introduction".

Defining Additional Instances in the Existing System Console

The Identity Server instance that you will add requires a WebPass. Before you can install either, however, you must define details for the additional instances in the existing COREid System Console. For additional details, see your earlier Oblix NetPoint or Oracle COREid Administration Guide (Volume 1 if you have a two volume set).

Note: For clarity, this discussion uses earlier terminology that you will see onscreen.

To add information for additional components in the System Console

1. Add information about the new WebPass instance in the COREid System Console. For example:
   - Navigate to the existing COREid System Console, as usual. For example:

     `http://hostname:port/identity/oblix/`

     where *hostname* refers to machine that hosts the existing WebPass Web server; *port* refers to the HTTP port number of the existing WebPass Web server instance; and \identity\oblix connects to the COREid System Console.
   - In the COREid System Console, select System Configuration, then select Configure WebPass and click the Add button.
   - On the Add a new WebPass page, fill in the following information:
     - **Name**: A unique identifier for this WebPass instance (it may include a release number and port number). For example: WebPass_611_6047.
     - **Hostname**: The name full DNS name of the machine hosting this WebPass instance. You may install this instance anywhere; there are no caveats.
     - **Port**: The port number on which this WebPass instance will listen.

- **Maximum Connections**: The maximum number of connections this WebPass opens to COREid Servers. Set this value to 1 for the COREid Server you will add.
- **Transport Security**: Select the security method used for communications between the Identity Server and its Web clients.

Note: Transport security between all Identity System components (Identity Servers and WebPass instances) must match: either all open, all Simple mode, or all Cert.

- **Maximum Session Time (Hours)**: 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.
- **Failover Threshold**: The minimum number of connections to Primary COREid Servers.
- **CoreID Server Timeout Threshold**: The period (in seconds) that the WebPass attempts to contact a non-responsive COREid Server before WebPass considers the server unreachable and attempts to contact another. If a value is not specified, it indicates that there is no timeout.
- **Sleep For (seconds)**: The interval at which WebPass checks its connection with the COREid Server.
- Save the information.

2. Add details for the additional COREid Server instance in the COREid System Console. For example:
   - In the COREid System Console, select System Configuration, then select Configure COREid Servers and click the Add button.
   - On the Add a new COREid Server page, fill in the following information:
     - **Name**: A unique identifier for this COREid Server instance (it may include a release number and port number). For example: *`COREidServer_611_6047`*.
     - **Hostname**: The name full DNS name of the machine hosting this COREid Server instance. You may install this instance anywhere; there are no caveats.
     - **Port**: The port number on which this instance will communicate with its Web clients (WebPass).
     - **Transport Security**: Select the security method used for communications between the COREid Server and WebPass.

Note: Transport security between all Identity System components (Identity Servers and WebPass instances) must match: either all open, all Simple mode, or all Cert.

     - **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.
  - Save the information.

3. In the System Console, associate this COREid Server with the WebPass and specify the priority as Secondary. For example:
   - From the COREid System Console, select System Configuration, then click Configure WebPass.
   - In the List all WebPasses page, click the link for the WebPass you just defined.
   - In the Details of WebPass page, click List COREid Servers.
   - In the page listing Primary and Secondary servers associated with this WebPass, click Add.
   - From the Select Server list (on the Add a new COREid Server to the WebPass page), click select the server you added a moment ago.
   - Indicate that this is a Secondary server.
   - In the Number of connections box, specify the maximum number of connections the WebPass instance opens to this COREid Server (the minimum is 1).
   - Click Add to associate this COREid Server with the WebPass.
4. Proceed to "Installing the Master COREid Server Instance", next.

Installing the Master COREid Server Instance

After defining the new instance in the System Console, you must perform this procedure using the earlier COREid Server installer.

During this installation, you must install this instance on the host you specified in the System Console. Also, you must indicate that this is *not* the first COREid Server for this directory server.

> **Caution:** During this component installation, do *not* update the schema or data. For clarity, this discussion uses earlier terminology that you will see on-screen.

This procedure provides abbreviated steps to complete this task. For more information, see your earlier Oblix NetPoint or Oracle COREid *Installation Guide*.

To install an earlier identity Server for the schema and data upgrade

1. Move earlier installed Identity System Language Packs into the same directory as the earlier COREid Server installer.
2. Log in as a user with administrator privileges to modify product configuration files, then launch the earlier COREid Server installer, as usual. For example:
   - **GUI Method**, Windows:

     NetPoint6_1_1_Win32_COREid_Server.exe
   - **Console Method**, Solaris:

     ./ NetPoint6_1_1_sparc-s2_COREid_Server

     The Welcome screen appears.

3. Specify a new installation directory for this component.
4. **Languages**: Be sure to include and specify all languages that are currently installed in your existing environment.
5. Choose the same transport security mode for this COREid Server that was specified in the System Console.
6. Specify configuration parameters for this instance based on the information you added to the COREid System Console. For example:
   - **Name**: Enter the unique name for this Identity Server. For example: *COREidServer_611_6047*.
   - **Hostname**: Enter the DNS hostname of the machine where you are installing this instance, as specified in the System Console.
   - **Port**: Enter the port number on which this COREid Server communicates with its clients, as specified in the System Console.
7. Specify directory server details for this instance (to ensure that it is installed as a secondary server for your original master Read/Write directory server instances). For example:
   - Select No when asked if this is the first COREid Server to be installed for the directory server.
   - Check the box beside the appropriate communication option (whether SSL-enabled or not) between this COREid Server and the directory server.
   - Complete the transport security dialog according to the mode you chose earlier.
   - Select the option that describes your environment. For example, Configuration data will be in the user data directory or whatever is appropriate for your environment.
   - Select No when asked if you want to update the schema.

   Note: Do not update the schema or data during this installation.
8. Finish the installation as usual.
9. Start the COREid Server service to confirm that the instance is installed and operating properly.
10. Proceed to "Installing the Master WebPass", next.

Installing the Master WebPass

After installing the master COREid Server instance, you now need to install a master WebPass as you defined it in the System Console.

To install the master WebPass

1. Move earlier installed Identity System Language Packs into the same directory as the earlier WebPass installer, if applicable.
2. Log in as a user with administrator privileges to modify the product and Web Server configuration files, then launch the earlier WebPass installer.
   - **GUI Method, Windows**:

NetPoint6_1_1_Win32_*API*_WebPass

- **Console Method, Solaris**:

  ./ NetPoint6_1_1_sparc-s2_*API*_WebPass

  The Welcome screen appears.

3. Dismiss the Welcome screen and respond to the administrator question based upon your platform.
4. Choose the installation directory. For example:

   \OracleAccessManager\Webcomponent
5. **Languages**: If asked, choose a Default Locale to use for the Administrator language and any other Locales to install, then continue.
6. Choose the same transport security mode for the WebPass as you did for the Identity Server.
7. Enter unique information for this WebPass:
   - **Name**: A unique name for this WebPass: *`WebPass_611_ABC`*
   - **Hostname**: DNS hostname of the COREid Server with which this WebPass should communicate: *`Identity_DNS_hostname`*
   - **Port**: Port number of the COREid Server with which this WebPass should communicate: *`Identity_port`*
8. Complete the transport security details based on your earlier specification.
9. Automatically update your Web server configuration file as indicated.
10. Confirm Web server permissions, as needed.
11. Establish communication with the Identity Server as follows:
    - Stop the WebPass Web server instance.
    - Stop then restart Identity Server service.
    - Start the WebPass Web server instance.
12. Proceed to "Setting Up the Master Identity System for the Schema and Data Upgrade" next.

Setting Up the Master Identity System for the Schema and Data Upgrade

After installing the additional COREid Server and WebPass, you must now set these up against your original master Read/Write directory server instances.

To set up the master Identity System for the schema and data upgrade

1. Stop all COREid Server services, if you haven't already.
2. Start the new COREid Server service only.
3. Navigate to the COREid System Console, as usual. For example:
4. Click the Setup button.

> **Note:** You may be prompted to upload the schema to your LDAP server. However, this is not required because this step has already been done.

5. Specify directory information as follows:
   - Specify your existing user data directory server type. For example: Sun.
   - Specify the existing user data directory server details based on your installation. For example:
     - **Host**—The user data directory server DNS hostname
     - **Port Number**—The user data directory server port number
     - **Root DN**—The user data directory server bind DN
     - **Root Password**—Password for the bind DN
     - **Directory Server Security Mode**—Unsecured or SSL-enabled between the user data directory server and Identity Server
     - **Is Configuration data stored in this directory also**?—Yes (default) or No

     Note: If user data is stored separately from configuration data, a similar page appears where you can enter information for the configuration data directory. However, that sequence is not repeated here.

   - On the new page that asks you to specify the location of user and configuration data, enter the configuration bind DN and user data searchbase to be used. For example:
     - Configuration DN—`o=my-company,c=us`
     - Searchbase—`o=my-company,c=us`

     Note: When setting up the instance as a secondary COREid Server, you are *not* prompted for Person or Group objectclass details. Instead, after specifying the location of user data and configuration data, the COREid Setup Complete page appears providing a Done button.

6. On the COREid Setup Complete page, click Done.
7. Create a worksheet for the master Identity Server and WebPass, as described in Appendix E, "Planning Worksheets and Tracking Checklists"
8. Proceed as follows for your environment: If you have an existing Access System in your environment, proceed with
   - **Existing Access System**: Complete activities in "Adding an Earlier Access Manager to Use as a Master", next.
   - **COREid System Only**: Proceed to Chapter 6, "Upgrading Identity System Schema and Data"

Adding an Earlier Access Manager to Use as a Master

This task must be completed only when your existing installation includes the Access System. You to create a master Access Manager (now known as the Policy Manager) as secondary server for your original master Read/Write directory server instances. This new instance will be used later during the Access System schema and data upgrade.

In this specific case, where you plan to use this Access Manager to upgrade the existing Access System schema and data, you do not need to associate and install another Access Server nor a WebGate.

In addition to the procedures in this chapter, you may refer to your earlier version of the Oblix NetPoint or Oracle COREid Access Administration Guide (Volume 2 if you have a two volume set).

Task overview: Adding an earlier Access Manager as a master includes

1. Installing the Master Access Manager for the Schema and Data Upgrade
2. Setting Up the Master Access Manager

> **Note:** If your earlier installation does not include the Access System, you may skip this discussion.

Before you begin, confirm that you have completed tasks in Table 5–3. Failure to complete prerequisites may adversely affect your upgrade.

Table 5–3 Master Access Manager Installation Prerequisites Checklist

Checklist	Master Access Manager Installation Prerequisites
	Perform all preparation activities in this chapter, including "Adding An Earlier Identity System to Use as a Master" on page 5-18, and: ■ If you have a multi-language environment, move 10*g* (10.1.4.0.1) Identity System Language Packs for currently installed languages into the same directory as the earlier WebPass installer that you will use here. ■ Check host compatibility in your earlier version of the Oblix NetPoint or Oracle COREid *Installation Guide* and complete any installation prerequisites needed for this COREid Server instance.
	Review information in Part I, "Introduction".

Installing the Master Access Manager for the Schema and Data Upgrade

After installing and setting up the master Identity System (formerly known as the COREid System), you may install an earlier Access Manager (now known as the Policy Manager) instance to use as a master for the policy data upgrade.

Again, the steps provided here are abbreviated. For more information, see your earlier Oblix NetPoint or Oracle COREid *Installation Guide*.

> **Note:** Do *not* update the schema and data during this installation.

To install an earlier Access Manager for the schema and data upgrade

1. Move earlier installed Access System Language Packs into the same directory as the earlier Access Manager installer, if applicable.
2. Log in as a user with administrator privileges to modify product and Web server configuration files, then launch the earlier Access Manager installer.
   - **GUI Method, Windows**:

     NetPoint6_1_1_Win32_*API*__Access_Manager.exe

   - **Console Method, Solaris:**

./ NetPoint6_1_1_sparc-s2_*API*_Access_Manager

3. Dismiss the Welcome screen, and respond to the question about administrator rights based on your platform.
4. Choose the same installation directory as the WebPass. For example:

   \OracleAccessManager\Webcomponent

5. **Languages**: If asked about languages, choose a Default Locale to use for the Administrator language and any other Locales (languages) to install, then click Next.
6. Respond when asked where policy data is stored and specify directory server details for this instance. For example:
   - Select your directory server type.
   - Respond to the question about where policy data is stored.
   - Select No when asked if you want to update the schema.

     Note: Do not update the schema or data during this installation.

   - On a Solaris system, when policy data is stored with other Oracle Access Manager (formerly known as NetPoint or COREid) data you are asked about the communication method for the existing directory server.
   - On a Windows system, when policy data is stored with other Oracle Access Manager data you are asked about communication with the directory server.
7. Specify the transport security mode this Policy Manager will use to communicate with the rest of the Access System.

   Note: Transport security between all Access System components must match: either all open, all Simple mode, or all Cert.

8. Automatically update your Web server configuration file for this instance and specify the path to your Web server configuration file (then apply changes if you are using a Sun Web server).
9. Stop the Policy Manager Web server instance, stop and restart the Identity Server service, then start the Policy Manager Web server instance.
10. Finish the installation as usual, verify any Web server permissions, then proceed to "Setting Up the Master Access Manager", next.

Setting Up the Master Access Manager

The earlier Access Manager you just added must be set up to communicate with your original master Read/Write directory server instances. The following procedures guide you as you make the connections that are necessary for this communication.

During setup, specifications are saved whenever you click the Next button. If you leave setup and restart it later, you are returned to the same place.

To start setting up the master Access Manager

1. Make sure your Web server is running.

2. Navigate to the Access System Console from your browser by specifying the URL of the WebPass instance that connects to the Policy Manager. For example:

   ```
   http://hostname:port/access/oblix
   ```

   where *hostname* refers to machine that hosts the WebPass Web server; *port* refers to the HTTP port number of the WebPass Web server instance; and \access\oblix connects to the Access System Console.

   You will see the main Access System page.

3. Click the Access System Console link.

   You are informed that the application is not yet set up.

4. Click the Setup button.

   The next page asks about the directory server type.

Specifying Directory Server Details and Data Locations

You need to specify details about the directory servers where user data, configuration data, and policy data are currently stored. You will be asked to provide information about the directory server for each type of data.

To specify directory server details

1. Select your user data directory server type, then click Next.
2. Specify the user data directory server details based on your installation, then click Next. For example:
   - **Machine***:* The user data directory server DNS hostname
   - **Port Number***:* The user data directory server port number
   - **Root DN***:* The user data directory server bind DN
   - **Root Password***:* The password for the bind DN

   Note: For Active Directory, a Domain Name field is included to fill in. With ADSI, a User-Principle-Name field is included where you enter the UserPrincipleName of the Root DN, such as:admin@mycompany.com.

3. Select your configuration data directory server type, then click Next.

   Next you are informed that you can store your user data and configuration data either in the same directory or in separate directories and asked to choose a configuration for your deployment.

4. Choose the item that describes where you user data and configuration data are stored (together or separately), then click Next.
   - If the data is stored together, you are asked where policy data should be stored. In this case, continue with step 5.
   - If the data is stored separately, you are asked to specify details for the configuration data directory server before you continue.
5. Choose the item that describes where your policy data and configuration data are stored (together or separately), then click Next.
   - If the data is stored together, continue with step 6.

- If the data is stored separately, you are asked to specify details for the policy data directory server before you continue.

The Setup Help button appears on the next page, which you can select to obtain additional information during the setup process. You are now asked to specify the location of the configuration DN, searchbase, and policy base.

Note: The configuration DN, searchbase, and policy base may be at the same level or at different levels of the directory tree. However, when the searchbase and the policy base are in separate directories, they must have unique DNs. That is, the searchbase *cannot* be `o=oblix,<Policy Base> or ou=oblix,<Policy Base>` if they are in separate directories. Similarly, the policy base and the configuration DN cannot be same if they are in separate directories.

---

6. Specify the appropriate information for your installation, then click Next. For example:

   - **Searchbase**: *o=my-company,c=us*

     This m*ust* be the same searchbase you specified during Identity System configuration.

   - **Configuration DN**: *o=my-company,c=us*

     This m*ust* be the same configuration DN you specified during Identity System configuration.

   - **Policy Base**: *o=my-company,c=us*

     This node resides within the policy directory server. If this node does not already exist, create it manually.

   You are now asked to specify the Person object class, which must match the one you specified during Identity System setup. For more information, see your preparation worksheets and "To specify Person and Group object class details" on page 6-7.

7. Enter the Person object class name, then click Next.

   For example:

   Person Object Class: *gensiteOrgPerson*

   At this point, you are prompted to restart your Web server.

   Note: If you are using IIS, be sure to follow additional on-screen instructions. Consider using `net stop iisadmin` and `net start w3svc` to stop and start IIS. The net commands help to ensure that the Metabase does not become corrupted.

   ---

8. Stop and restart your WebPass/Access Manager Web server instances and the related COREid Server instance, then click Next to continue.

   Now you are asked to specify the root directory for Oracle Access Manager policy domains.

   Oracle recommends that you accept the default value "/" unless you want to restrict the Master Administrator's ability to define and protect policy domains. For more information, see the *Oracle Access Manager Access Administration Guide.*

9. Accept the default root directory for policy domains (or specify a new root directory), then click Next. For example:

   Policy Domain Root /

   The next page asks about configuring authentication schemes.

Configuring Authentication Schemes

During this Access Manager setup, two authentication schemes are configured automatically. In addition, you can automatically configure a Basic and a Client Certificate authentication scheme based on the configuration information from your user directory.

To configure authentication schemes

1. Define the same authentication schemes for this Access Manager as you have for others.
2. Configure the same policies to protect Oracle Access Manager-related (formerly NetPoint or COREid) URLs.

> **Note:** In this specific case, where you plan to use this Access Manager setup to upgrade the existing Access System schema and data, you do not need to associate and install another Access Server nor a WebGate.

Finishing the Master Access Manager Setup

You finalize setting up the master Access Manager component as follows.

To finalize the master Access Manager setup

1. Complete the set up process as described onscreen.
2. Create a worksheet for the master Access Manager, as described in Appendix E, "Planning Worksheets and Tracking Checklists".

   Proceed to "Finishing Preparation".

Finishing Preparation

The following tasks should be performed on the master instances that you have added to use during the schema and data upgrade. These topics can be found in Chapter 8, "Preparing Components for the Upgrade".

Task overview: Completing preparation for the schema and data upgrade includes

1. Preparing Release 6.x Environments on page 8-4 (if needed)
2. Preparing Multi-Language Installations on page 8-6 (if needed)
3. Backing Up the Existing Installed Directory of master components is described on page 8-8
4. Backing Up the Existing Web Server Configuration File of master Web components is described on page 8-8
5. **Windows**: Backing Up Windows Registry Data on page 8-9 (if needed)
6. Stopping Servers and Services on page 8-9

7. Logging in with Appropriate Administrative Rights on page 8-10
8. When you finish all preparation tasks, you are ready to upgrade the Identity System schema and data as described in Chapter 6, "Upgrading Identity System Schema and Data".

6

Upgrading Identity System Schema and Data

This chapter is intended to be used by directory server administrators who are responsible for updating directory schemas and data. This chapter focuses on upgrading the Oracle Access Manager (formerly known as Oblix NetPoint or Oracle COREid) Identity System schema and data. The following topics are provided:

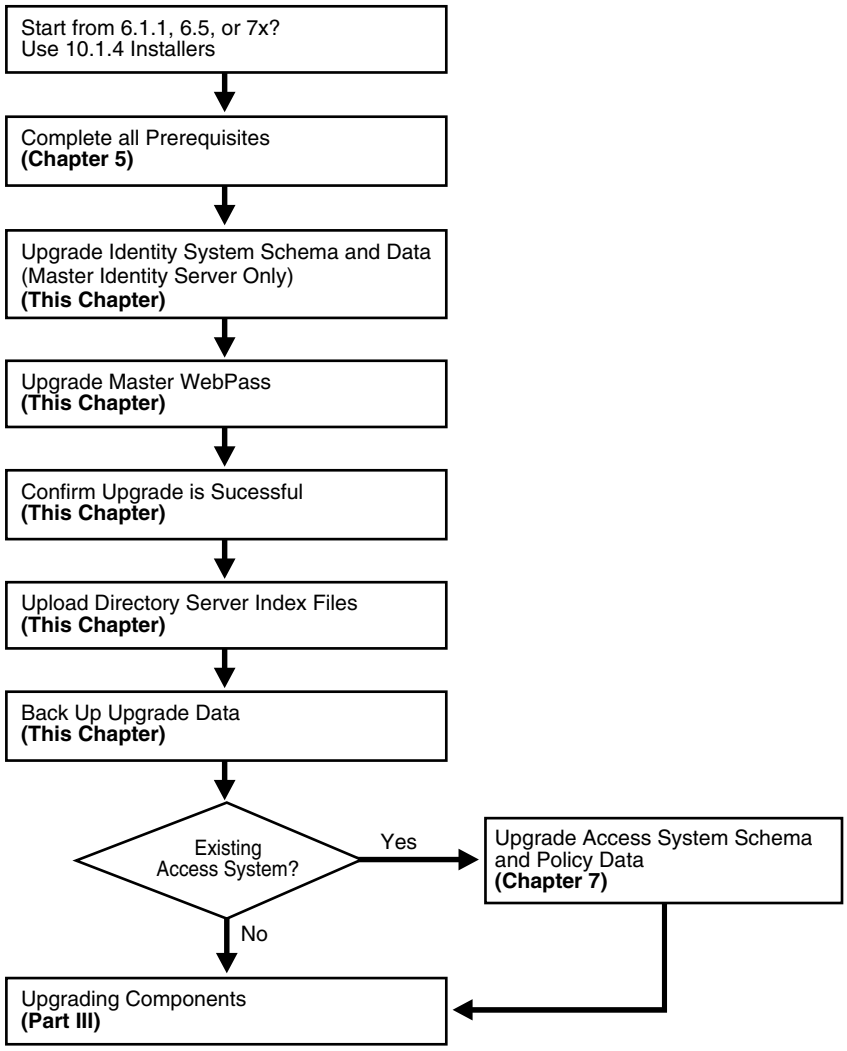
- About Upgrading the Identity System Schema and Data
- Upgrading the Schema and Data with the Master Identity Server
- Upgrading the Master WebPass
- Verifying the Identity System Schema and Data Upgrade
- Uploading Directory Server Index Files
- Backing Up Upgraded Identity Data
- Recovering From an Identity System Schema or Data Upgrade Failure
- Looking Ahead

Note: If your starting Oracle Access Manager release is earlier than 6.1.1, contact Oracle Support before upgrading:
`http://www.oracle.com/support/contact.html`

About Upgrading the Identity System Schema and Data

Figure 6–1 illustrates the tasks you must perform to complete the Identity System schema and data upgrade. Additional notes follow the figure. You will see references to the Identity Server (formerly known as the NetPoint or COREid Server). Refer to your own planning worksheets and use the checklists in Appendix E, "Planning Worksheets and Tracking Checklists" to track your progress.

Figure 6–1 Identity System Schema and Date Upgrade Task



Task overview: Upgrading the Identity System schema and data

1. Complete all prerequisite tasks outlined in "Master Identity System Schema and Data Upgrade Prerequisites" on page 6-4.
2. Upgrade the newly added master Identity Server and accept the automatic schema and data upgrade as explained in "Upgrading the Schema and Data with the Master Identity Server" on page 6-3.

> **Note: Problems During the Upgrade**: See Appendix F, "Troubleshooting the Upgrade Process".

3. Upgrade the master WebPass you added, as discussed in "Upgrading the Master WebPass" on page 6-13
4. Confirm the upgrade was successful, as described in "Verifying the Identity System Schema and Data Upgrade" on page 6-17.
5. **Upgrade Successful**: Perform remaining activities in the following sequence:

- Uploading Directory Server Index Files
- Backing Up Upgraded Identity Data

6. **Upgrade Not Successful**: Proceed to "Recovering From an Identity System Schema or Data Upgrade Failure" on page 6-21.

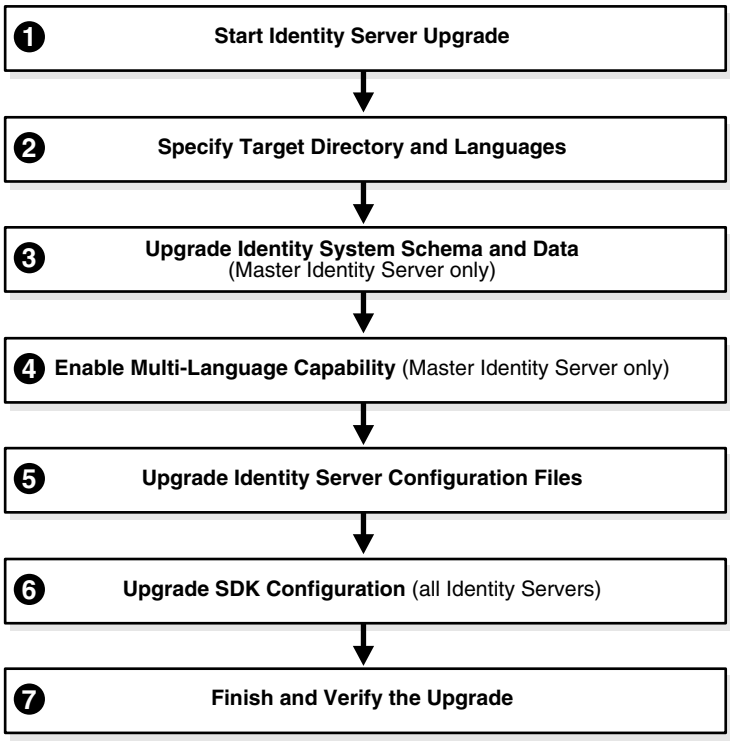
Checklists that can help you track the completion of upgrade tasks in your environment are provided in Appendix E, "Planning Worksheets and Tracking Checklists".

Upgrading the Schema and Data with the Master Identity Server

In this task, you will use the 10*g* (10.1.4.0.1) Identity Server installer to upgrade the earlier COREid Server instance that you added as a master for this purpose. After launching the 10*g* (10.1.4.0.1) Identity Server installer, the sequence of events and questions to which you must respond are directed by the program.

Figure 6–2 illustrates the program-driven process and decision points where you are asked to provide specific responses to continue. Each box in the diagram points to a similarly named discussion that guides you through the procedure. You must complete all procedures for a successful upgrade.

Figure 6–2 Identity System Schema and Data Upgrade Process



Task overview: Upgrading the schema and data with the master Identity Server includes

1. Starting the Master Identity Server Upgrade on page 6-5 describes how to launch the upgrade using your preferred method (GUI or Console).

2. Specifying the Target Directory and Languages on page 6-6 describes how you indicate the directory where the existing component is installed as well as any languages to upgrade.
3. Updating the Identity System Schema and Data on page 6-7 describes how to accept the automatic schema and data upgrade.
   - Oracle recommends that you accept the automatic schema and data upgrade for the master Identity Server.

     Note: With ADAM you must manually upgrade the schema; however, the data upgrade is automatic. For more information, see "Active Directory Application Mode Considerations and Preparation" on page 5-12.

   - Later, when upgrading remaining COREid Servers, the upgraded schema and data are detected automatically; related messages and events are suppressed.
4. Enabling Multi-Language Capability on page 6-8 describes the automated process that occurs only when you upgrade a release 6.1.1 master COREid Server.

   Note: Enabling multi-language capability is automatically skipped if your starting release is 6.5 or 7.x.

5. Upgrading Identity Server Configuration Files on page 6-9 describes how to apply changes to the component-specific configuration files.
6. Upgrading the Software Developer Kit (SDK) Configuration on page 6-12 shows how to accept or decline the SDK configuration changes.

   Note: If this master COREid Server does not have caching configured, you may decline the automatic SDK configuration upgrade.

7. Finishing and Verifying the Master COREid Server Upgrade on page 6-13 describes critical actions that you need to take to determine the success of the upgrade.

Master Identity System Schema and Data Upgrade Prerequisites

Before you begin upgrading the schema and data with the master Identity Server, ensure that you have completed the tasks in Table 6–1. Failure to complete prerequisites may adversely affect your upgrade.

Table 6–1 Schema and Data Upgrade Prerequisites Checklist

Checklist	Schema and Data Upgrade Prerequisites
	Perform all required steps in Chapter 5, "Preparing for Schema and Data Upgrades". ▪ If you have a multi-language environment, see also "Preparing Multi-Language Installations" on page 8-6. ▪ If you are upgrading a release 6.x installation, see also "Preparing Release 6.x Environments" on page 8-4

Table 6–1 (Cont.) Schema and Data Upgrade Prerequisites Checklist

Checklist	Schema and Data Upgrade Prerequisites
	Familiarize yourself with information in Part I, "Introduction"

Starting the Master Identity Server Upgrade

This manual uses the GUI method and Automatic mode to illustrate the sequence of events, sample responses, and messages you may see. In this manual, differences are noted as needed. Even in Automatic mode, you are required to respond to questions about the schema and data upgrade.

If a step does not relate to your environment, you may ignore it. For example, if you have a Windows environment, ignore steps for Unix and vice versa:

- **Windows**: If you are logged in with administrator rights, click Next.
- **Unix**: Specify the username and group that the component will use, then click Next.

The sample upgrade explained in the following procedure starts from a release 6.1.1 installation and includes enabling a multi-language environment.

> **Note:** If errors are reported during the process, check the named log file. and other details in Appendix F, "Troubleshooting the Upgrade Process".

To start the master Identity Server upgrade

1. Ensure that all prerequisites are completed as described in "Master Identity System Schema and Data Upgrade Prerequisites".
2. Turn off the master COREid Server service and log in as a user with the appropriate administrator privileges to update the schema and Oracle Access Manager files.
3. Launch the 10*g* (10.1.4.0.1) Identity Server installer as usual. For example:

   GUI Method, Windows:
   Oracle_Access_Manager10_1_4_0_1_win32_Identity_Server.exe

   Console Method, Solaris:
   ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_Identity_Server

   The Welcome screen appears.
4. Dismiss the Welcome screen by clicking Next.
5. Respond to the administrator question based upon your platform. For example:
   - **Windows**: If you are logged in with administrator rights, click Next (otherwise click Cancel, log in as a user with administrator privileges, then restart the installation).
   - **Unix**: Specify the username and group that the Identity Server will use, then click Next. Typically, the defaults are "nobody."
6. Proceed as described in "Specifying the Target Directory and Languages" next.

Specifying the Target Directory and Languages

In this sequence, you must specify the same target directory for the upgrade as the master COREid Server you just installed. When the earlier component is detected, you are asked if you want to upgrade. When you accept the upgrade, the target directory is created and 10*g* (10.1.4.0.1) files are extracted into it.

After the target directory is created, you are asked to select the languages to upgrade. Unless you have 10*g* (10.1.4.0.1) Language Packs stored in the same directory as the 10*g* (10.1.4.0.1) Identity Server installer, only English is upgraded. After upgrading, you may install languages as described in the *Oracle Access Manager Installation Guide*. You configure Oracle Access Manager to use installed languages as described in the *Oracle Access Manager Identity and Common Administration Guide*.

Unless indicated in the steps in the following procedure, the questions that you see and must respond to are the same regardless of the installation method (GUI versus Console) and mode (Automatic versus Confirmed) that you choose.

To specify the target directory and languages

1. Choose the same installation directory as the master COREid Server you installed and set up, then click Next.
2. Accept the upgrade by clicking Yes, then click Next.
3. If asked, ensure that a check mark appears beside English and any other languages you are upgrading, then click Next.

   You may be presented with a list of languages that will be upgraded.
4. Confirm the languages selected by clicking Next.

   The next screen tells you that the existing installation has been saved and provides the name of the renamed, time-stamped source directory that contains all files from the previous installation.
5. Record the time-stamped directory name and continue the upgrade as instructed.

   A new screen confirms the installation directory for 10*g* (10.1.4.0.1) and tells you how much space is needed for the installation.
6. Start the file extraction into the target directory.

   A status bar indicates the progress of the file extraction.
7. Press Enter to continue.

   ```
   Enter
   ```

   You are asked to specify a mode for the upgrade process: Automatic or Confirmed.

> **Note:** If you are upgrading using the Console method, you are asked to run the command displayed in the transcript. On Unix, the command is printed to a file (start_migration), and a message is printed to run this file.

```
-------------------------------------
Please specify the mode for migration:
'1' - Automatic (recommended)
 Each step is performed automatically.
 No interaction from the user is required.
'2' - Confirmed
```

```
 Each step needs confirmation from the user.
Enter choice ( '1' or '2' ) : 1
-----------------------------------------
```

8. Enter the number that corresponds to the upgrade mode you prefer: For example:
   - **Automatic (recommended)**: Enter the number `1` to observe as the process completes automatically and respond to a few specific questions when needed.
   - **Confirmed**: Enter the number `2` to receive a prompt that you must respond to before each activity.

   The declarative messages in this guide are based on Automatic mode. In this case, you are informed as folders are created, files are copied, and catalogs are upgraded. For example:

```
Creating original folders ...
-----------------------------------------------------
Copying general configuration files
OK.
-----------------------------------------------------
Updating parameter catalogs ...
OK.
-----------------------------------------------------
```

   When the upgrade program connects with the directory server, a transcript appears as shown next.

```
Starting migration (6.1.1 -> 6.5.0)
-----------------------------------------------------
Oracle Access Manager schema migration ...
-----------------------------------------------------
```

9. Regardless of the mode you have chosen, continue with "Updating the Identity System Schema and Data", next.

Updating the Identity System Schema and Data

Oracle recommends that you upgrade the schema and data automatically. Aside from the instructions related to specific directory servers, the upgrade transcript is similar regardless of the *from* and *to* versions or the directory server type.

Unless you are upgrading from Oracle Access Manager 7.0, some portions of the transcript will repeat during the component-specific upgrade sequence. For example if you are upgrading from release 6.1.1, a portion of the dialog will appear once during the upgrade to release 6.5, then repeat during the upgrade to release 7, then repeat again during the upgrade to release 10*g* (10.1.4.0.1).

> **Note:** All schema modifications may be applied to only a master Read/Write directory instance (not against a read-only replica, if any). For more information, see "Strategies for Upgrading in a Replicated Environment" on page 5-4.

The following steps presume that you have chosen Automatic mode. Even so, you will be asked to respond to certain questions.

To upgrade the schema and data

1. Review the information about the schema upgrade and note the *from* and *to* versions. For example:

```
Oracle Access Manager schema migration ...

Retrieving Oracle configuration parameters ...
OK.
   The following directory server's schema will be updated:
      Host:DNShostname.domain.com
      Port: port#
      Type:ns
   NOTE:  If you do not want to migrate schema at this time,
          type 'SKIP'.
   Please type 'Yes" to proceed:
```

Note: You are asked if you want to migrate (upgrade) the schema. Do *not* skip this activity. With ADAM, automatic schema updates are *not* supported. See "Active Directory Application Mode Considerations and Preparation" on page 5-12.

---

2. At the prompt, type the full word "yes" to load the schema.

```
yes
```

   The program updates the schema, while the transcript keeps you informed:

```
Updating schema. Please wait ...
OK.
-------------------------------------------------
```

   During this step, configuration data is also retrieved, parameters are upgraded, and you are informed in the transcript.

Note: You are asked if you want to migrate (upgrade) the data. Do *not* skip this activity.

---

3. At the prompt, type the full word "yes" to load the data.

```
yes
```

   The program converts configuration data, removes older configuration data that is no longer needed, imports new configuration parameters, and so on, while the transcript keeps you informed.

4. Continue as instructed, and proceed to "Enabling Multi-Language Capability" on page 6-8.

Enabling Multi-Language Capability

If your starting release is 6.5 or later, this process does not occur and you may skip this discussion. releases 6.5 and 7.x automatically support a multi-language environment. As a result, when you start upgrading from release 6.5 or 7.x, this event is skipped automatically.

Enabling multi-language capability occurs only during the incremental upgrade of the schema and configuration data from release 6.1.1 to release 6.5. During this phase, the \lang directory structure is included in your upgraded environment and the \en-us subdirectory is provided. Other language subdirectories are included for each additional language that you are upgrading. For more information, see Appendix A, "Oracle Access Manager Directory Structure Changes".

The following sample shows the messages that keep you informed, and actions you need to take, during the multiple-language-enabling sequence. In Automatic mode, the only input required from you is acknowledgment of the events.

To respond during the multi-language sequence

1. Read the messages on language upgrades.

   For example:

```
-------------------------------------
Oracle Access Manager language migration....
Retrieving Oracle configuration parameters...
OK.
Support for multiple languages is not enabled.
Performing language migration...
Updating language migration parameters...
OK.

The following directory server's data will be updated to
support multiple languages:

      Host:DNShostname.domain.com
      Port: port#
         Type:ns

The default language (detected from your existing installation) is: en-us

Press <Enter> to continue
```

2. Press the Enter key to continue:

```
ENTER
```

   The transcript now describes that data is converted for enabling multiple languages and that new language migration data is being imported.

3. At the prompt following successful language migration, press the Enter key to continue:

```
ENTER
```

Upgrading Identity Server Configuration Files

Each component upgrade includes a sequence of events that upgrade the component configuration files. Depending on your starting release, aspects of the sequence may repeat to bring the earlier release up to 10*g* (10.1.4.0.1) incrementally. For example, if your starting release is 6.1.1 the schema and data are upgraded with component configuration data for release 6.5. During the component sequence in the next procedure, the schema and data are upgraded again for release 7.0, then again for 10*g* (10.1.4.0.1).

In Automatic mode, you must type the full word "yes" or press the Enter key when asked to continue the upgrade through each sequence of events.

Note: Your environment may vary. If interim schema and data upgrade messages appear, respond to continue. Do *not* skip any events. However, if interim schema upgrade messages do *not* appear, skip to step 5.

To accept the Identity Server configuration file changes

1. Read the messages regarding the component upgrade. For example:

```
Updating component-specific configuration  ...
OK.
Starting migration ( 6.5.0 -> 7.0.0 )...
-------------------------------------
Oracle Access Manager schema migration....
Retrieving Oracle configuration parameters...
OK.

The following directory server's schema will be updated:
      Host:DNShostname.domain.com
      Port: port#
         Type:ns
NOTE: If you do not want to migrate schema at this time,
     type 'SKIP'.

Please type 'yes' to proceed:
```

2. **If Interim Schema Upgrade Messages Appear**: Type the full word "yes," when asked on the screen, then review the next set of messages and proceed with step 3.

```
yes

Updating schema. Please wait...
OK.
Retrieving User configuration parameters...
OK.
-------------------------------------
Oracle Access Manager data migration....
Retrieving Oracle configuration parameters...
OK.

Could not detect the language for your installation!

Checking product version...
Version not up to date. Performing Configuration data migration...
Updating Oracle migration parameters...

The following directory server's schema will be updated:
      Host:DNShostname.domain.com
      Port: port#
         Type:ns
NOTE: If you do not want to migrate schema at this time,
     type 'SKIP'.

Please type 'yes' to proceed:
```

3. Respond when requested to continue and review messages to track the process. For example:

```
OK.
Converting Configuration data. Please wait...
.........................
 OK.
Removing old Configuration data. Please wait...
.........................
.........................   ............
 OK.

Cleaning up obsolete schema from the directory.
Deleting obsolete schema for osd. Please wait...
Importing new Configuration data. Please wait ...
OK.
-------------------------------------
Oracle Access Manager data migration has completed successfully!
Press <ENTER> to continue :
 Enter

 Updating component-specific configuration files.
```

4. Review messages for the upgrade from release 7.x to 10*g* (10.1.4.0.1). For example:

```
Starting migration ( 7.0.0 -> 10.1.4 )...
-------------------------------------
Oracle Access Manager schema migration....
Retrieving Oracle configuration parameters...
OK.

The following directory server's schema will be updated:
      Host:DNShostname.domain.com
      Port: port#
         Type:ns
NOTE: If you do not want to migrate schema at this time,
     type 'SKIP'.

Please type 'yes' to proceed:
```

5. Continue as directed. For example:

```
  yes
```

The final sequence you see during the upgrade from release 7.x through to 10*g* (10.1.4.0.1) is shown next. Again, you are required to type the full word "yes" or press the Enter key when asked to continue. For example:

```
Updating schema. Please wait...
OK.
Retrieving User configuration parameters...
OK.
-------------------------------------
Oracle Access Manager data migration....
Retrieving Oracle configuration parameters...
OK.
Could not detect the language for your installation!

Checking product version...
Version not up to date. Performing Configuration data migration...
Updating Oracle migration parameters...
```

```
The following directory server's schema will be updated:
      Host:DNShostname.domain.com
      Port: port#
         Type:ns
NOTE: If you do not want to migrate schema at this time,
     type 'SKIP'.

Please type 'yes' to proceed:
  yes
```

6. Review messages as the process continues. For example:

```
OK.
Converting Configuration data. Please wait...
.........................
 OK.
Removing old Configuration data. Please wait...
.........................
.........................
 OK.
Importing new Configuration data. Please wait ...
OK.
-------------------------------------
Oracle Access Manager data migration has completed successfully!
Press <ENTER> to continue :
```

7. Continue when asked and review the final message. For example:

```
Enter

Updating component-specific configuration files...
OK.

Migration has completed successfully!
Press <ENTER> to continue :
-----------------------++++++++++++--------------
```

8. Proceed with "Upgrading the Software Developer Kit (SDK) Configuration" next.

Upgrading the Software Developer Kit (SDK) Configuration

This event may be skipped when you are upgrading a master COREid Server that has *not* been configured to use the SDK. However, if the SDK was configured for this COREid Server, Oracle recommends that you upgrade the configuration now to preserve current settings.

Note: If you do not upgrade current SDK configuration settings, these are *not* preserved and you must reconfigure them later using the configureAccessGate tool. See the *Oracle Access Manager Identity and Common Administration Guide*.

To skip the SDK upgrade for the master COREid Server instance

1. Enter the appropriate number to respond to the question (about migrating the SDK), based on this instance in your environment.

```
This component has the Access Server SDK installed
```

```
Would you like to automatically migrate the SDK at this time?

Note: If you do not want to migrate the SDK at this time, you will
need to reconfigure the SDK after migration has finished
by running the 'configureAccessGate' program
  '1' - Yes
  '2' - No
Enter choice ( '1' or '2' ) :
   2
```

2. Press Enter to continue when asked. For example:

```
 Enter
```

3. Go to "Finishing and Verifying the Master COREid Server Upgrade".

Finishing and Verifying the Master COREid Server Upgrade

You complete this procedure to finish the upgrade of the master COREid Server and the Identity System schema and data.

To finish the master COREid Server upgrade

1. Start the COREid Server service to confirm that it will start (notice that the name has not changed from the one originally assigned).
2. **COREid Server Service Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
3. Check the migration log files for any errors reported during the schema or data upgrade. See "Accessing Log Files" on page F-1.
4. **Upgrade Not Successful**: Proceed to "Recovering From an Identity System Schema or Data Upgrade Failure" on page 6-21.
5. **Upgrade Successful**: Proceed with "Upgrading the Master WebPass" next.

Note: The new product term for COREid Server is Identity Server, which will be used in the remainder of this guide. For more information, see "Product and Component Name Changes" on page -xxi.

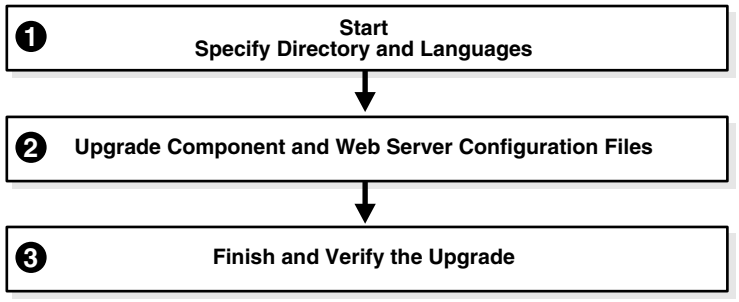
Upgrading the Master WebPass

After upgrading the master Identity Server, you must upgrade the master WebPass instance. There is no WebPass connection to a directory server. Therefore, there is no schema or data upgrade during a WebPass upgrade.

When upgrading WebPass (and other Oracle Access Manager Web components), the component-specific upgrade includes both Web component and Web server configuration file updates. There are no differences between upgrading the master WebPass now and upgrading subsequent WebPass instances later on.

Figure 6–3 illustrates the program-driven WebPass upgrade process and the points at which you need to provide specific input or acknowledge events. Additional comments follow the figure.

Figure 6–3 Master WebPass Upgrade Process



Task overview: Upgrading master WebPass includes

1. Starting the Master WebPass Upgrade, Specifying a Target Directory and Languages is described on page 6-14
2. Upgrading WebPass Configuration Files and Web Server Configurationis described on page 6-15
3. Finishing and Verifying the Master WebPass Upgrade is described on page 6-16

Again, unless you are upgrading from release 7, certain processes repeat automatically for each major release until you reach 10*g* (10.1.4.0.1).

Master WebPass Upgrade Prerequisites

Before you begin upgrading the master WebPass instance, check Table 6–2 to ensure you have completed all tasks. Failure to complete prerequisites may adversely affect your upgrade.

Table 6–2 WebPass Upgrade Prerequisites Checklist

Checklist	WebPass Upgrade Prerequisites
	Upgrade the master Identity Server as described in "Upgrading the Schema and Data with the Master Identity Server" on page 6-3.
	Complete activities in Chapter 8, "Preparing Components for the Upgrade" for this WebPass instance, and: ▪ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6 ▪ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4

Starting the Master WebPass Upgrade, Specifying a Target Directory and Languages

The sample upgrade described here starts from release 6.1.1 using the GUI method and Automatic mode. The sequence of events and messages directed by the program require very little input from you. Much of this is similar to upgrading the master Identity Server. The target directory for the 10*g* (10.1.4.0.1) master WebPass upgrade must be the same as the earlier component.

Note: If errors are reported at any time during the process, check the named log file. See "Accessing Log Files" on page F-1.

To start the master WebPass upgrade

1. Ensure that all prerequisites are completed as described in "Master WebPass Upgrade Prerequisites".
2. Turn off this WebPass Web server instance.
3. Log in as a user with the administrator privileges to update the WebPass and Web server configuration.
4. Locate and launch the 10*g* (10.1.4.0.1) WebPass installer for your Web server. For example:

   GUI Method
   Windows: Oracle_Access_Manager10_1_4_0_1_win32_NSAPI_WebPass.exe

   Console Method
   Solaris: ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_NSAPI_WebPass

   The Welcome screen appears.
5. Dismiss the Welcome screen, then respond when asked about your administrator rights.
6. Choose the directory that contains the earlier WebPass instance.
7. Accept the upgrade when asked.
8. Ensure that a check mark appears beside English and any other languages you want to upgrade, then continue.

   You may be presented with a list of languages that will be upgraded or added.
9. Confirm the languages listed by clicking Next.
10. Record the name of the time-stamped directory, then continue.
11. Start the file extraction.

    A status bar indicates the progress of the file extraction.

    With the GUI method, a new window appears asking you to specify either Automatic or Confirmed mode for the upgrade. Using the Console method, you are asked to run the command displayed in the transcript, then continue as instructed.

Upgrading WebPass Configuration Files and Web Server Configuration

Here you specify the mode to use (Oracle recommends Automatic). For brevity, steps are provided with little explanatory text.

To upgrade the WebPass and Web server configuration

1. Enter the number that corresponds to the mode you prefer and follow the dialog on screen. For example:

```
  1

Creating orig folders ...
-------------------------------------------------
Copying general configuration files ...
OK.
-------------------------------------------------
Updating parameter catalogs ...
OK.
-------------------------------------------------
```

```
Starting migration (6.1.1 -> 6.5.0)
------------------------------------
Updating component-specific configuration files...
OK.
------------------------------------
Starting migration ( 6.5.0 -> 7.0.0 )...
------------------------------------
Updating web server configuration files...
OK.
------------------------------------
Updating component-specific configuration files...
OK.
------------------------------------
Starting migration (7.0.0 -> 10.1.4)
------------------------------------
Updating web server configuration files...
OK.
------------------------------------
Updating component-specific configuration files...
OK.
------------------------------------
Migration has completed successfully!
Press <ENTER> to continue :
```

2. Press the Enter Key.

```
    Enter

If the Access System is also configured, you need to create a DB Profile
manually after first WebPass component upgrade is completed and before
upgrading the first Policy Manager. The profile gives the Access Server write
permission to Policy data in the directory server and will be used while
upgrading the WebGate component. The profile can be deleted after all
the WebGates are successfully upgraded.

Directory permissions copied ...
C:\NetPoint\WebComponent\identity_20060223_180406\oblix)
C:\NetPoint\WebComponent\identity\oblix)
-------------------------------------------------
Migration has completed successfully!
Press <ENTER> to continue.
```

3. Conclude the master WebPass upgrade and proceed to the next discussion, "Finishing and Verifying the Master WebPass Upgrade".

Note: If you have a joint deployment that includes the Access System, you create a temporary directory profile *after* finishing all activities in this chapter and *after* upgrading the Access System schema and data with the master Access Manager component upgrade. Details are provided in the next chapter.

Finishing and Verifying the Master WebPass Upgrade

You finish this master WebPass upgrade as described in the following procedure.

To finish the master WebPass upgrade

1. Apply Web server changes, if needed.

2. Stop, then restart Identity Server service.
3. Start the WebPass Web server instance.
4. **Web Server Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
5. Check the migration log files for any errors reported during the master WebPass upgrade. See "Accessing Log Files" on page F-1.
6. **Upgrade Successful**: Proceed with "Verifying the Identity System Schema and Data Upgrade" next.
7. **Upgrade Not Successful**: Proceed to "Recovering From an Identity System Schema or Data Upgrade Failure" on page 6-21.

Verifying the Identity System Schema and Data Upgrade

You complete this task to confirm that the Identity System upgrade has been successful.

To verify the schema and data upgrade

1. Check to ensure that the schema contains 10*g* (10.1.4.0.1) attributes `obPolicyEnabled` and objectclass `oblixLPMPolicy`.
2. View the configuration node in the configuration directory server and confirm that the value of the `obver` attribute is `10.1.4.0.1`.
3. **Upgrade Successful**: Perform the tasks in the following list, as described in:
   - Uploading Directory Server Index Files
   - Backing Up Upgraded Identity Data
   - Looking Ahead
4. **Upgrade Not Successful**: Proceed to "Recovering From an Identity System Schema or Data Upgrade Failure" on page 6-21.

Uploading Directory Server Index Files

During the master Identity Server (and master Access Manager if you have the Access System installed) upgrade, schema files that include only changes from one release to the next are used to upgrade the existing schema. As a result, the schema and data upgrade repeats for each major product release (for example, from release 6.1.1 to release 6.5, from release 6.5 to release 7.x, and from release 7.x to release 10*g* (10.1.4.0.1)).

For many directory servers, the indexes are automatically updated during the schema and data upgrade. However, when your Oracle Access Manager deployment includes Sun (formerly iPlanet) directory, Novell eDirectory (NDS), and Oracle Internet Directory, you must manually update the directory index files after upgrading the master Identity Server (and master Policy Manager), you need to update the directory index files. The files you use to perform this task are stored in:

IdentityServer_install_dir/identity/oblix/data/common

PolicyManager_install_dir/access/oblix/common

Two index files are provided for each directory server upgrade: the Sun (formerly iPlanet) directory, Novell eDirectory (NDS), and Oracle Internet Directory. One file contains the complete set of 10*g* (10.1.4.0.1) attributes for the user data index and the

other contains the complete set of 10*g* (10.1.4.0.1) attributes for the Oracle Access Manager configuration and policy data index, respectively:

- iPlanet5_user_index_add.ldif *and* iPlanet5_oblix_index_add.ldif
- NDS_user_index_add.ldif *and* NDS_oblix_index_add.ldif
- OID_user_index_add.ldif *and* OID_oblix_schema_index_add.ldif

If policy data is stored on the same directory instance as user data, the _oblix_index_ add.ldif is added once after first Identity Server upgrade only. However, if the policy data is stored on a different directory instance, you must upload the oblix_index_ add.ldif file after both the first (Master) Identity Server upgrade and after the first (Master) Policy Manager upgrade.

Note: In addition to uploading the index files mentioned earlier for your specific directory server, you will also need to manually add an index for the `obpolicykeyword` attribute after the master Policy Manager (formerly known as the Access Manager component) upgrade (if you have Access System configured). The `obpolicykeyword` attribute is currently missing from all 10*g* (10.1.4.0.1) index ldif files and cannot be added automatically during the master Policy Manager upgrade.

Table 6–3 provides a list of the specific attributes to which indexes may need to be manually applied after schema and data upgrades. These apply to all directory servers *except* Oracle Internet Directory. As mentioned earlier, for most directories the indexes are automatically updated during the schema and data upgrade. However, for Sun, Novell eDirectory, and Oracle Internet Directory you must manually upload directory index files, as described in this chapter.

Table 6–3 Indexed Attributes for All Directories Except Oracle Internet Directory

Specific Attributes
obactionname
obactordn
obapp
obattr
obclass
obdatecreated
obdateprocessed
obdirectreports
obentrycondition
obgroupadministrator
obgroupcreator
obgroupdynamicfilter
obgroupexpandeddynamic
obgrouppuredynamic
obgroupsubscribemessage
obgroupsubscribenotification

Table 6–3 (Cont.) Indexed Attributes for All Directories Except Oracle Internet Directory

Specific Attributes
obgroupsubscriptionfilter
obgroupsubscriptiontype
obgroupunsubscribemessage
obid
obindirectmanager
oblocationdn
oblocationname
oblocationtitle
oblockedby
obname
obobjectclass
obpaneltype
obparentlocationdn
obparentstep
obparentworkflow
obparticipant
obpasswordpolicyid
obpolicyconditiongroupStr
obpolicyconditionuidStr
obready
obrectangle
obresourceattribute
obresourceoperation
obresourcetype
obresourceuidStr
obretrycount
obtargetdn
obuniquememberStr
obuseraccountcontrol
obwfinstanceid
obwfstatus
obwfstepid
obwfstepinstid
obwftypename
obworkflowname
obworkflowtype
obwftargetlabel

Table 6–3 (Cont.) Indexed Attributes for All Directories Except Oracle Internet Directory

Specific Attributes
obworkflowdn
obworkflowstepdn
obisworkflowprovisioned
obdynamicparticipantsset
obLPMName
oburlprefix
obSiteDomainID
obHostContext
obdescription
obpolicyKeyword

The steps you must complete to update the indexes depend on your directory server type, as described in:

- Verifying and Uploading Oracle Internet Directory and Sun Directory Indexes
- Verifying and Uploading Novell eDirectory Indexes

> **Note:** If you do not upload the indexes for iPlanet and NDS directories, the product will work. However, searching will be inefficient and impact performance. If you do not upload the indexes for Oracle Internet Directory, users will not be able to login.

Verifying and Uploading Oracle Internet Directory and Sun Directory Indexes

The goal here is to obtain the newly introduced indexes associated with 10*g* (10.1.4.0.1) attributes and ignore any earlier indexes that may be present. If you see errors this is because your environment may already include existing indexes that belong to an earlier release. You can use the Continuous Mode option to continue adding new 10*g* (10.1.4.0.1) attributes in the event that one or more attributes were found to be indexed in an earlier release

To upload the Sun (formerly iPlanet) or Oracle Internet Directory index files

1. **Oracle Internet Directory**: Manually execute directory-specific commands or the directory Administrator Interface to confirm that the indexes have been added using information in the *Oracle Access Manager Schema Description* as a guide.
2. **Sun (formerly iPlanet)**: Manually execute directory-specific commands or the directory Administrator Interface to confirm that the indexes have been added using Table 6–3 as a guide.
3. Locate the appropriate files for your directory server. For example:

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

4. Run the `ldapmodify` command (or use any import tool provided by your directory vendor) and use the Continuous Mode option to avoid any errors that may result when an earlier indexed attribute is found. For example

```
\IdentityServer_install_dir\identity\oblix\tools\ldap_tools\ldapmodify

run ldapmodify.exe -h DS_hostname -p DS_port_number -D bind_dn -w password
-f OID_user_index_add.ldif -a -e reject_filename -c
```

5. Repeat step 2 using the *directory*_oblix_schema_index_add.ldif for your specific directory.
6. When finished, proceed to "Backing Up Upgraded Identity Data" on page 6-21.

Verifying and Uploading Novell eDirectory Indexes

When you have Novell eDirectory, the goal is to obtain the newly introduced indexes associated with 10*g* (10.1.4.0.1) attributes and ignore any earlier indexes that may be present. However, in this case, you *cannot* use the Continuous Mode option.

To confirm or update indexes for Novell eDirectory

1. Manually execute NDS-specific commands or the NDS Administrator Interface to confirm that the indexes have been added using Table 6–3 as a guide.
2. If needed, manually execute NDS-specific commands or the NDS Administrator Interface to upload the indexes.
3. Manually index the `obpolicykeyword` attribute using the appropriate NDS-specific commands or the NDS Administrator Interface.
4. When finished, proceed to "Backing Up Upgraded Identity Data".

Backing Up Upgraded Identity Data

After you finish the schema and data upgrade, Oracle recommends that you back up the 10*g* (10.1.4.0.1) component directory and directory server instances. This will enable you to easily restore your environment to the newly upgraded state should that be a requirement.

To back up critical information after the upgrade

1. Back up the 10*g* (10.1.4.0.1) Identity Server directory and store it in a new location.
2. Back up upgraded directory server instances using your directory vendor documentation as a guide.
3. Back up Identity System data as described in"Backing Up Existing Oracle Access Manager Data" on page 5-15.
4. **Windows**: Back up the upgraded registry for the component as described in "Backing Up Windows Registry Data".
5. **WebPass Web Server**: Back up the upgraded Web server configuration file using your vendor documentation as a guide.
6. Proceed to "Looking Ahead" on page 6-22.

Recovering From an Identity System Schema or Data Upgrade Failure

If the schema and data upgrade was not successful, you may perform the following steps to rollback this upgrade, then try again.

Note: Step 4 is only for WebPass. If only your master WebPass upgrade failed, skip step 1 in the procedure here and complete step 4.

To recover from an unsuccessful schema and data upgrade

1. Restore the directory instance that you backed up before the upgrade to recover the earlier schema and data from backup.
2. Restore the earlier component installation directory that you backed up before the upgrade (to recover the earlier environment), then back it up again. You will retain one of the earlier directories as a backup copy and use one to restart the upgrade
3. **Windows**: Restore the backed up registry for the component.
4. **WebPass Web Server**: Restore the backed up Web server configuration file.
5. Using a backup copy of your earlier information and component installation directory, restart the upgrade, as described in "Upgrading the Schema and Data with the Master Identity Server" on page 6-3.

Looking Ahead

Upgraded Identity System components send and receive information sent in UTF-8 encoding. Earlier components send and receive data in Latin-1 encoding. As a result, the 10*g* (10.1.4.0.1) Identity System does *not* work with an earlier Access System.

When all earlier Identity System components are successfully upgraded, proceed as appropriate for your earlier installation. For example:

- **Identity System Only**: When your earlier installation does *not* include the Access System, you may complete activities in the following sequence using information in:
  - Chapter 11, "Upgrading Integration Components and an Independently Installed SDK"
  - Chapter 12, "Upgrading Your Identity System Customizations" after upgrading all Identity System components.
  - Chapter 14, "Validating the Entire System Upgrade"
- **Joint Identity and Access Systems**: When your earlier installation does include the Access System, you must complete activities in the next sequence using information in:
  - Chapter 10, "Upgrading Access System Components"
  - Chapter 11, "Upgrading Integration Components and an Independently Installed SDK"
  - Chapter 12, "Upgrading Your Identity System Customizations" after upgrading all Identity System components.
  - Chapter 13, "Upgrading Your Access System Customizations"
  - Chapter 14, "Validating the Entire System Upgrade"

For more information about expected system behaviors, see Chapter 4, "System Behavior and Backward Compatibility".

7

Upgrading Access System Schema and Data

If your installation does *not* include Access System components, you may skip this chapter. This chapter is intended to be used by directory server administrators who are responsible for maintaining and updating directory schemas and data.

This chapter explains what you must do and the order in which you must perform the Access System schema and data upgrade. Topics include:

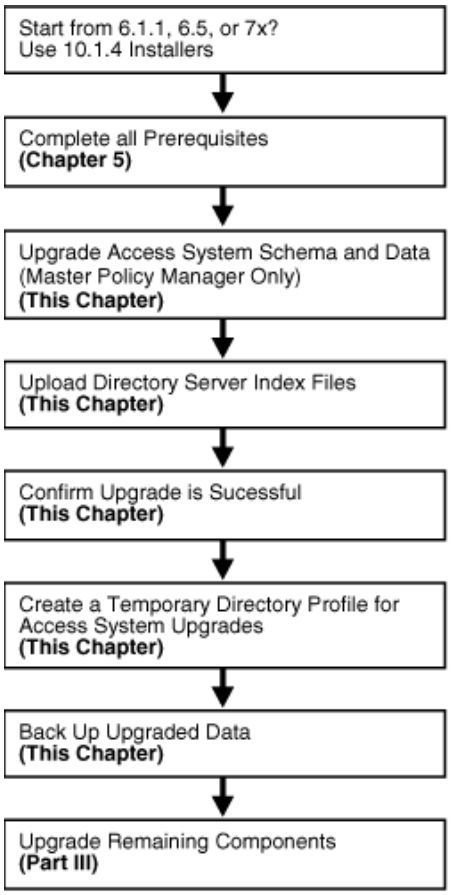
- About Access System Schema and Data Upgrades
- Upgrading the Schema and Data with the Master Access Manager Component
- Uploading Directory Server Index Files
- Verifying the Access Schema and Data Upgrade
- Creating a Temporary Directory Profile For Access System Upgrades
- Backing Up Upgraded Policy Data
- Recovering From an Access System Schema or Data Upgrade Failure
- Looking Ahead

About Access System Schema and Data Upgrades

After upgrading the Identity System schema and data (with the master Identity Server and including a master WebPass upgrade), you are ready to upgrade the Access System schema and data.

Figure 7–1 illustrates the Access System schema and data upgrade tasks. As you can see, in addition to performing and verifying this upgrade you must create a temporary directory profile for later Access System component upgrades. Additional notes follow the figure. Refer to your own planning worksheets and use the checklists in Appendix E, "Planning Worksheets and Tracking Checklists" to track your progress.

Figure 7–1 Access System Schema and Data Upgrade Tasks



Task overview: Upgrading Access System schema and data

1. Complete all prerequisite tasks in Chapter 5, "Preparing for Schema and Data Upgrades".
2. Upgrade the newly added master Access Manager and accept the automatic schema and data upgrade, as explained in "Upgrading the Schema and Data with the Master Access Manager Component" on page 7-3.

> **Note: Problems During the Upgrade**: See Appendix F, "Troubleshooting the Upgrade Process".

3. **Upgrade Successful**: Perform the activities in the following list, in sequence:
   - Uploading Directory Server Index Files
   - Verifying the Access Schema and Data Upgrade
   - Creating a Temporary Directory Profile For Access System Upgrades that grants the Access Server write access to policy data and ensures that information in the WebGatestatic.lst file is written to the directory server appropriately.
   - Backing Up Upgraded Policy Data

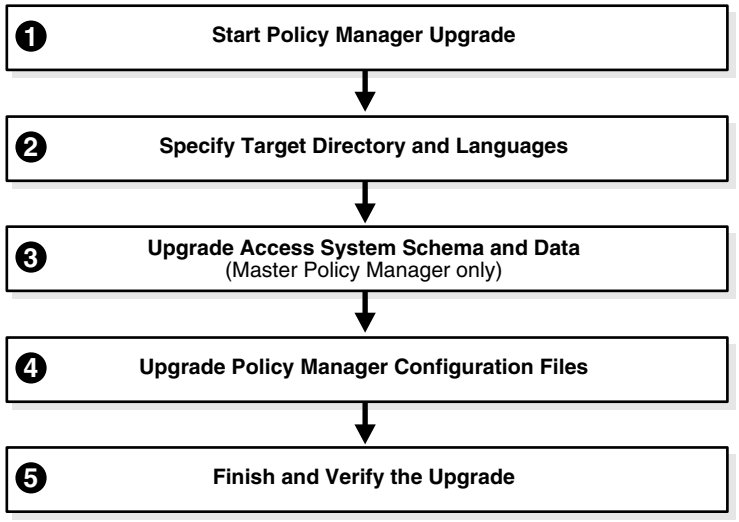
4. **Upgrade Not Successful**: Proceed to "Recovering From an Access System Schema or Data Upgrade Failure" on page 7-13.

Upgrading the Schema and Data with the Master Access Manager Component

During this task, you upgrade the master Access Manager component (now known as the Policy Manager) instance that you added for this purpose, and accept the automatic schema and data upgrade.

Figure 7–2 illustrates the program-driven upgrade process and the points at which you must respond during this upgrade.

Figure 7–2 Access System Schema and Policy Data Upgrade Process



Task overview: Upgrading the Access System schema and data includes

1. Starting the Master Access Manager Upgrade using your preferred method is described on page 7-4.
2. Specifying the Target Directory and Languages described on page 7-4 provides similar steps to those you completed for Identity System components.
3. Updating the Access System Schema and Policy Data described on page 7-5 is similar to steps you performed for the Identity System schema and data upgrade.
4. Upgrading the Access Manager and Web Server Configuration Files described on page 7-7 shows how to accept events in this process.
5. Finishing and Verifying the Access System Schema and Data Upgrade provides vital steps to help you validate the success of the upgrade as described on page 7-9.

Access System Schema and Data Upgrade Prerequisites

Before you begin upgrading the master Access Manager, check the tasks in Table 7–1 to ensure you have completed these.

Failure to complete prerequisites may adversely affect your upgrade.

Table 7–1 Access System Schema and Data Upgrade Prerequisites Checklist

Checklist	Access System Schema and Data Upgrade Prerequisites
	Familiarize yourself with information in Part I, "Introduction"
	Perform all required steps in Chapter 5, "Preparing for Schema and Data Upgrades". ■ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6. ■ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4.
	Perform the Identity System schema and data upgrade, and back up upgraded data, as described in Chapter 6, "Upgrading Identity System Schema and Data".

Starting the Master Access Manager Upgrade

Again, you use the 10*g* (10.1.4.0.1) Policy Manager installer for your specific Web server to launch the upgrade. The sample upgrade described here starts from release 6.1.1. The GUI method and recommended Automatic mode are used to illustrate messages you see, responses you give, and the sequence of events. Your starting release and environment may differ.

Note: Should an error occur, the name of the log file that contains information about the error is identified on the screen. For more information, see "Accessing Log Files" on page F-1.

To start the Access System schema and data upgrade (master Access Manager)

1. Confirm that you have completed all prerequisites for this upgrade, as listed in "Access System Schema and Data Upgrade Prerequisites".
2. Log in as a user with the appropriate administrator privileges to upgrade the schema and Oracle Access Manager files.
3. Stop the master Access Manager Web server.
4. Locate and launch the 10*g* (10.1.4.0.1) Policy Manager installer using your preferred method:

   GUI Method, Windows:

   Oracle_Access_Manager10_1_4_0_1_Win32_NSAPI_PolicyManager.exe

   Console Method, Solaris:

   ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_NSAPI_PolicyManager

   The Welcome screen appears.
5. Dismiss the Welcome screen.
6. Respond to the administrator question based upon your platform. For example:

Specifying the Target Directory and Languages

You specify the same target directory as the master Access Manager component. When you accept the upgrade, the target directory is created and 10*g* (10.1.4.0.1) files are

extracted into it. You are then asked to select the languages that you would like to upgrade.

To specify the target Access Manager directory and languages

1. Choose the directory where you installed the instance you added, then click Next.
2. Accept the upgrade by clicking Yes, then click Next
3. Ensure that a check mark appears beside English and any other languages you are upgrading, then click Next.
4. Confirm the languages listed.
5. Record the time-stamped directory name and continue.
6. Note the amount of disk space required, then start the file extraction into the target directory.

   You are asked to specify a mode for the upgrade process: Automatic or Confirmed.

   If you are using Console method, the installation script exits and a transcript appears. Run the command in the transcript then continue with step 9. (On Unix, the command is printed to a file (start_migration), and a message is printed to run this file.)
7. Press the number for your choice., then review messages that appear. For example:

```
     1

Creating orig folders ...
-----------------------------------------------------
Copying general configuration files
OK.
-----------------------------------------------------
Updating parameter catalogs ...
OK.
-----------------------------------------------------
```

8. Proceed with "Updating the Access System Schema and Policy Data" next.

Updating the Access System Schema and Policy Data

Oracle recommends that you accept the automatic update of the schema and data. The Access System schema and data are upgraded as follows:

- If Oracle Access Manager policy data is stored in the same directory as user and configuration data, the schema was updated during the master Identity Server upgrade. In this case, only policy data is updated during the master Access Manager upgrade.
- If Oracle Access Manager policy data is stored separately from user and configuration data, both the schema and policy data are upgraded during the master Access Manager upgrade.

The first update is detected automatically and you are not asked about schema or data updates during remaining Access Manager upgrades.

Starting with release 6.5, the Access System began using directory server profiles and database instance profiles for accessing user data. As a result, during the incremental upgrade from 6.1.1 to 6.5, a message informs you that a directory server profile is created ("DB Profiles created"), as illustrated in the next procedure. If your starting release is 6.5 or later, you won't see this message.

To upgrade the Access System schema and policy data

1. Review the messages and note the directory path when it appears.

```
-------------------------------------
Starting migration 6.1.1 -> 6.5.0 )...
-------------------------------------
Oracle Access Manager schema migration....
  Retrieving Policy configuration parameters...
  OK.
-------------------------------------
Oracle Access Manager data migration....
  Retrieving Policy configuration parameters...
  OK.
Checking Access Policy version
Version not up to date. Performing Access Policy data migration
Updating Access Policy migration parameters..

The following directory server's schema will be updated:
Host:DNShostname.domain.com
Port: port#
Type:ns

NOTE: If you do not want to migrate schema at this time,
type 'SKIP'.
Please type 'yes' to proceed:
```

2. Type the full word "yes" to update policy data, which may also include a schema upgrade. For example:

```
Yes
OK
```

   The transcript continues.

```
-------------------------------------
Converting Access Policy data. Please wait..
.....
OK
Removing old Access Policy data. Please wait ..
........
OK
Importing new Access Policy data. Please wait ...
OK
-----------------------------
Oracle Access Manager data migration has completed successfully.
Press <ENTER> to continue :
```

3. Press the Enter Key when the data upgrade completes and continue with the retrieval of Oracle Access Manager configuration parameters and database profile creation (if your starting release was 7.x you will not see the DB Profiles created message.

```
-------------------------------------
Retrieving Oracle configuration parameters...
DB Profiles created.
-----------------------------
```

4. Continue with "Upgrading the Access Manager and Web Server Configuration Files" next.

Upgrading the Access Manager and Web Server Configuration Files

During this sequence the component-configuration upgrade is completed for the master Access Manager. This includes Web server configuration updates and policy data configuration parameters.

To upgrade the Web Server and Access Manager configuration

1. Review messages and respond appropriately for your environment when asked.

```
-------------------------------------
Updating web server configuration files...
Connecting to server ...Done.
OK.
-------------------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Starting migration ( 6.5.0 -> 7.0.0 )...
-------------------------------------
Oracle Access Manager schema migration....
Retrieving Policy configuration parameters...
OK.
-------------------------------------
Checking Access Policy version ...
Version not up to date. Performing Access Policy data migration ...

Updating Access Policy migration parameters...
The following directory server's schema will be updated:
Host:DNShostname.domain.com
Port: port#
Type:ns
NOTE: If you do not want to migrate schema at this time,
type 'SKIP'.
Please type 'yes' to proceed:
```

2. Continue the upgrade as directed.

```
yes
```

   The process continues, as indicated here.

```
Converting Access Policy data. Please wait...
.....
OK.
Removing old Access Polidy data. Please wait ...
.....
OK.
Cleaning up obsolete schema from the directory.
Deleting Obsolete schema for policy. Please wait.
Importing new Access Policy data. Please wait...
OK.
----------------------------------------
Oracle Access Manager data migration has completed successfully.
Press <ENTER> to continue :
```

3. Respond after the data upgrade and notice that Web server configuration and component-specific upgrades occur next.

```
     Enter
-------------------------------------
```

```
Updating web server configuration files...
OK.
-------------------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Starting migration ( 6.5.0 -> 7.0.0 )...
-------------------------------------
Oracle Access Manager schema migration....
Retrieving Policy configuration parameters...
OK.
-------------------------------------
Oracle Access Manager data migration....
Retrieving Policy configuration parameters...
OK.
-------------------------------------
Checking Access Policy version ...
Version not up to date. Performing Access Policy data migration ...

Updating Access Policy migration parameters...
The following directory server's schema will be updated:
Host:DNShostname.domain.com
Port: port#
Type:ns
NOTE: If you do not want to migrate schema at this time,
type 'SKIP'.
Please type 'yes' to proceed:
```

4. Type the full word "yes" to continue.

```
    yes

OK.
Converting Access Policy data. Please wait...
OK.
Removing old Access Policy data. Please wait ...
OK.
Importing new Access Policy data. Please wait...
OK.
----------------------------------------
Oracle Access Manager data migration has completed successfully.
Press <ENTER> to continue :
```

5. Continue with component-specific configuration for release 7.0 to 10*g* (10.1.4.0.1), if needed:

```
     Enter
Updating component-specific configuration files.
...
Converting Access Policy data. Please wait...
OK.
Removing old Access Policy data. Please wait ...
OK.
Importing new Access Policy data. Please wait...
OK.
----------------------------------------
Oracle Access Manager data migration has completed successfully.
Press <ENTER> to continue :

Directory permissions copied ...
C:\NetPoint\WebComponent\access_20060223_180406\oblix)
```

```
C:\NetPoint\WebComponent\access\oblix)
-------------------------------------------------
Migration has completed successfully!
Press <ENTER> to continue.
```

6. When this phase completes, continue as instructed on the screen and proceed to "Finishing and Verifying the Access System Schema and Data Upgrade".

Finishing and Verifying the Access System Schema and Data Upgrade

You finish the master Access Manager upgrade as described next.

To finish and verify the Access System schema and data upgrade

1. Apply any changes to the Web server configuration file, if needed.
2. Start the upgraded Access Manager Web server to confirm that this upgrade was successful.
3. **Web Server Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
4. View Access Manager migration log files and error ldifs to see if they contain any errors. See "Accessing Log Files" on page F-1.
5. **Upgrade Successful**: Proceed with "Uploading Directory Server Index Files" on page 7-9 to ensure that all attributes are included for the Access System schema and data (and be sure to manually add an index for the `obpolicykeyword` attribute.
6. **Upgrade Not Successful**: Proceed to "Recovering From an Access System Schema or Data Upgrade Failure" on page 7-13.

> **Note:** The new product term for the Access Manager component is Policy Manager, which will be used in the remainder of this guide. For more information, see "Product and Component Name Changes" on page -xxi.

Uploading Directory Server Index Files

This procedure is the same as the one you completed after upgrading the Identity System schema and data.

For Access System data, be sure to manually add an index for the `obpolicykeyword` attribute. For more information, complete appropriate activities for your environment in "Uploading Directory Server Index Files" on page 6-17.

After uploading index files for the Access System, continue with "Verifying the Access Schema and Data Upgrade", next.

Verifying the Access Schema and Data Upgrade

You complete this procedure to validate the Access System schema and data upgrade.

To verify the Access System schema and data upgrade

1. Using your directory administration console, confirm that the schema contains all the object classes and attributes as defined in the *Oracle Access Manager Schema Description*.

2. Using your directory administration console, verify that all the indexes have been added.
3. **Different Directory Server Instances**: Perform the steps in the following list to ensure that the schema was also updated:
   - View the configuration node in the configuration directory server and confirm that the value of the `obver` attribute is `10.1.4.0.1`.
   - Check to ensure that the schema contains 10*g* (10.1.4.0.1) attributes `obPolicyEnabled` and objectclass `oblixLPMPolicy`.
4. **Upgrade Successful**: Proceed as indicated in the next list:
   - Creating a Temporary Directory Profile For Access System Upgrades
   - Backing Up Upgraded Policy Data
   - Looking Ahead
5. **Upgrade Not Successful**: Proceed to "Recovering From an Access System Schema or Data Upgrade Failure" on page 7-13.

Creating a Temporary Directory Profile For Access System Upgrades

After upgrading the master Policy Manager, and before upgrading *any* other Access System component, a Master Access Administrator must create a specific temporary directory server profile using the Identity System Console. This profile grants the Access Server write access to policy data stored in the directory server and updated during the Policy Manager upgrade.

During WebGate upgrades, the Access Server gathers configuration information stored in the WebGatestatic.lst file and updates the directory server using the temporary directory profile created for this purpose. After writing information to the directory server, the Access Server returns status information to the WebGate. Any unknown parameters in the WebGateStatic.lst file are moved to the directory server.

> **Note:** Upgrading any Access System components *before* creating this profile could result in a failed upgrade. The exception to this rule is the master Policy Manager that you upgraded with the Access System schema and data.

In earlier releases, WebGate configuration parameters were stored in the WebGatestatic.lst file. However, in Oracle Access Manager 10*g* (10.1.4.0.1), WebGate configuration is accomplished using the Access System Console. Proper migration of earlier WebGate configuration parameters during an upgrade is required to enable you to change the parameter values, and add new ones, using the Access System Console. After upgrading a WebGate to 10*g* (10.1.4.0.1), you must use the System Console to adjust parameters. You cannot continue to use the WebGatestatic.lst file after upgrading.

Guidelines for the Temporary Directory Profile

When creating this temporary directory profile you must:

- Assign a profile name of migration_wgstatic_profile; do *not* use another name.
- Create the migration_wgstatic_profile for the directory where the policy data is stored. If your user, configuration, and policy data are stored together on a single directory, create this new profile for that directory. However, if your policy data is

stored in the same directory as configuration data, create this new profile for that directory.

- Assign permissions for all operations to the migration_wgstatic_profile.
- Use the same namespace as the policy base stored in *PolicyManager_install_dir*/access/oblix/config/configInfo.xml. The value of the `ldapPolicyBase` parameter should be used: for example, *`obapp=PSC, o=Oblix, o=company,c=us`*.
- If your directory server supports LDAP referrals, enable LDAP referrals in this temporary directory server profile. A referral directs a client request to another server to find requested information in another location. See the *Oracle Access Manager Identity and Common Administration Guide* for details.
- If the policy data directory server is SSL-enabled, the CA certificate is needed by the Access Server to connect to the directory. The CA certificate must be manually added (using the certutil tool) to the certificate store in *AccessServer_install_dir*/access/oblix/config/cert8.db or cert7.db. However, if the existing policy data directory used by the Access Server is already in SSL mode and uses the same CA certificate, this step need not be done.

Important: This procedure must be completed before upgrading any additional Access System components. For more information about directory server profiles, see the *Oracle Access Manager Identity and Common Administration Guide*.

To create the temporary directory server profile for the Access Server

1. Navigate to the Identity System Console (formerly known as the COREid System Console). For example:

   `http://`*`hostname:port`*`/identity/oblix/`

2. From the Identity System Console, click the System Configuration tab.
3. Click Directory Profiles to display the Configure Profiles page.
4. Locate the Configure LDAP Directory Server Profiles section and click Add to display the Create Directory Server Profile page.
5. Fill in the following information for this temporary profile: In the Name field, enter the following name and the namespace for your environment:

   Name: `migration_wgstatic_profile`

   Name Space: *`obapp=PSC`*`,o=Oblix,o=company,c=us`

   where the Name Space is the value of the LDAP PolicyBase parameter in *PolicyManager_install_dir*/oblix/config/configInfo.lst
6. Select the All Operations button to give this profile permission to perform all operations.
7. In the Used By field, select the Access Servers option.

   Next you must create a database instance profile where you identify the directory server where your policy data is stored. If your policy data is stored on a separate directory server, the new database instance profile should be created for that directory server. If user, configuration, and policy data are all on one directory server, the new database instance profile should be created for that directory server

8. In the Database Instances section of the Create Directory Server Profile page, click Add.

   The Create Database Instance page appears.

9. Fill in the following information to configure a database instance profile for your policy data directory server:

   Name:
   Machine:
   Port:
   Root DN:
   Root DN Password:

   For more information, see the *Oracle Access Manager Identity and Common Administration Guide* for details.

10. In the Flags field, if your directory supports LDAP referrals click the LDAP referrals check box if appropriate for your environment.

    See the *Oracle Access Manager Identity and Common Administration Guide* for details on configuring LDAP referrals.

11. Save the database instance profile and the associated directory server profile.

12. If the policy directory server operates in SSL mode, the Access Server requires a CA certificate to connect to it.

    If the policy directory server uses the same CA certificate as the Access Server, no additional configuration is required. Otherwise, you must add the CA certificate (cert8.db or cert7.db) to the certificate store in the following directory:

    AccessServer_install_dir/oblix/config

    Where *AccessServer_install_dir* is the directory where the Access Server was installed. See the appendix on adding a new certificate store in the *Oracle Access Manager Installation Guide* for details.

13. Proceed to Backing Up Upgraded Policy Data next.

Backing Up Upgraded Policy Data

As mentioned earlier, Oracle recommends that you finish the schema and data upgrade by backing up the 10*g* (10.1.4.0.1) component directory and directory server instances. This will enable you to easily restore your environment to the newly upgraded state should that be a requirement.

To back up critical policy information after the upgrade

1. Back up the upgraded 10*g* (10.1.4.0.1) Policy Manager directory and store it in a new location.

2. Back up upgraded directory server instances using your directory vendor documentation as a guide.

3. Backup upgraded policy data, as described in "Backing up Oracle Access Manager Configuration and Policy Data" on page 5-16.

4. Back up the upgraded Web server configuration file as described in your vendor documentation.

5. **Windows**: Back up Windows registry data, if required, as described in "Backing Up Windows Registry Data" on page 8-9.

6. Proceed to "Looking Ahead" on page 7-13.

Recovering From an Access System Schema or Data Upgrade Failure

If the schema and data upgrade was not successful, you may perform the following steps to rollback this upgrade, then try again.

To recover from an unsuccessful Access System schema and data upgrade

1. Restore the directory instance that you backed up before the upgrade to recover the earlier schema and data from backup.
2. Restore the earlier component installation directory that you backed up before the upgrade (to recover the earlier environment), then back it up again. You will retain one of the earlier directories as a backup copy and use one to restart the upgrade.
3. **Policy Manager Web Server**: Restore the earlier Web server configuration file.
4. **Windows**: Restore the backed up registry, if needed.
5. Using a backup copy of your earlier data and component installation directory, restart the upgrade, as described in "Upgrading the Schema and Data with the Master Access Manager Component" on page 7-3.

Looking Ahead

After upgrading the Access System schema and data, proceed in sequence with the following chapters and tasks:

- Chapter 8, "Preparing Components for the Upgrade"
- Chapter 9, "Upgrading Remaining Identity System Components"
- Chapter 10, "Upgrading Access System Components"
- Chapter 11, "Upgrading Integration Components and an Independently Installed SDK"
- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"
- Chapter 14, "Validating the Entire System Upgrade"

For more information about expected system behaviors, see Chapter 4, "System Behavior and Backward Compatibility".

Part III

Upgrading Components

This part of the book describes how to upgrade your earlier Identity and Access System components to 10*g* (10.1.4.0.1) after upgrading the schema and data.

Part III contains the following chapters:

- Chapter 8, "Preparing Components for the Upgrade"
- Chapter 9, "Upgrading Remaining Identity System Components"
- Chapter 10, "Upgrading Access System Components"
- Chapter 11, "Upgrading Integration Components and an Independently Installed SDK"

8

Preparing Components for the Upgrade

This chapter provides important information to help you prepare your earlier environment before you begin upgrading remaining Identity and Access System components. Refer to your own planning worksheets and use the checklists in Appendix E, "Planning Worksheets and Tracking Checklists" to track your progress.

Topics in this chapter include:

- Checking Compatibility with Previous Releases
- Copying Custom Identity Event Plug-ins
- Preparing Earlier Customizations
- Preparing the Default Logout in the Policy Manager
- Preparing Host Machines
- Preparing Release 6.x Environments
- Preparing Multi-Language Installations
- Backing Up Directories, Web Server Configurations, and Registry Details
- Stopping Servers and Services
- Logging in with Appropriate Administrative Rights

> **Note:** New product and component names are used in this chapter even when referring to the earlier product and components. For example, Oracle Access Manager is used instead of Oblix NetPoint or Oracle COREid. For more information, see "What's New in Oracle Access Manager?" on page -xxi.

Checking Compatibility with Previous Releases

There are several actions you need to take to confirm compatibility between your earlier installation and 10*g* (10.1.4.0.1). As mentioned earlier, support for some items has been deprecated. In some cases, you may need to decide how to proceed given the removed support.

To confirm compatibility with 10*g* (10.1.4.0.1)

1. Review "Support Deprecated" on page 2-8
2. Review 10*g* (10.1.4.0.1) platform support under the Certify tab on `https://metalink.oracle.com`.
   - Log in as directed.

- Click the Certify tab.
- Click View Certifications by Product.
- Select the Application Server option and click Submit.
- Choose Oracle Application Server and click Submit.

3. When directory server or Web server versions or platform support has changed, see "Upgrade Strategies When Support is Changed or Deprecated" on page 2-9, for information about how to proceed.

Copying Custom Identity Event Plug-ins

All standard plug-ins are copied during the upgrade, as are custom authorization and authentication plug-ins. However, custom Identity Event plug-ins created using the Identity Event Plug-in API are not copied during the upgrade.

Oracle recommends that you complete the following procedure to prepare custom Identity Event plug-ins for possible redesign before the upgrade, in a test or development environment.

To copy Identity Event Plug-ins before the upgrade

1. Before the upgrade, create a directory for your old Identity Event plug-ins in the top level of your Identity Event API directory.
2. Copy custom Identity Event plug-ins in to the new directory.
3. Proceed to "Preparing Earlier Customizations" next.

Preparing Earlier Customizations

Oracle recommends that you start manually processing customizations in your existing environment well in advance of upgrading components. This should occur in a test or development environment to minimize service disruptions. After completing the upgrade and testing your customizations, you can deploy them in the upgraded environment as discussed in "Customization Upgrade Planning" on page 1-13.

For many customizations, you can start work before upgrading components. However, a few customization upgrades can be accomplished only after upgrading components. The overview here lists Identity System customization work that must be completed and where to find more information within Chapter 12, "Upgrading Your Identity System Customizations".

Task overview: Upgrading earlier Identity System customizations includes

1. Upgrading Auditing and Access Reporting for the Identity System on page 12-2
2. Combining Challenge and Response Attributes on a Panel on page 12-8
3. Confirming Identity System Failover and Load Balancing on page 12-9
4. Migrating Custom Identity Event Plug-Ins on page 12-10
5. Ensuring Compatibility with Earlier Portal Inserts on page 12-11
6. Incorporating Customizations from Release 6.5 and 7.x on page 12-12
7. Incorporating Customizations from Releases Earlier than 6.5 on page 12-13

The next overview outlines Access System customization upgrades you need to perform manually and where to find more information within Chapter 13, "Upgrading Your Access System Customizations".

Task overview: Upgrading earlier Access System customizations includes

1. Upgrading Auditing and Reporting for the Access Server on page 13-2
2. Upgrading Forms-based Authentication on page 13-4
3. Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins on page 13-5
4. Associating Release 6.1.1 Authorization Rules with Access Policies on page 13-5
5. Assuring Proper Authorization Failure Re-directs After Upgrading from 6.1.1 on page 13-6
6. Updating the ObAMMasterAuditRule_getEscapeCharacter in Custom C Code on page 13-7

For more information, see "About Upgrading and Backward Compatibility" on page 4-2.

Preparing the Default Logout in the Policy Manager

Before the upgrade, the default logout should be unprotected in the Policy Manager. Otherwise, after the upgrade, users will be challenged when they click the Logout link.

To prepare the default logout for an upgrade

1. Confirm that the default logout is not protected in the Policy Manager.
2. If you use oblogout.cgi for WebGate logouts, be sure that it is installed on the target server.

Preparing Host Machines

Preparing machines hosting the earlier installation includes the following procedures:

- Changing Read Permissions on Password Files
- Confirming Free Disk Space

For additional information, see "Backing Up Directories, Web Server Configurations, and Registry Details" on page 8-8.

Changing Read Permissions on Password Files

If you are running the Identity System using Simple or Cert mode, your password.xml file in the *IdentityServer_install_dir*\identity\oblix\config directory is not readable. The same issue applies to the Access System password.lst file in *install_dir*\access\oblix\config.

To prepare password files for the upgrade

1. For Identity System upgrades, assign read permissions to password.xml for the duration of the upgrade process. See also, "Logging in with Appropriate Administrative Rights" on page 8-10.
2. Reset password.xml to the desired permissions after the upgrade is complete.

3. For Access System upgrades, repeat the steps in this list on the password.lst file.

Confirming Free Disk Space

You need enough disk space on the machine hosting the earlier component for both the earlier Oracle Access Manager release and the new release.

To confirm you have enough disk space

1. Check the *Oracle Access Manager Installation Guide* for the disk space required for the new component.
2. On the machine hosting the component to be upgraded, check the amount of disk space required for the earlier installation that will be retained in a renamed time-stamped source directory.

Preparing Release 6.x Environments

To ensure success when upgrading certain Oracle Access Manager 6.x installations (excepting 6.5.1), you must add specific bundles in to your original *Component_install_dir* in addition to the standard 10*g* (10.1.4.0.1) installation packages. Discussions here identify additional files needed when you are starting the upgrade from specific releases only:

- Adding Packages for Release 6.1.1 on AIX
- Adding Packages for Release 6.5.0.x
- Adding Packages for Release 6.5.2.x Patch

WARNING: Use only the files that are relevant to your specific installation. See also "Preparing Multi-Language Installations" on page 8-6.

Adding Packages for Release 6.1.1 on AIX

During the upgrade, the installer for each component creates a directory named "orig" and compares the environment to ensure appropriate files are upgraded. However, the original Oracle Access Manager release 6.1.1 installers for WebPass and Policy Manager releases for AIX platforms did not create a file named "orig".

As a result, before you upgrade to 10*g* (10.1.4.0.1), you must extract and add the following packages to your original *Component_install_dir*:

Untar 611_orig Packages to the Original *Component_install_dir*
Netpoint_611_orig_WebPass_param.tar
Netpoint_611_orig_Access_Manager_param.tar

To prepare an Oracle Access Manager 6.1.1 AIX installation

1. Obtain the 611 for AIX packages in the preceding list from the 10*g* (10.1.4.0.1) installation media.
2. On each AIX machine hosting a WebPass or Policy Manager, extract (untar) or unzip files in to the original component installation directory (*Component_install_dir*).

This creates a new directory named "orig" for each component. For example: *Component_install_dir*/identity/oblix/orig (or *Component_install_dir*/access/oblix/orig).

3. Finish preparing your environment as described in this chapter and start the upgrade.

Processing is automatic and no further action is required by you.

Adding Packages for Release 6.5.0.x

During the upgrade to 10*g* (10.1.4.0.1), the installer for each component creates a directory named "orig" and compares the environment to ensure appropriate files are upgraded. The original Oracle Access Manager 6.5.0 release did not provide an upgrade capability and, therefore, did not create a file named "orig".

As a result, before you upgrade from Oracle Access Manager 6.5.0.x, you must extract and add the following packages to your original *Component_install_dir*.

Extract 65-orig Packages to the Original *Component_install_dir*
Netpoint_65_orig_en_COREid_Server_msg.zip
Netpoint_65_orig_COREid_Server_param.zip
Netpoint_65_orig_en_Access_Manager_msg.zip
Netpoint_65_orig_Access_Manager_param.zip
Netpoint_65_orig_en_WebPass_msg.zip
Netpoint_65_orig_WebPass_param.zip
Netpoint_65_orig_en_Access_Server_msg.zip
Netpoint_65_orig_Access_Server_param.zip
Netpoint_65_orig_en_WebGate_msg.zip
Netpoint_65_orig_WebGate_param.zip

To prepare a 6.5 environment for the upgrade

1. Obtain the 65_orig packages listed in the preceding list on the Oracle Access Manager 10*g* (10.1.4.0.1) installation media and extract (untar) these to the original installation directory for each component.

   This creates a new directory named "orig" for each component. For example: *Component_install_dir*/identity/oblix/orig (or *Component_install_dir*/access/oblix/orig).

2. If your installation includes multiple languages, see "Preparing Multi-Language Installations" on page 8-6.

3. Finish preparing your environment as described in this chapter and start the upgrade.

Adding Packages for Release 6.5.2.x Patch

During the upgrade, the 10*g* (10.1.4.0.1) component installerrs create a directory named "orig" and compare the environment to ensure appropriate files are upgraded. If you originally installed release 6.5.0.x, then patched it to 6.5.2.x, you must extract and add the following packages to your original *Component_install_dir* before the upgrade.

Extract 652_orig Packages to the Original *Component_install_dir*
Netpoint_652_orig_en_COREid_Server_msg.zip
Netpoint_652_orig_COREid_Server_param.zip
Netpoint_652_orig_en_WebPass_msg.zip
Netpoint_652_orig_WebPass_param.zip
Netpoint_652_orig_en_Access_Manager_msg.zip
Netpoint_652_orig_Access_Manager_param.zip
Netpoint_652_orig_en_Access_Server_msg.zip
Netpoint_652_orig_Access_Server_param.zip
Netpoint_652_orig_en_WebGate_msg.zip
Netpoint_652_orig_WebGate_param.zip

Note: You complete the next procedure only if the 6.5.2 patch was applied to a 6.5.0 installation. If your 6.5.2 installation was installed without a patch, ignore this procedure.

To prepare a 6.5.2 environment for the upgrade

1. Obtain the 652 packages in the preceding list from the 10*g* (10.1.4.0.1) installation media and extract (untar) these to the original installation directory for each component.

   This creates a new directory named "orig" for each component. For example: *Component_install_dir*/identity/oblix/orig (or *Component_install_dir*/access/oblix/orig).

2. If your installation includes multiple languages, see "Preparing Multi-Language Installations" on page 8-6.

3. Finish preparing your environment as described in this chapter and start the upgrade.

Preparing Multi-Language Installations

There are two situations to consider, and both require action on your part before starting the upgrade:

- Preparing to Upgrade Release 6.5 with Multi-language Functionality
- Preserving 6.5 or 7.x Multi-language Functionality

Preparing to Upgrade Release 6.5 with Multi-language Functionality

When your release 6.5 installation includes Language Packs for French or German (or both), you need to extract and add to each source *Component_install_dir* the original component message files for all languages previously installed. No such files are provided nor required for release 7.0 multi-language environments.

Download and Extract 65-orig Packages to the Original *Component_install_dir*
Netpoint_65_orig_fr_COREid_Server_msg.zip

Download and Extract 65-orig Packages to the Original *Component_install_dir*
Netpoint_65_orig_COREid_Server_param.zip
Netpoint_65_orig_fr_Access_Manager_msg.zip
Netpoint_65_orig_Access_Manager_param.zip
Netpoint_65_orig_fr_WebPass_msg.zip
Netpoint_65_orig_WebPass_param.zip
Netpoint_65_orig_fr_Access_Server_msg.zip
Netpoint_65_orig_Access_Server_param.zip
Netpoint_65_orig_fr_WebGate_msg.zip
Netpoint_65_orig_WebGate_param.zip

Download and Extract 65-orig Packages to the Original *Component_install_dir*
Netpoint_65_orig_de_COREid_Server_msg.zip
Netpoint_65_orig_COREid_Server_param.zip
Netpoint_65_orig_de_Access_Manager_msg.zip
Netpoint_65_orig_Access_Manager_param.zip
Netpoint_65_orig_de_WebPass_msg.zip
Netpoint_65_orig_WebPass_param.zip
Netpoint_65_orig_de_Access_Server_msg.zip
Netpoint_65_orig_Access_Server_param.zip
Netpoint_65_orig_de_WebGate_msg.zip
Netpoint_65_orig_WebGate_param.zip

To prepare Release 6.5 multi-language installations for an upgrade

1. Before starting the upgrade, extract (untar) and add release 6.5 message and parameter bundles for each installed language to each original *Component_install_dir*. For example:

   Netpoint_65_orig_de_*Component*_msg.zip

   Netpoint_65_orig_fr_*Component*_msg.zip

   Netpoint_65_orig_*Component*_param.zip

   WARNING: In addition, you need to obtain the 10*g* (10.1.4.0.1) Language Packs.

2. Complete activities in "Preserving 6.5 or 7.x Multi-language Functionality" on page 8-8.

The files you added in this procedure are used during the upgrade. No further action is required by you.

Preserving 6.5 or 7.x Multi-language Functionality

If you upgrade a multi-language installation without including the corresponding 10*g* (10.1.4.0.1) Language Packs, you will lose the multi-language functionality. When you include 10*g* (10.1.4.0.1) Language Packs to the same directory as the 10*g* (10.1.4.0.1) installer, you can preserve your multi-language capability. After upgrading, you may add new languages as described in the *Oracle Access Manager Installation Guide*.

To preserve existing multi-language functionality

1. Locate and add any 10*g* (10.1.4.0.1) Language Packs that correspond to your earlier installed languages (there is no Language Pack for English) to the same directory as the 10*g* (10.1.4.0.1) component installer.
2. Upgrade your earlier installation as described in this manual.

Backing Up Directories, Web Server Configurations, and Registry Details

Oracle recommends that you complete activities in the topics here before upgrading to help ensure that you can roll back to the original installation should any problem arise:

- Backing Up the Existing Installed Directory
- Backing Up the Existing Web Server Configuration File
- Backing Up Windows Registry Data

Backing Up the Existing Installed Directory

Before starting each component (or customization) upgrade, Oracle recommends that you back up the installation directory for the earlier instance and store this backup in a different location. This will enable you to retrieve the original directory later should you need to restore the environment and rerun the upgrade.

As described earlier, a time-stamped directory of original files is created when you upgrade each component. This system-generated directory contains earlier original files that are sometimes accessed to compare content or extract customized information. The time-stamped directory is stored in the same location as the upgraded 10*g* (10.1.4.0.1) directory. For example:

C:\611*identity_server*\identity

C:\611*identity_server**identity_20060714_1701*

To back up the existing installed directory

1. On the machine hosting the component you will upgrade, locate the current installation directory. For example:

   C:\611*identity_server*\identity

2. Copy the directory and store the copy in a new location. For example:

   D:\611_backup*identity_server*\identity

Backing Up the Existing Web Server Configuration File

Before starting any Web server component upgrade (WebPass, Policy Manager, WebGate), Oracle recommends that you back up the existing Web server configuration file and store this backup in a different location.

To back up the existing Web Server configuration file

1. On the machine hosting the Web component you will upgrade, locate the existing Web server configuration file. For example:

   C:/*IHS_install_dir*/conf/httpd.conf

2. Copy the Web server configuration file and store the copy in a new location. For example:

   D:/*IHS_install_dir*/conf/httpd.conf

Backing Up Windows Registry Data

Before starting each component upgrade on a Microsoft Windows system, Oracle recommends that you back up any registry data that includes Oracle Access Manager (formerly NetPoint or COREid) data.

To back up Windows Registry data

1. Run the regedit command
2. Chose the key "My Computer\HKEY_LOCAL_ MACHINE\SOFTWARE\Oblix\Oblix NetPoint"
3. Click "Registry" in the menu then select "Export registry file…" in the drop-down. menu.
4. Specify the export file name in the "Export Registry File" dialog.
5. Save the exported registry file to a known location

Again, this will enable you to roll back to the previous installed environment and restart the upgrade, should a failure occur. Refer to your Windows documentation for details.

Stopping Servers and Services

Before you start to upgrade to10*g* (10.1.4.0.1), you must stop the earlier server or service on the machine hosting the component to be upgraded.

For example, if you use WebPass, you need to stop the Web server on which the WebPass is installed. For an Identity Server, you need to stop the Identity Server service.

Note: IIS users need to stop the IIS Admin Service.

To stop servers or services before the upgrade

1. Locate the machine hosting the component you will upgrade.
2. Stop the Web server (WebPass, Policy Manager, and WebGate) or the service (Identity Server and Access Server).
3. If you are upgrading any integration components, stop the corresponding Application or Portal Server. For example if you are upgrading Security Provider for WebLogic SSP, you must stop the corresponding WebLogic Application Server.

Logging in with Appropriate Administrative Rights

Whether you upgrade or install an Oracle Access Manager component, you must log in as a user with administrative rights. On Solaris, you need to run the upgrade as the user who installed the previous release of Oracle Access Manager, or as a user with higher privileges.

Whenever a schema and data upgrade is involved in the upgrade process, you need to login as a user who has permission to change the schema and data in the directory server. In other words, the bind DN you use must have permission to update the directory.

To login before the upgrade

1. Ensure that you have a userid and password that provides the rights you need to perform the upgrade as well as any schema and data changes that occur during the upgrade.
2. On the machine hosting the component to upgrade, log in as a user with administrative rights.

9

Upgrading Remaining Identity System Components

Activities in this chapter are intended for administrators responsible to upgrade earlier Identity System components (Identity Servers (formerly known as COREid Servers) and WebPass instances. Topics include:

- About Identity System Upgrades
- Upgrading Remaining Identity Servers
- Upgrading Remaining WebPass Instances
- Validating the Identity System Upgrade
- Backing Up Upgraded Identity Component Information
- Recovering From an Identity Component Upgrade Failure
- Looking Ahead

> **Note:** Ensure that the schema and data have been upgraded, as described in Part II, "Upgrading the Schema and Data". If your starting Oracle Access Manager release is earlier than 6.1.1, contact Oracle Support before upgrading: `http://www.oracle.com/support/contact.html`.

About Identity System Upgrades

Activities in this chapter must be completed in the sequence described herein:

- After upgrading the Identity System schema and data as described in Part II, "Upgrading the Schema and Data" (and if you have an existing Access System, after upgrading the Access System schema and data as described in Chapter 7, "Upgrading Access System Schema and Data")
- After preparing Identity system components as described in Chapter 8, "Preparing Components for the Upgrade", which may be performed just before upgrading each specific component instance

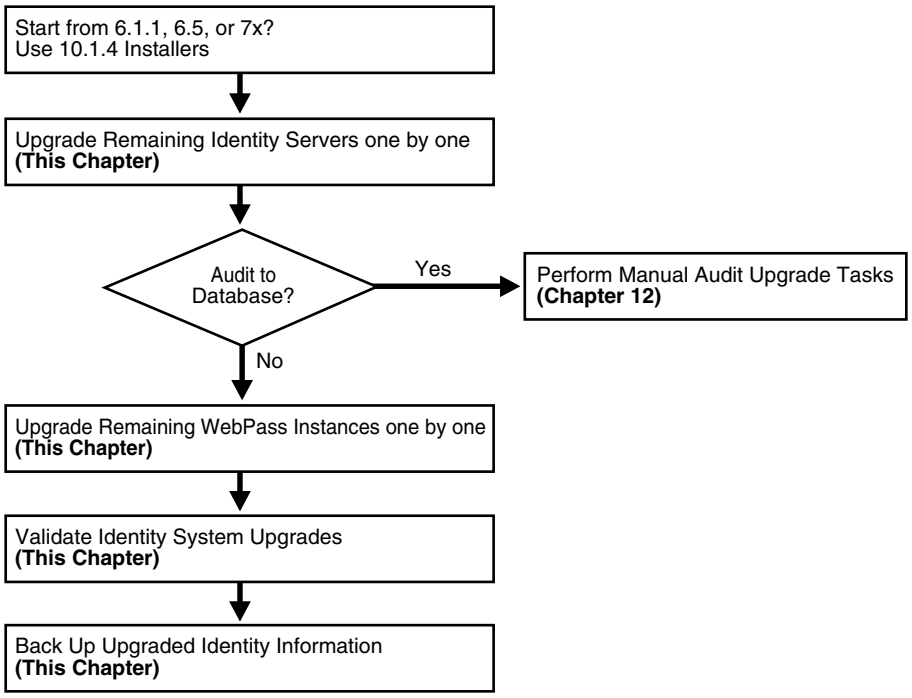
To upgrade remaining Identity System components, you use corresponding 10*g* (10.1.4.0.1) component installers and specify the same target directory as the existing component.

When your starting 6.5 or 7.x release includes multiple languages, you should upgrade these to retain your existing multiple language functionality.

Note: If you experience problems during any component upgrade, see "Accessing Log Files" on page F-1 and other topics in Appendix F, "Troubleshooting the Upgrade Process".

Figure 9–1 provides an overview of Identity System upgrade tasks. Additional details follow the graphic.

Figure 9–1 Identity System Upgrade Task Overview



Task overview: Upgrading Identity System components

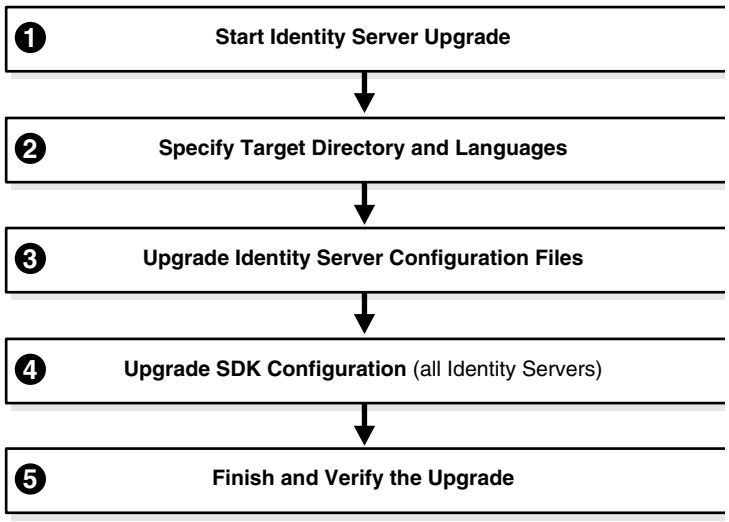
1. Upgrade remaining Identity Servers, one by one, as described in "Upgrading Remaining Identity Servers" on page 9-3.
2. **Audit to Database**: If you have auditing to a database configured in your earlier installation, before restarting the upgraded Identity Server service you must perform certain tasks manually to ensure proper auditing in 10*g* (10.1.4.0.1). See "Upgrading Auditing and Access Reporting for the Identity System" on page 12-2.
3. Upgrade remaining WebPass components, one by one, as described in"Upgrading Remaining WebPass Instances" on page 9-8.
4. Perform activities in "Validating the Identity System Upgrade" on page 9-11 to ensure that the upgrade is successful.
5. **Component Upgrade Successful**: Proceed to "Backing Up Upgraded Identity Component Information" on page 9-12.
6. **Component Upgrade Not Successful**: Proceed to "Recovering From an Identity Component Upgrade Failure" on page 9-12.

Note: If you experience problems during any component upgrade, see "Accessing Log Files" on page F-1 and other topics in Appendix F, "Troubleshooting the Upgrade Process".

Upgrading Remaining Identity Servers

Figure 9–2 illustrates the sequence of events during the program-driven upgrade process for remaining Identity Servers (and the decision points where you are asked to provide specific responses or input). These are a subset of the processes that occur during the schema and data upgrade, because the program automatically detects those changes and suppresses messages related to those events during subsequent Identity Server upgrades.

Figure 9–2 Remaining Identity Server Upgrade Process



Task overview: Upgrading remaining Identity Servers includes

1. Starting the Identity Server Upgrade
2. Specifying the Target Directory and Languages

   When upgrading remaining Identity Servers, you won't be asked about schema and data upgrades, because those upgrades are detected automatically.
3. Upgrading Identity Server Configuration Files
4. Upgrading the Software Developer Kit Configuration

   Oracle recommends that you accept the Software Developer kit (SDK) configuration upgrade for each Identity Server during the component upgrade. Certain Identity server functions depend on the SDK configuration.
5. Finishing and Verifying the Identity Server Upgrade

Identity Server Upgrade Prerequisites

Before you begin upgrading remaining Identity Servers, check the tasks in Table 9–1 and be sure to perform all tasks for each component instance before the upgrade. Failure to complete the prerequisites may adversely affect your upgrade.

Table 9–1 Identity Server Upgrade Prerequisites Checklist

Checklist	Identity Server Upgrade Prerequisites
	Review Part I, "Introduction"
	Complete activities in Part II, "Upgrading the Schema and Data"
	Complete activities in Chapter 8, "Preparing Components for the Upgrade" for this Identity Server instance, and: ■ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6. ■ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4.

Starting the Identity Server Upgrade

You complete the upgrade using the appropriate 10*g* (10.1.4.0.1) installer. This manual describes the process using GUI method and Automatic mode.

The process is similar regardless of the method and mode you choose, or your operating system. Differences are noted as needed and you may skip items that do not apply). For example, if you have a Unix environment you may skip steps related to Windows and vice versa:

To start an Identity Server upgrade

1. Complete all prerequisites for this instance as described in "Identity Server Upgrade Prerequisites".
2. Turn off the Identity Server service for this instance and log in as a user with the appropriate administrator privileges to update the Oracle Access Manager files.
3. Locate the component installer and launch the program:

   GUI Method, **Windows**:

   Oracle_Access_Manager10_1_4_0_1_win32_Identity_Server.exe

   Console Method, **Solaris**:

   ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_Identity_Server

   The Welcome screen appears.
4. Dismiss the Welcome screen by clicking Next.
5. Respond to the administrator question based upon your platform. For example:
   - **Windows**: If you are logged in with administrator rights, click Next (otherwise click Cancel, log in as a user with administrator privileges, then restart the installation).
   - **Unix**: Specify the username and group that the Identity Server will use, then click Next. Typically, the defaults are "nobody."

Specifying the Target Directory and Languages

During this sequence you must specify the same target directory as the existing Identity Server instance. When the earlier component is detected, you are asked if you want to upgrade. When you accept the upgrade, the target directory is created and 10*g* (10.1.4.0.1) files are extracted into it.

Even when your earlier environment is English only, you are asked to confirm the language to use as the default locale (default Administrator language). You are also asked to specify any languages to upgrade. You may install additional Language Packs after upgrading, as described in the *Oracle Access Manager Installation Guide*.

Unless indicated in the next steps the questions that you must respond to are the same regardless of your chosen installation method and mode.

To specify the target directory and languages

1. Choose the same installation directory as the earlier Identity Server, then click Next.
2. Accept the upgrade by clicking Yes, then click Next.
3. Ensure that a check mark appears beside English and any other languages you want to upgrade, then click Next.

   You may be presented with a list of languages that will be upgraded.
4. Confirm the languages listed by clicking Next.

   The next screen tells you that the existing installation has been saved and provides the time-stamped directory name containing all files from the previous installation.
5. Continue the upgrade by clicking Next.

   A new screen confirms the installation directory for 10*g* (10.1.4.0.1) and tells you how much space is needed for the installation.
6. Start the file extraction into the target directory by clicking Next.

   A status bar indicates the progress of the file extraction.
7. Press Enter to continue.

   ```
   Enter
   ```

   You are asked to specify a mode for the upgrade process: Automatic or Confirmed.

   Note: If you are installing in using the Console method, you are asked to run the command displayed in the transcript. On Unix, the command is printed to a file (start_migration), and a message is printed to run this file.

   ```
   ------------------------------------
   Please specify the mode for migration:
   '1' - Automatic (recommended)
      Each step is performed automatically.
      No interaction from the user is required.
   '2' - Confirmed
      Each step needs confirmation from the user.
   Enter choice ( '1' or '2' ) : 1
   ------------------------------------------
   ```

8. Enter the number that corresponds to the upgrade mode you prefer: For example:
   - **Automatic (recommended)**: Enter the number 1 to observe as the process completes automatically and respond to a few specific questions when needed.

- **Confirmed**: Enter the number 2 to receive a prompt that you must respond to before each and every event during the entire Identity Server upgrade process.

The declarative messages in this guide are based on the Automatic mode. In this case, you are informed as folders are created, files are copied, and catalogs are upgraded. For example:

```
Creating original folders ...
-----------------------------------------------------
Copying general configuration files
OK.
-----------------------------------------------------
Updating parameter catalogs ...
OK.
-----------------------------------------------------
```

When the upgrade program connects with the directory server, a transcript appears as shown next.

```
Starting migration (6.1.1 -> 6.5.0)
-----------------------------------------------------
```

9. Regardless of the mode you have chosen, continue with "Upgrading Identity Server Configuration Files", next.

Upgrading Identity Server Configuration Files

Component-specific configuration files are upgraded during this sequence. Depending on your starting release, aspects of the sequence may be repeated to bring your starting release up to 10*g* (10.1.4.0.1) incrementally. For example if your starting release is 6.1.1, component configuration files are incrementally upgraded to release 6.5, then again to release 7.0, then again to 10*g* (10.1.4.0.1).

During this sequence, you must type the full word "yes" or press the Enter key when asked to continue the upgrade through each sequence. In the example here, however, not all messages are shown.

To accept Identity Server-specific changes

1. Review messages for the migration to 10*g* (10.1.4.0.1).
2. Continue as directed, and review the final message. For example:

```
Enter

Updating component-specific configuration files...
OK.

Migration has completed successfully!
Press <ENTER> to continue :
------------------------++++++++++++++-------------
```

3. Proceed with "Upgrading the Software Developer Kit Configuration" next.

Upgrading the Software Developer Kit Configuration

The following functions in the Identity System require the Software Developer Kit (SDK, formerly known as the Access Server SDK (or Access SDK)):

- Automatic cache flush between the Identity System and Access System
- Automatic login to the Access System after self-registration

The SDK may have been manually configured to enable required functions, as described in your earlier version of the Oblix NetPoint or Oracle COREid *Administration Guide* (Volume 1 if you have a two volume set).By default, the SDK is installed in *IdentityServer_install_dir*\identity.

If your environment was configured to perform these functions, Oracle recommends that you upgrade the SDK during each Identity Server upgrade to preserve current configuration settings. When you accept the SDK upgrade, the process is launched automatically.

> **Note:** If you do not accept the automatic SDK configuration upgrade now, current SDK configuration settings are not preserved and you must reconfigure the SDK later using the configureAccessGate tool. For details, see the *Oracle Access Manager Identity and Common Administration Guide*. If the SDK was not configured for this specific Identity Server, you may skip this event when asked.

To upgrade the SDK configuration during the Identity Server upgrade

1. Review the SDK statements.

```
This component has the Access Server SDK installed

Would you like to automatically migrate the SDK at this time?

Note: If you do not want to migrate the SDK at this time, you will
need to reconfigure the SDK after migration has finished
by running the 'configureAccessGate' program
  '1' - Yes
  '2' - No
Enter choice ( '1' or '2' ) :
```

2. Respond to the question about migrating the SDK based on your environment.

```
   1
```

3. Continue as directed, then specify a mode for the SDK upgrade process: Automatic or Confirmed.

```
  -------------------------------------
  Please specify the mode for migration:
  '1' - Automatic (recommended)
     Each step is performed automatically.
     No interaction from the user is required.
  '2' - Confirmed
     Each step needs confirmation from the user.
  Enter choice ( '1' or '2' ) : 1
-----------------------------------------
    1
```

4. Continue as directed, then go to "Finishing and Verifying the Identity Server Upgrade" next.

Finishing and Verifying the Identity Server Upgrade

You complete this procedure to finish the upgrade for this Identity Server.

Caution: When your earlier environment includes auditing to a database, do not start the Identity Server service until you finish tasks in "Upgrading Auditing and Access Reporting for the Identity System" on page 12-2.

To finish and verify the Identity Server upgrade

1. **Auditing and Access Reporting**: If your earlier installation included auditing and access reporting, go immediately to "Upgrading Auditing and Access Reporting for the Identity System" on page 12-2 before performing step 2.
2. Start the Identity Server service to confirm that it will start (notice that the name has not changed from the one originally assigned).
3. **Identity Server Service Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
4. Check the migration log files for any errors reported during the upgrade, as described in "Accessing Log Files" on page F-1.
5. **Upgrade Not Successful**: Proceed to "Recovering From an Identity Component Upgrade Failure" on page 9-12.
6. **Upgrade Successful**: Upgrade every earlier Identity Server instance in your environment.
7. After upgrading *all* earlier Identity Server instances, proceed to "Upgrading Remaining WebPass Instances" next.

Upgrading Remaining WebPass Instances

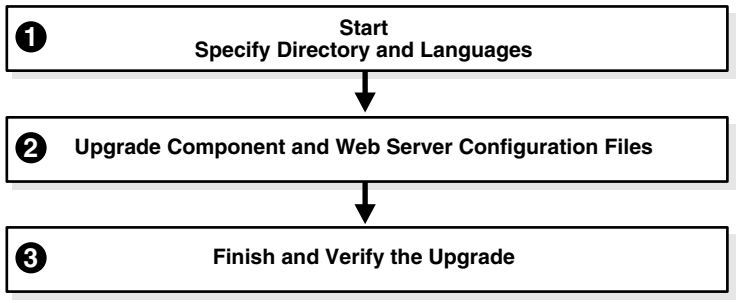
After all Identity Servers are upgraded, you can begin upgrading WebPass instances.

With WebPass, there is no connection to a directory server and, therefore, no schema or data upgrades. The component-specific upgrade includes both WebPass configuration files and Web server configuration updates. There are no differences between upgrading the master WebPass (accomplished earlier for the schema and data upgrade) and upgrading remaining WebPass instances.

Again, unless you are upgrading from release 7, the process repeats for each major release until you reach 10*g* (10.1.4.0.1).

Figure 9–3 illustrates events in the program-driven WebPass upgrade process as well and the points at which you must provide input.

Figure 9–3 WebPass Upgrade Process



Task overview: Upgrading remaining WebPass instances includes

1. Starting the WebPass Upgrade, Specifying the Target Directory and Languages
2. Upgrading WebPass Configuration Files and Web Server Configuration File
3. Finishing and Verifying the WebPass Upgrade

WebPass Upgrade Prerequisites

Before you begin upgrading any WebPass instance, check Table 9–2 to ensure you have completed all tasks. Failure to complete prerequisites may adversely affect your upgrade.

Table 9–2 WebPass Upgrade Prerequisites Checklist

Checklist	WebPass Upgrade Prerequisites
	Upgrade all Identity Servers as described in "Upgrading Remaining Identity Servers" on page 9-3.
	Complete activities in Chapter 8, "Preparing Components for the Upgrade" for this WebPass instance, and: ▪ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6. ▪ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4.

Starting the WebPass Upgrade, Specifying the Target Directory and Languages

The sample WebPass upgrade described here starts from release 6.1.1. The sequence of events and messages is directed by the program with very little input from you.

To start the WebPass upgrade

1. Complete all prerequisites for this instance as described in "WebPass Upgrade Prerequisites" on page 9-9.
2. Turn off this WebPass Web server.
3. Log in as a user with the administrator privileges to update the Web server configuration and Oracle Access Manager files.
4. Locate and launch the appropriate 10*g* (10.1.4.0.1) WebPass installer for this instance. For example:

   GUI Method Windows:
   Oracle_Access_Manager10_1_4_0_1_win32_NSAPI_WebPass.exe

   Console Method, Solaris:
   ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_NSAPI_WebPass

   The Welcome screen appears.
5. Dismiss the Welcome screen, then r.espond when asked about your administrator rights.
6. Specify the directory that contains the earlier WebPass instance.
7. Accept the upgrade when asked.
8. Ensure that a check mark appears beside English and any other languages you have or want installed, then continue.

   You may be presented with a list of languages that will be upgraded or added.

9. Confirm the languages listed by clicking Next.
10. Record the name of the time-stamped directory, then continue.
11. Start the file extraction.

    A status bar indicates the progress of the file extraction.

    Using the GUI method a new window appears asking you to specify either Automatic or Confirmed mode for the upgrade. Using the Console method, you are asked to run the command displayed in the transcript, then continue as instructed.

Upgrading WebPass Configuration Files and Web Server Configuration File

For brevity, steps are provided with little explanatory text. The command provided in the Console method transcript is referenced but not shown.

To upgrade the WebPass and Web server configuration

1. Enter the number that corresponds to the mode you prefer and follow the dialog on screen. For example:

```
-------------------------------------
Please specify the mode for migration:
'1' - Automatic (recommended)
   Each step is performed automatically.
   No interaction from the user is required.
'2' - Confirmed
   Each step needs confirmation from the user.
Enter choice ( '1' or '2' ) : 1
-------------------------------------------

  1

Creating orig folders ...
-----------------------------------------------------
Copying general configuration files ...
OK.
-----------------------------------------------------
Updating parameter catalogs ...
OK.
-----------------------------------------------------
Starting migration (6.1.1 -> 6.5.0)
-------------------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Starting migration ( 6.5.0 -> 7.0.0 )...
-------------------------------------
Updating web server configuration files...
OK.
-------------------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Starting migration (7.0.0 -> 10.1.4)
-------------------------------------
Updating web server configuration files...
OK.
-------------------------------------
Updating component-specific configuration files...
```

```
OK.
-------------------------------------
Migration has completed successfully!
Press <ENTER> to continue :
```

2. Continue as requested.

```
  Enter

If the Access System is also configured, you need to create a DB Profile
manually after first WebPass component upgrade is completed and before
upgrading the first Policy Manager. The profile gives the Access Server write
permission to Policy data in the directory server and will be used while
upgrading the WebGate component. The profile can be deleted after all
the WebGates are successfully upgraded.

Changing ownership of directory ...
(C:\NetPoint\webcomponent-iis\identity_20060426_163742\oblix ) ->
(C:\NetPoint\webcomponent-iis\identity\oblix )
-------------------------------------
```

3. Conclude the WebPass upgrade and proceed to the next discussion, "Finishing and Verifying the WebPass Upgrade".

> **Note:** Ignore the message about creating a temporary directory profile. This was performed after the schema and data upgrade.

Finishing and Verifying the WebPass Upgrade

You finish this WebPass upgrade as described in the following steps.

To finish the WebPass upgrade

1. Apply Web server changes, if needed.
2. Stop, then restart the associated Identity Server service.
3. Start the WebPass Web server instance.
4. **Web Server Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
5. Check the migration log files for any errors reported during the upgrade, as described in "Accessing Log Files" on page F-1.
6. **Upgrade Not Successful**: Proceed to "Recovering From an Identity Component Upgrade Failure" on page 9-12.
7. **Upgrade Successful**: Upgrade every WebPass instance in your environment.
8. After upgrading *all* WebPass instances, proceed to "Validating the Identity System Upgrade" next.

Validating the Identity System Upgrade

It is a good idea to quickly validate the following items to ensure that the overall Identity System upgrade was successful.

To confirm your Identity System upgrade

1. Delete all Web browser caches once the upgrade is complete.

2. Make sure your Identity Server service and WebPass Web server instance are running.
3. Check that your message and parameter catalog customizations have been preserved. For example, if you have changed any message in a particular message catalog file, then it needs to be retained.
4. Proceed to "Backing Up Upgraded Identity Component Information" next.

Backing Up Upgraded Identity Component Information

As mentioned earlier, Oracle recommends that you finish each component upgrade by backing up the upgraded 10*g* (10.1.4.0.1) component directory. This will enable you to easily restore your environment to the newly upgraded state should that be needed.

To back up critical information after the upgrade

1. Back up the 10*g* (10.1.4.0.1) component directory and store it in a new location.
2. **WebPass Web Server**: Back up the upgraded Web server configuration file, if required, using instructions from your vendor.
3. **Windows**: Back up the upgraded registry for the component as described in "Backing Up Windows Registry Data" on page 8-9.
4. Proceed to "Looking Ahead" on page 9-12.

Recovering From an Identity Component Upgrade Failure

If a component upgrade was not successful, you may perform the following steps to rollback this upgrade, then try again.

To recover from an unsuccessful Identity component upgrade

1. Restore the earlier component installation directory that you backed up before the upgrade (to recover the earlier environment), then back it up again. You will retain one of the earlier directories as a backup copy and use one to restart the upgrade.
2. **WebPass Web Server**: Restore the upgraded Web server configuration file, if required.
3. **Windows**: Restore the backed up registry for the component.
4. Using a backup copy of your earlier component installation directory (and Web server configuration, if needed), restart the upgrade as described in this chapter.

Looking Ahead

Upgraded Identity System components send and receive information sent in UTF-8 encoding. Earlier components send and receive data in Latin-1 encoding. As a result, the 10*g* (10.1.4.0.1) Identity System does *not* work with earlier Access System components.

When all earlier Identity System components are successfully upgraded, proceed as appropriate for your earlier installation. For example:

- **Identity System Only**: When your earlier installation does *not* include the Access System, you may complete activities in the following sequence using information in:

  - Chapter 11, "Upgrading Integration Components and an Independently Installed SDK"
  - Chapter 12, "Upgrading Your Identity System Customizations" after upgrading all Identity System components.
  - Chapter 14, "Validating the Entire System Upgrade"
- **Joint Identity and Access System**: In this case, you must complete activities in the sequence listed next using information in:
  - Chapter 10, "Upgrading Access System Components"
  - Chapter 11, "Upgrading Integration Components and an Independently Installed SDK"
  - Chapter 12, "Upgrading Your Identity System Customizations" after upgrading all Identity System components.
  - Chapter 13, "Upgrading Your Access System Customizations"
  - Chapter 14, "Validating the Entire System Upgrade"

For more information about expected system behaviors, see Chapter 4, "System Behavior and Backward Compatibility".

10

Upgrading Access System Components

If your environment does *not* include Access System components, you may skip this chapter. Activities in this chapter are intended for administrators responsible to upgrade an earlier Access System components. Topics include:

- About Access System Component Upgrades
- Upgrading Remaining Policy Managers
- Upgrading Access Servers
- Upgrading WebGates
- Backing Up Upgraded Access System Component Directories
- Recovering From an Access System Upgrade Failure
- Looking Ahead

> **Note:** You must upgrade the Access System as described in this chapter before upgrading integration components or an independently installed SDK.

About Access System Component Upgrades

Before you can use Oracle Access Manager access policies, you must upgrade the Access System components. Activities in this chapter must be completed in the sequence described herein:

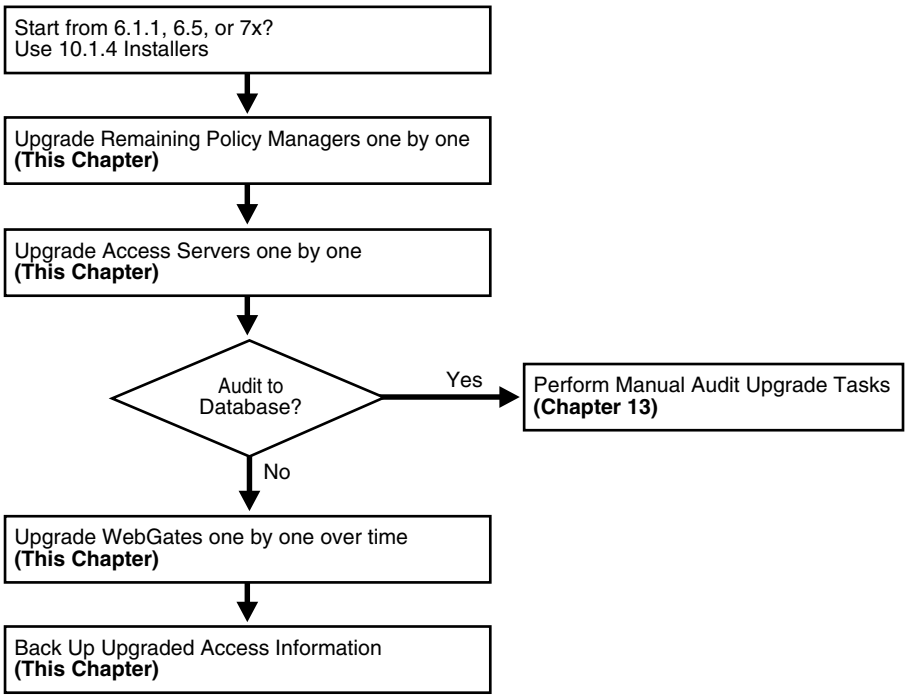
- After upgrading the schema and data as described in Part II, "Upgrading the Schema and Data"
- After upgrading remaining Identity System components as described in Chapter 9, "Upgrading Remaining Identity System Components"
- After preparing individual components as described in Chapter 8, "Preparing Components for the Upgrade", which may be performed just before upgrading each specific instance

To upgrade remaining Access System components, you use corresponding 10*g* (10.1.4.0.1) component installers and specify the same target directory as the existing component.

When your starting 6.5 or 7.x release includes multiple languages, you should upgrade these to retain your existing multiple language functionality.

Figure 10–1 provides an overview of Access System upgrade tasks.

Figure 10–1 Access System Upgrade Tasks



Task overview: Upgrading Access System components includes

1. Upgrading Remaining Policy Managersis described on page 10-2.
2. Upgrading Access Servers is described on page 10-6.
3. **Audit to Database**: If you have auditing to a database configured in your earlier installation, before restarting the upgraded Access Server service you must perform certain tasks manually to ensure proper auditing in 10*g* (10.1.4.0.1). See "Upgrading Auditing and Reporting for the Access Server" on page 13-2.
4. Upgrading WebGates is described on page 10-9. This activity does not need to occur all at one time because upgraded Access Servers are automatically backward compatible with earlier WebGates.
5. **Component Upgrade Successful**: Proceed to "Backing Up Upgraded Access System Component Directories" on page 10-13. This task should be performed after every component successful upgrade to enable you to quickly roll back to this upgrade if needed.
6. **Component Upgrade Not Successful**: Proceed to "Recovering From an Access System Upgrade Failure" on page 10-13.

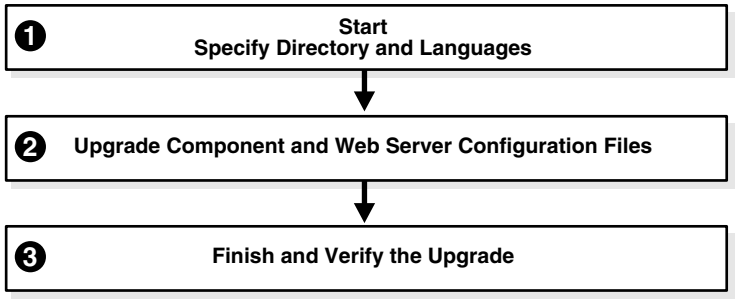
> **Note:** If you experience problems during any component upgrade, see "Accessing Log Files" on page F-1 and other topics in Appendix F, "Troubleshooting the Upgrade Process".

Upgrading Remaining Policy Managers

The name Policy Manager (formerly known as the Access Manager component) is used throughout this chapter. The master Policy Manager upgrade occurred with the Access System schema and data upgrade.

This discussion is divided into the events and decision points you must respond to when upgrading remaining Policy Manager instances, as shown in Figure 10–2. The updated schema and data is detected automatically and any corresponding messages and events are skipped.

Figure 10–2 Upgrade Process for Remaining Policy Managers



Note: If an earlier Policy Manager instance is installed in the same directory as an earlier WebGate on the same machine, you must upgrade the Policy Manager, the all Access Servers that communicate with the WebGate, and the WebGate before restarting the Web server.

Task overview: Upgrading remaining Policy Managers includes

1. Starting the Policy Manager Upgrade, Specifying a Target Directory and Languages
2. Upgrading Policy Manager and Web Server Configuration Files
3. Finishing and Verifying the Policy Manager Upgrade

Policy Manager Upgrade Prerequisites

Before you begin upgrading remaining Policy Managers, check the tasks in Table 10–1 to ensure you have completed these tasks before upgrading each instance in your earlier environment. Failure to complete prerequisites may adversely affect your upgrade.

Table 10–1 Policy Manager Upgrade Prerequisites Checklist

Checklist	Policy Manager Upgrade Prerequisites
	Familiarize yourself with information in Part I, "Introduction"
	Complete tasks in Part II, "Upgrading the Schema and Data".
	Perform all tasks in Chapter 9, "Upgrading Remaining Identity System Components".
	Complete activities in Chapter 8, "Preparing Components for the Upgrade" for this instance, and: ▪ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6. ▪ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4.

Starting the Policy Manager Upgrade, Specifying a Target Directory and Languages

In this sequence you start the process, specify the same target directory as the earlier Policy Manager component, and specify languages to upgrade.

Again, the steps here use GUI method and the recommended Automatic mode to illustrate messages you see, responses you give, and the sequence of events. The sample upgrade in this procedure starts from a release 6.1.1. Your starting release and environment may differ.

Note: Skip any details that do not apply to your installation. For example if you have a Unix environment, skip Windows details.

To start the Policy Manager upgrade, and specify a target directory and languages

1. Confirm that all prerequisites described in "Policy Manager Upgrade Prerequisites" on page 10-3 have been completed for this instance.
2. Stop the Policy Manager Web server instance and log in as a user with the appropriate administrator privileges to update the Oracle Access Manager files.
3. Locate and launch the installation program using your preferred method:

   GUI Method, Windows:

   Oracle_Access_Manager10_1_4_0_1_Win32_NSAPI_PolicyManager.exe

   Console Method, Solaris:

   ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_NSAPI_PolicyManager

   The Welcome screen appears.
4. Dismiss the Welcome screen as directed, then respond to the administrator question based upon your platform.
5. Choose the directory where you installed the earlier release, then continue as directed.
6. Accept the upgrade by clicking Yes, then click Next
7. Ensure that a check mark appears beside English and any other languages you have installed, then click Next.
8. Confirm the languages listed by clicking Next.
9. Record the time-stamped directory name, then click Next to continue.
10. Note the amount of disk space required, then click Next to start the file extraction into the target directory.

    You are asked to specify a mode for the upgrade process: Automatic or Confirmed.

    If you are using Console method, the installation script exits and a transcript appears. Run the command in the transcript then continue with step 9. (On Unix, the command is printed to a file (start_migration), and a message is printed to run this file.)
11. Press the number of your choice., then review messages that appear. For example:

```
    1

Creating orig folders ...
-------------------------------------------------
```

```
Copying general configuration files
OK.
-------------------------------------------------
Updating parameter catalogs ...
OK.
-------------------------------------------------
```

12. Continue with "Upgrading Policy Manager and Web Server Configuration Files".

Upgrading Policy Manager and Web Server Configuration Files

During this sequence the component-specific upgrade is performed. With the Policy Manager, this includes Web server configuration updates and upgrades for Policy Manager configuration parameters.

The following procedure provides only an abbreviated set of messages to give you an idea of what to expect. Your environment will vary.

To upgrade the Web Server and Policy Manager

1. Review messages and respond appropriately for your environment when asked.

```
-------------------------------------
Updating web server configuration files...
Connecting to server ...Done.
OK.
-------------------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Starting migration ( 6.5.0 -> 7.0.0 )...
-------------------------------------
Please type 'yes' to proceed:
```

2. Continue with component-specific configuration for release 7.0 or 10*g* (10.1.4.0.1), if needed.

```
     Enter
Updating component-specific configuration files.
```

3. Review the message to confirm the upgrade finished successfully.

```
Directory permissions copied ...
C:\NetPoint\WebComponent\access_20060223_190102\oblix)
C:\NetPoint\WebComponent\access\oblix)
-------------------------------------------------
Migration has completed successfully!
Press <ENTER> to continue.
```

4. When this phase completes, continue as instructed, then proceed to "Finishing and Verifying the Policy Manager Upgrade".

Finishing and Verifying the Policy Manager Upgrade

You finish the upgrade as described in this procedure.

To finish the Policy Manager upgrade

1. Apply any changes to the Web server configuration file, if needed.

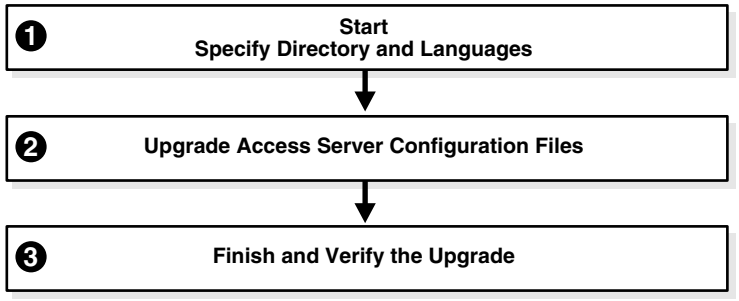
Important: If an earlier Policy Manager component is installed in the same directory as an earlier WebGate on the same machine, you must upgrade the Policy Manager, the all Access Servers that communicate with the WebGate, and the WebGate before restarting the Web server.

2. Start the Web server to confirm that this upgrade was successful.
3. **Policy Manager Web Server Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
4. View Policy Manager migration log files to see if they contain any errors. See "Accessing Log Files" on page F-1.
5. **Upgrade Successful**: Perform activities in "Backing Up Upgraded Access System Component Directories" on page 10-13 for this instance, then continue upgrading remaining Policy Managers.
6. **Upgrade Not Successful**: Proceed to "Recovering From an Access System Upgrade Failure" on page 10-13.
7. When all Policy Managers are upgraded and backed up, proceed with "Upgrading Access Servers" next.

Upgrading Access Servers

This discussion is divided into the events and decision points you will encounter when upgrading Access Server instances, as shown in Figure 10–3. There is no Web server involved in Access Server upgrades.

Figure 10–3 Access Server Upgrade Process and Tasks



Task overview: Upgrading the Access Server includes

1. Starting the Access Server Upgrade, Specifying a Directory and Languages
2. Upgrading Access Server Configuration Files
3. Finishing and Verifying the Access Server Upgrade

Access Server Upgrade Prerequisites

Before you begin upgrading the Access Server, check the tasks inTable 10–2 to ensure you have completed these tasks. Failure to complete prerequisites may adversely affect your upgrade.

Table 10–2 Access Server Upgrade Prerequisites Checklist

Checklist	Access Server Upgrade Prerequisites
	Upgrade the Schema and Data Part II, "Upgrading the Schema and Data"
	Upgrade all Identity System Components as described in Chapter 9, "Upgrading Remaining Identity System Components"
	Upgrade all Policy Managers as described in "Upgrading Remaining Policy Managers" on page 10-2.
	Complete activities in Chapter 8, "Preparing Components for the Upgrade" for this instance, and: ■ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6. ■ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4.

Starting the Access Server Upgrade, Specifying a Directory and Languages

The sample upgrade here starts from an existing Oracle Access Manager 6.1.1 installation. Again, you specify the same target directory as the earlier component, and languages to upgrade.

To start the Access Server upgrade and specify a target directory and languages

1. Confirm that all prerequisites described in "Access Server Upgrade Prerequisites" have been completed.
2. Log in as a user with the appropriate administrator privileges to update the Oracle Access Manager files.
3. Stop the Access Server service.
4. Locate and launch the program in your preferred method:

   GUI Method, **Windows**:

   Oracle_Access_Manager10_1_4_0_1_Win32_AccessServer.exe

   Console Method, **Solaris**:

   ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_AccessServer

   The Welcome screen appears.
5. Dismiss the Welcome screen, then r.espond to the next question based upon your platform.
6. Choose the directory where you installed the earlier component, then click Next.
7. Accept the upgrade by clicking Yes, then click Next.
8. Select a default administrator language from the list, and any others you are upgrading.
9. Ensure that a check mark appears beside English and any other languages you are upgrading, then click Next.
10. Confirm the languages, and click Next.
11. Record the time-stamped directory name, then click continue as directed.
12. Start the file extraction into the target directory.

13. Proceed to "Upgrading Access Server Configuration Files".

Upgrading Access Server Configuration Files

This sequence includes upgrading message and parameter catalogs, creating a directory profile for the Access Server, and completing the component configuration upgrade.

This example starts from Oracle Access Manager 6.1.1. If you started with another release, numbers in the following sequence will differ.

To upgrade the Access Server configuration files

1. Type a 1 to use Automatic mode (or 2 for Confirmed mode), then review and respond to messages as they appear. For example:

   1

   Messages begin.

```
Creating orig folders...
-------------------------------------
Copying general configuration files...
OK.
-------------------------------------
Updating parameter catalogs...
OK.
-------------------------------------
Starting migration ( 6.1.1 -> 6.5.0 )...
DBProfiles created.
-------------------------------------
Updating component-specific configuration files...
OK.
Please note the name of the Oracle Access Manager Access Server service :
NetPoint AAA Server (aaa-viking)
OK.
-------------------------------------
Starting migration ( 6.5.0 -> 7.0.0 )...
-------------------------------------
Updating component-specific configuration files...
OK.
Please note the name of the Oracle Access Manager Access Server service :
NetPoint AAA Server (aaa-viking)
OK.
-------------------------------------
Starting migration ( 7.0.0 -> 10.1.4 )...
-------------------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Migration has completed successfully!
Press <ENTER> to continue:
-------------------------------------
```

2. Record the name of the Access Server service, then press Enter.
3. Press Enter.

   This completes the sequence, and the usual ReadMe information appears.

Finishing and Verifying the Access Server Upgrade

You finish the upgrade of each instance as described in the next procedure.

Caution: If you have auditing to a database configured in your earlier environment, before restarting the Access Server service be sure to complete appropriate activities in "Upgrading Auditing and Reporting for the Access Server" on page 13-2.

To finish the Access Server upgrade

1. **Auditing and Access Reporting**: If your earlier installation included auditing and access reporting, go immediately to "Upgrading Auditing and Reporting for the Access Server" on page 13-2 before performing step 2.
2. Start the Access Server service. For example, if you do not store the server password in the password.lst file, use the following command:

   ```
   start_access_server -P mypassword port -d -t 61
   ```

   Certain command options may disable the hide option and cause a password to appear in the command line.
3. Provide the password at the prompt, if needed.

   On an IBM SecureWay directory server, the next time you start the Access Server it may take a few minutes for the dialog requesting the PEM pass phrase to appear.
4. **Access Server Service Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
5. View Access Server migration log files to see if they contain any errors. See "Accessing Log Files" on page F-1.
6. **Upgrade Not Successful**: Proceed to "Recovering From an Access System Upgrade Failure" on page 10-13.
7. **Upgrade Successful**: Perform activities in "Backing Up Upgraded Access System Component Directories" on page 10-13 for this instance, then repeat the procedure to upgrade all Access Servers in your environment.
8. After upgrading all Access Servers, you may continue with:
   - Upgrading WebGates, next.
   - For other options, see "Looking Ahead" on page 10-14.

Upgrading WebGates

Oracle recommends that you upgrade all WebGates. However, this does not need to occur all at one time. As mentioned before, earlier WebGates can communicate with upgraded Access Servers, which have backward compatibility automatically enabled during the upgrade. For details, see "Access System Behavior Changes" on page 4-24.

Before you start upgrading WebGates, however, be aware of the following important changes. When you upgrade a WebGate, the WebGateStatic.lst configuration file is removed. Configuration parameters that resided in this file move to the directory server and are made available through the AccessGate Configuration function in the Access System Console.

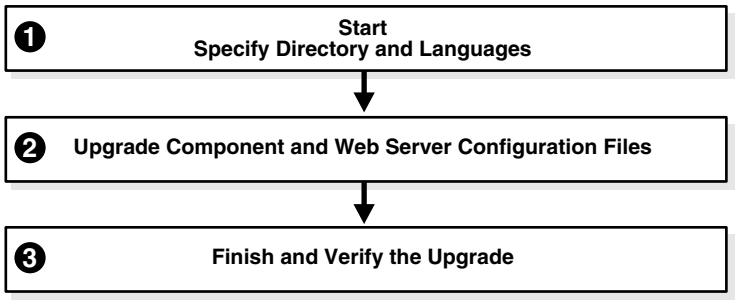
During a WebGate upgrade, the upgrade tool communicates with the Access Server to send configuration information from the WebGatestatic.lst file to be written to the

directory server. A temporary directory profile was created following the master Policy Manager upgrade for this purpose.

If WebGate configuration parameters do not migrate properly, you will not be able to add or change parameter values using the AccessGate Configuration function in the Access System Console. Following a WebGate upgrade, you cannot continue to use the WebGatestatic.lst file. For details about configuring WebGates, see the *Oracle Access Manager Identity and Common Administration Guide* for details. For more information about WebGate changes, see "WebGates" on page 4-29.

The procedures needed to guide you through a WebGate upgrade are provided next and shown in Figure 10–4. There is no update to the schema and data.

Figure 10–4 WebGate Upgrade Process and Tasks



Task overview: Upgrading the WebGate includes

1. Starting the WebGate Upgrade, Specifying a Target Directory and Languages
2. Upgrading WebGate and Web Server Configuration Files
3. Finishing and Verifying the WebGate Upgrade

WebGate Upgrade Prerequisites

Before you begin upgrading the WebGate, check the tasks in Table 10–3 to ensure you have completed these tasks.

Failure to complete prerequisites may adversely affect your upgrade.

Table 10–3 WebGate Upgrade Prerequisites Checklist

Checklist	WebGate Upgrade Prerequisites
	Upgrade the Schema and Data Part II, "Upgrading the Schema and Data"
	Upgrade all Identity System Components as described in Chapter 9, "Upgrading Remaining Identity System Components"
	Confirm that all Policy Managers are successfully upgraded, as described in "Upgrading Remaining Policy Managers".
	Confirm that all Access Servers are successfully upgraded, as described in "Upgrading Access Servers".
	Complete activities in Chapter 8, "Preparing Components for the Upgrade" for this instance, and: ▪ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6. ▪ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4.

Starting the WebGate Upgrade, Specifying a Target Directory and Languages

This is the same process as other upgrades you have completed. You start the upgrade, specify the same target directory as the earlier WebGate, and indicate the languages to upgrade. The sample here starts from release 6.1.1. Your environment may vary.

To launch the WebGate upgrade, and specify a target directory and languages

1. Confirm that all prerequisites described in "WebGate Upgrade Prerequisites" have been completed.
2. Stop the WebGate Web server instance, then log in as a user with the appropriate administrator privileges to update the Oracle Access Manager files.
3. Locate and launch the program in your preferred method:

   GUI Method, **Windows**:

   Oracle_Access_Manager10_1_4_0_1_Win32_NSAPI_WebGate.exe

   Console Method, **Solaris**:

   ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_NSAPI_WebGate

   The Welcome screen appears.
4. Dismiss the Welcome screen as instructed, then respond to the administrator question based upon your platform.
5. Choose the directory where you installed the earlier WebGate, then continue as directed.
6. Accept the upgrade, then continue.
7. Ensure that a check mark appears beside English and any other languages you have installed, then continue.
8. Confirm the languages that will be upgraded and continue.
9. Record the time-stamped directory name, then continue.
10. Note the amount of disk space required, then click Next.

    You complete activities according to the method you have chosen for the upgrade and respond as needed to continue.

    Note: On Windows, the path to directory security permissions is logged in obmigratenp.log.

11. Proceed to "Upgrading WebGate and Web Server Configuration Files".

Upgrading WebGate and Web Server Configuration Files

During this sequence the WebGate configuration files, and Web server configuration upgrades occur. Very little input from you is required during this automated process.

To upgrade the WebGate and Web server configuration files

1. Type a 1 to continue in Automatic mode (or 2 for Confirmed mode), then review and respond to messages as they appear. For example:

   1

Messages scroll by as the process continues.

```
Creating orig folders...
-------------------------------------
Copying general configuration files...
OK.
-------------------------------------
Updating parameter catalogs...
OK.
-------------------------------------
Starting migration ( 6.1.1 -> 6.5.0 )...
-----------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Starting migration ( 6.5.0 -> 7.0.0 )...
-------------------------------------
Updating web server configuration files...
OK.
-----------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Starting migration ( 7.0.0 -> 10.1.4 )...
-------------------------------------
Updating web server configuration files...
OK.
-----------------------------
Updating component-specific configuration files...
OK.
-------------------------------------
Starting the WebgateStatic.lst Migration ...
Completed the WebgateStatic.lst Migration successfully.
OK.

Directory permissions copied...
C:\NetPoint\access\webcomponent-iis\access_20040426_164541\oblix ) -> (
C:\NetPoint\access\webcomponent-iis\access\oblix )
-------------------------------------
Migration has completed successfully!
Press <ENTER> to continue:
-------------------------------------
```

2. Continue as directed, then proceed to "Finishing and Verifying the WebGate Upgrade".

Finishing and Verifying the WebGate Upgrade

There are a few differences when finishing the WebGate upgrade, relative to other component upgrades. For example, you need to ensure that the Access Management service for the WebGate is turned off.

To finish the WebGate upgrade

1. Apply any changes to the Web server configuration file, if needed.
2. **Turn off the Access Management service for this WebGate**: In the Access System Console, click Access System Configuration, then click AccessGate Configuration and click the link for the WebGate and see the *Oracle Access Manager Access Administration Guide* for details.

3. Start the WebGate Web server.
4. **WebGate Web Server Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
5. View WebGate migration log files to see if they contain any errors. See "Accessing Log Files" on page F-1.
6. Confirm that the WebGate performs as expected and that your 10*g* (10.1.4.0.1) environment is working. For more information, see Chapter 4, "System Behavior and Backward Compatibility".
7. **Upgrade Successful**: Perform activities in "Backing Up Upgraded Access System Component Directories" on page 10-13 for this instance, then continue upgrading earlier WebGates.
8. **Upgrade Not Successful**: Proceed to "Recovering From an Access System Upgrade Failure" on page 10-13.
9. Continue upgrading WebGates or proceed to "Looking Ahead" on page 10-14.

Backing Up Upgraded Access System Component Directories

As mentioned earlier, Oracle recommends that you finish each component upgrade by backing up the 10*g* (10.1.4.0.1) component directory after verifying that it is working properly. This will enable you to easily restore your environment to the newly upgraded state should that be a requirement.

To back up critical information after the upgrade

1. Back up the 10*g* (10.1.4.0.1) component directory and store it in a new location.
2. **Web Server**: Back up the upgraded Web server configuration file, if needed, using your vendor documentation as a guide.
3. **Windows**: Back up Windows registry data, if required, as described in "Backing Up Windows Registry Data" on page 8-9.
4. Proceed to "Looking Ahead" on page 10-14.

Recovering From an Access System Upgrade Failure

If the component was not successful, you may perform the following steps to rollback this upgrade, then try again.

To recover from an unsuccessful Access System component upgrade

1. Restore the earlier component installation directory that you backed up before the upgrade (to recover the earlier environment), then back it up again. You will retain one of the earlier directories as a backup copy and use one to restart the upgrade.
2. **Web Server**: Restore the backed up Web server configuration file, if required for this component (Policy Manager or WebGate).
3. **Windows**: Restore the backed up registry for the component, if needed for this instance.
4. Using a backup copy of your earlier component installation directory (and Web server configuration, if needed), restart the component upgrade as described in this chapter.

Looking Ahead

Upgraded Access System components send and receive information sent in UTF-8 encoding. Earlier components send and receive data in Latin-1 encoding. When all earlier Access System components are successfully upgraded, proceed as follows:

- Chapter 11, "Upgrading Integration Components and an Independently Installed SDK"
- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"
- Chapter 14, "Validating the Entire System Upgrade"

For more information about expected system behaviors, see Chapter 4, "System Behavior and Backward Compatibility".

11

Upgrading Integration Components and an Independently Installed SDK

When your installation includes only the Identity System, you may skip the upgrade of Access System integration connectors and upgrade the independently installed SDK. However, when your earlier installation includes the Access System and Oracle Access Manager integration connectors for certain third-party products, you must upgrade integration connectors before upgrading the SDK. Topics in this chapter include:

- Upgrading Third-Party Integration Connectors
- Upgrading Independently Installed Software Developer Kits
- Backing Up Upgraded Integration Connector or SDK Data
- Recovering From an Integration Connector or SDK Upgrade Failure
- Looking Ahead

Upgrading Third-Party Integration Connectors

When your environment includes the following integrations, you must complete procedures here to ensure compatibility with 10*g* (10.1.4.0.1):

- Security Provider for WebLogic SSPI
- Oracle Access Manager Connector for WebSphere

The task here is similar to other upgrades. The example provided in this chapter illustrates how to upgrade the Oracle Access Manager Security Provider for WebLogic SSPI. However, the procedures are similar for other integration connectors.

For the latest information about configuring release 10*g* (10.1.4.0.1) integrations, see the *Oracle Access Manager Integration Guide*.

Task overview: Upgrading third-party Integrations includes

1. Completing Integration Upgrade Prerequisites.
2. Starting the Integration Upgrade
3. Upgrading Security Provider for WebLogic SSPI
4. Finishing the Integration-Component Upgrade

The sample upgrade here starts from a Oracle Access Manager 6.1.1 installation. Your starting release may differ.

Integration Upgrade Prerequisites

Failure to complete prerequisites in Table 11–1 may adversely affect your upgrade.

Table 11–1 Integration Upgrade Prerequisites Checklist

Checklist	Integration Upgrade Prerequisites
	Schema and Data upgrade is successful as described in Part II, "Upgrading the Schema and Data".
	Component upgrades are successful as described in Part III, "Upgrading Components".
	Perform all required steps in Chapter 8, "Preparing Components for the Upgrade" for this instance and host, and: ■ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6. ■ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4.
	WebSphere: When upgrading the Oracle Access Manager Connector for WebSphere: ■ To run the Web Content Management Portlet on the 5.1.x Portal Server, ensure that `wmmGenerateExtId="false"` in the Portal Server wmm.xml, wmm_custom.xml, and wmm_DB.xml files. ■ To run the Web Content Management Portlet on the 5.0.x Portal Server, ensure that `wmmGenerateExtId="false"` in the Portal Server wmm.xml file.
	Weblogic: Ensure that the NetPointProvidersConfig.properties file in your current connector installation directory is synchronized with the one in your Weblogic server's domain directory.
	Stop the corresponding Application/Portal Server. For example if you are upgrading Security Provider for WebLogic SSPI then you must stop the corresponding WebLogic Application server.

Starting the Integration Upgrade

This is similar to upgrading other components. Should an error occur, the name of the log file that contains information about the error is identified. Skip any details that do not apply to your installation.

The sample upgrade in this procedure starts from an installation that is integrated with the Oracle Security Provider for WebLogic SSPI. Your environment may vary.

To launch the integration upgrade

1. Ensure that you have completed prerequisites for this instance as described in "Integration Upgrade Prerequisites".
2. Stop the corresponding Application/Portal Server. For example if you are upgrading Security Provider for WebLogic SSPI then you must stop the corresponding WebLogic Application server.
3. Log in as a user with administrator privileges.
4. Locate and launch the 10*g* (10.1.4.0.1) installer in your preferred method:

   GUI Method, Windows:
   Oracle_Access_Manager10_1_4_0_1_Win32_BEA_WL_SSPI.exe

   Console Method, Solaris:

./Oracle_Access_Manager10_1_4_0_1_sparc-s2_BEA_WL_SSPI

The Welcome screen appears.

5. Dismiss the Welcome screen by clicking Next, then respond to the question about administrator privileges based upon your platform.
6. Choose the directory where you installed the earlier integration component, then click continue as directed.
7. Accept the upgrade by clicking Yes, then click Next.
8. Complete any language questions, as described earlier, then click Next.
9. When the status screen indicates that this phase is complete, click Next.
10. Proceed to "Upgrading Security Provider for WebLogic SSPI" next.

Upgrading Security Provider for WebLogic SSPI

This procedure is the similar to other component upgrades. However, it does include several steps that are unique to the Security Provider for WebLogic SSPI.

To upgrade the Security Provider for WebLogic SSPI

1. Choose an upgrade mode: Automatic or Confirmed.
2. Follow the prompts onscreen.

   The GUI exits, and a command-line window appears with messages that keep you informed.

```
-----------------------------------
Starting migration (6.1.1 -> 6.5.0)
-----------------------------------------------------
Updating component-specific configuration files ...
OK.
-----------------------------------------------------
Starting migration (6.5.0 -> 7.0.0)
-----------------------------------------------------
Updating component-specific configuration files ...
OK.
-----------------------------------------------------
Starting migration (7.0.0 -> 10.1.4)
-----------------------------------------------------
Updating component-specific configuration files ...
OK.
-----------------------------------------------------
Migration has completed successfully!
Press <ENTER> to continue :
```

3. Upgrade the software developer kit (SDK); otherwise, current SDK configuration settings are not preserved and you must reconfigure the SDK later using the configureAccessGate tool, as described in the *Oracle Access Manager Access Administration Guide*.

Finishing the Integration-Component Upgrade

If you are upgrading the Security Provider for WebLogic SSPI, complete the following steps.

Note: If you are upgrading the integration component for WebSphere Application Server and Portal Server, you must copy the NetPointCMR.jar file to the *Portal_install_dir* and the NetPointWASRegistry.jar file and jobaccess.jar to the *WAS_install_dir* then restart the servers. See the *Oracle Access Manager Integration Guide* for details.

To finish the Security Connector upgrade

1. Copy the appropriate mbean jar file from following location. For example:

   From: *SecurityProvider_install_dir*/oblix/lib/mbeantypes

   To: *WebLogic_Home*/server/lib/mbeantypes

2. Copy the files here from your *SecurityProvider_install_dir* to your WebLogic domain folder.

   NetPointProvidersConfig.properties

   NetPointResourceMap.conf (only for the Application Server domain)

3. Start the Application/Portal/Web server to confirm that this upgrade was successful.

4. **Server Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".

5. View migration log files to see if they contain any errors. See "Accessing Log Files" on page F-1.

6. **Upgrade Successful**: Perform activities in "Backing Up Upgraded Integration Connector or SDK Data" on page 11-6 for this instance, then continue upgrading earlier Policy Managers.

7. **Upgrade Not Successful**: Proceed to "Recovering From an Integration Connector or SDK Upgrade Failure" on page 11-7.

8. After upgrading all integration connectors, proceed with "Upgrading Independently Installed Software Developer Kits" next.

Upgrading Independently Installed Software Developer Kits

The SDK (formerly known as the Access Server SDK) is now named the Access Manager SDK in 10*g* (10.1.4.0.1).

You need to upgrade any independently installed SDK as described here. The SDK upgrade that is invoked automatically as the last step when upgrading components bundled with the SDK (the Identity Server and Oracle Access Manager Security Connector for WebSphere SSPI, for example), has no impact on independently installed SDKs.

Task overview: Upgrading the Software Developer Kit includes

1. Completing all SDK Upgrade Prerequisites
2. Starting the SDK Upgrade, Specifying a Target Directory and Languages
3. Upgrading the SDK Configuration and Verifying the Upgrade

SDK Upgrade Prerequisites

Before you begin upgrading the Software Developer Kit, check the tasks in Table 11–2 to ensure you have performed these. Failure to complete prerequisites may adversely affect your upgrade.

Table 11–2 SDK Upgrade Prerequisites Checklist

Checklist	SDK Upgrade Prerequisites
	Complete activities in Part II, "Upgrading the Schema and Data".
	Complete activities in Part III, "Upgrading Components", as needed for your environment.
	Integration Components: Upgrade integration components, as described in "Upgrading Third-Party Integration Connectors" on page 11-1, if appropriate for your environment.
	Perform all required steps in Chapter 8, "Preparing Components for the Upgrade" for this instance and host, and: ▪ If you have a multi-language environment, see "Preparing Multi-Language Installations" on page 8-6. ▪ If you are upgrading a release 6.x installation, see "Preparing Release 6.x Environments" on page 8-4.

Starting the SDK Upgrade, Specifying a Target Directory and Languages

The sample upgrade here starts from a release 6.1.1 installation. Your starting release and environment may vary. Should an error occur, the name of the log file that contains information about the error is identified.

You may skip any details that do not apply to your installation.

To launch the SDK upgrade

1. Confirm that all activities in "SDK Upgrade Prerequisites" have been completed.
2. Turn off the server or service then log in as a user with administrator privileges.
3. Locate and launch the program in your preferred method:

   GUI Method, Windows:
   Oracle_Access_Manager10_1_4_0_1_Win32_AccessServerSDK.exe

   Console Method, Solaris:
   ./Oracle_Access_Manager10_1_4_0_1_sparc-s2_AccessServerSDK

   The Welcome screen appears.
4. Dismiss the Welcome screen, then respond to the administrator question based upon your platform.
5. Choose the directory where you installed the earlier SDK, then click Next
6. Accept the upgrade by clicking Yes, then click Next.
7. Ensure that a check mark appears beside English and any other languages you have installed, then click Next.
8. Confirm the languages listed by clicking Next.
9. Record the time-stamped directory name, then continue.

10. Record the amount of disk space required, then start the file extraction into the target directory.
11. **Unix**—Run the command indicated, then press Enter to continue.
12. Proceed to "Upgrading the SDK Configuration and Verifying the Upgrade" next.

Upgrading the SDK Configuration and Verifying the Upgrade

This procedure requires little input from you.

To upgrade the SDK configuration

1. Specify either Automatic or Confirmed, then continue.

   Status messages about the upgrade start scrolling by:

   ```
   -------------------------------------
   Starting migration ( 6.1.1 -> 6.5.0 )...
   -------------------------------------
   Copying general configuration files...
   OK.
   -------------------------------------
   Updating message catalogs...
   OK.
   -------------------------------------
   Updating parameter catalogs...
   OK.
   -------------------------------------
   Updating component-specific configuration files...
   OK.
   -------------------------------------
   ```

   The sequence will repeat until 10*g* (10.1.4.0.1) is reached, then you will see the message:

   ```
   -------------------------------------
   Migration has completed successfully!
   Press <ENTER> to continue :
   ```

2. Finish the upgrade as directed, then restart the server service.
3. **Server or Service Does Not Start**: See Appendix F, "Troubleshooting the Upgrade Process".
4. View migration log files to see if they contain any errors. See "Accessing Log Files" on page F-1.
5. **Upgrade Successful**: Perform activities in "Backing Up Upgraded Integration Connector or SDK Data" on page 11-6.
6. **Upgrade Not Successful**: Proceed to "Recovering From an Integration Connector or SDK Upgrade Failure" on page 11-7.
7. Repeat for each independently installed SDK in your environment, then see "Looking Ahead" on page 11-7.

Backing Up Upgraded Integration Connector or SDK Data

As mentioned earlier, Oracle recommends that you finish each component upgrade by backing up the 10*g* (10.1.4.0.1) component directory after verifying that it is working

properly. This will enable you to easily restore your environment to the newly upgraded state should that be a requirement.

To back up critical information after the integration connector or SDK upgrade

1. Back up the upgraded 10*g* (10.1.4.0.1) integration connector or SDK directory and store it in a new location.
2. **Web Server**: Back up the upgraded Web server configuration file, if needed, using your vendor documentation as a guide.
3. Back up Windows registry data if required.

Recovering From an Integration Connector or SDK Upgrade Failure

If the component was not successful, you may perform the following steps to rollback this upgrade, then try again.

To recover from an unsuccessful integration connector or SDK upgrade

1. Restore the earlier directory that you backed up before this upgrade (to recover the earlier environment), then back it up again. You will retain one of the earlier directories as a backup copy and use one to restart the upgrade.
2. **Windows**: Restore the backed up registry for the component (to recover the earlier environment).
3. **Web Server**: Restore the earlier backed up Web server configuration file, if required for this component (to recover the earlier environment).
4. Using a backup copy of your earlier environment, restart the upgrade as described in this chapter.

Looking Ahead

Upgraded Identity and Access System components send and receive information sent in UTF-8 encoding. Earlier components send and receive data in Latin-1 encoding. To continue the upgrade, proceed as appropriate for your earlier installation. For example:

- **Identity System Only**: When your earlier installation does *not* include the Access System, you complete activities in the sequence listed here using information in:
  - Chapter 12, "Upgrading Your Identity System Customizations"
  - Chapter 14, "Validating the Entire System Upgrade"
- **Joint Identity and Access System**: In this case, you must complete activities in the following sequence using information in:
  - Chapter 12, "Upgrading Your Identity System Customizations"
  - Chapter 13, "Upgrading Your Access System Customizations"
  - Chapter 14, "Validating the Entire System Upgrade"

For more information about expected system behaviors, see Chapter 4, "System Behavior and Backward Compatibility".

Part IV

Upgrading Your Customizations

This part of the book describes how to upgrade your earlier customizations to ensure compatibility with 10*g* (10.1.4.0.1) functionality.

Part IV contains the following chapters:

- Chapter 12, "Upgrading Your Identity System Customizations"
- Chapter 13, "Upgrading Your Access System Customizations"

12

Upgrading Your Identity System Customizations

After all Identity System components have been upgraded, you may need to perform one or more activities in this chapter to ensure that any Identity System customizations in your earlier environment are working in your 10*g* (10.1.4.0.1) implementation. This chapter includes the following topics:

- Prerequisites and Guidelines
- Upgrading Auditing and Access Reporting for the Identity System
- Combining Challenge and Response Attributes on a Panel
- Confirming Identity System Failover and Load Balancing
- Migrating Custom Identity Event Plug-Ins
- Ensuring Compatibility with Earlier Portal Inserts
- About Custom Items and Upgrades
- Incorporating Customizations from Release 6.5 and 7.x
- Incorporating Customizations from Releases Earlier than 6.5
- Validating Identity System Customization Upgrades
- Backing Up Upgraded Identity System Customizations
- Recovering from an Identity System Customization Upgrade Failure
- Looking Ahead

> **Note:** The activities here depend on what was implemented in your earlier installation. You may skip any task in this chapter that is not relevant for your earlier environment.

Prerequisites and Guidelines

Before starting to upgrade any Identity System customizations, Oracle recommends that you:

- Review information in "Customization Upgrade Planning" on page 1-13.
- Back up the directory containing the earlier customization and store it in a new location to help you if you need to roll back to this later.

After completing and testing each upgraded customization, Oracle recommends that you back up the directory containing the upgraded customization and store it in a new location.

Upgrading Auditing and Access Reporting for the Identity System

If your earlier installation was configured for auditing and access reporting, you need to complete specific activities in this discussion before starting your Identity Server after the upgrade. However, if your earlier installation was not configured for auditing and access reporting, you may skip this discussion.

Oracle Access Manager 10*g* (10.1.4.0.1) supports the Unicode standard. The Oracle equivalent for the Unicode UTF-8 standard is the AL32UTF8 character set. The code used to process this character set resides within the libraries bundled with each Oracle Access Manager 10*g* (10.1.4.0.1) component and is installed automatically. To support all the languages available with Oracle Access Manager 10*g* (10.1.4.0.1), the definitions of auditing and reporting tables have changed.

WARNING: Retain your earlier auditing database to preserve the original data. Simply upgrading or altering existing database instances and tables is not supported and could result in permanent truncation and loss of existing data.

After upgrading the first Identity Server, you need to create a new database instance to operate with 10*g* (10.1.4.0.1). All Identity and Access Servers audit to the same database instance. Therefore, you need only create a new database instance following the upgrade of the first Identity Server (not for additional Identity Server instances nor for the Access Server).

You need to upload the new Audit table schema (to support the auditing of 10*g* (10.1.4.0.1) UTF-8 data and the writing of this data to the new SQL Server instance). To accomplish this, you must create a new oblix_audit_events table for the auditing application. This schema upgrade includes datatype changes within the Audit table columns, as discussed in "Database Record Sizing" on page 12-5.

Next you need to create tables for the reporting application (oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users) in 10*g* (10.1.4.0.1).

Note: Whether you have only one or multiple Identity Server instances, you set up a new audit database instance, upload the audit schema, and create new tables for the reporting application only once.

To query or generate any report that requires data from both the old and new database, you need to import data from the original database instance into the new instance *before* you start auditing with 10*g* (10.1.4.0.1). For each Identity Server instance (and Access Server instance), you import earlier audit data into the new audit database instance. Otherwise, you cannot generate any report that requires data from both the old and new database.

Note: Be sure to retain the earlier database to preserve the original data. Importing earlier data may result in truncation of data and some data loss.

Finally, you need to change the DSN (ODBC Data Source Name used by the RDBMS profile of audit & reporting applications) to refer to the new database instance. If you have multiple Identity Servers on the same machine, be sure to upgrade all instances on the machine before you change the DSN to refer to the new database. For each Identity Server instance (and Access Server instance), you need to change the DSN to refer to the new database instance.

If you observe any problem with the characters (for example, Latin-1) after importing data and before auditing with 10*g* (10.1.4.0.1):

- For the first Identity Server, you need to create a new database instance with the proper configuration and import your original data again.
- For later Identity Servers, you need to delete all newly inserted audit records (which can be differentiated with the help of the `serverId` field) and try importing them again.

Even when you have an English only environment, certain steps depend on the type of database you are using. For more information, see:

- Upgrading Auditing and Reporting with a Microsoft SQL Server
- Upgrading Auditing and Reporting with an Oracle Database

Upgrading Auditing and Reporting with a Microsoft SQL Server

The default character set for the Microsoft SQL Server is UCS-2 (also known as UTF-16 Unicode format). UCS-2 includes all languages supported by Oracle Access Manager 10*g* (10.1.4.0.1) and mimics the way in which the 32-bit Windows kernel stores information so that data does not need to be converted to another format. As a result, no character set change is required for the Microsoft SQL Server for Oracle Access Manager 10*g* (10.1.4.0.1) globalization support.

Earlier data may be truncated during the import, as described in "Database Record Sizing" on page 12-5.

Refer to the task overview here for the sequence of tasks you must perform. For more information about individual steps, including uploading the audit schema, see the *Oracle Access Manager Identity and Common Administration Guide*.

Task overview: Ugrading auditing and reporting with a Microsoft SQL Server

1. Retain the original database, as is, to preserve your original data.
2. After upgrading the first Identity Server (and before restarting the Identity Server Service), set up a new audit database instance for 10*g* (10.1.4.0.1) using instructions in the *Oracle Access Manager Identity and Common Administration Guide*.
3. Create a new oblix_audit_events table for the 10*g* (10.1.4.0.1) auditing application (which will upload the 10*g* (10.1.4.0.1) schema definition). For information about uploading the audit schema, see the *Oracle Access Manager Identity and Common Administration Guide*.
4. Create new oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users. tables for the 10*g* (10.1.4.0.1) reporting application as described in the *Oracle Access Manager Identity and Common Administration Guide*.

> **Note:** You complete steps 1 through 4 for only the first Identity Server in your environment (even when you have multiple Identity Servers).

5. Review information in "Database Record Sizing" on page 12-5.

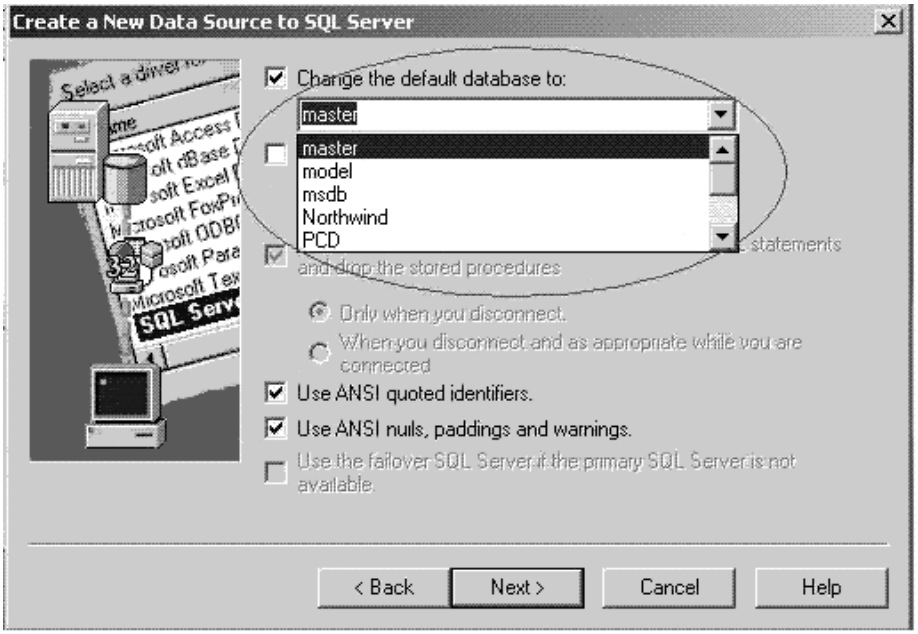
6. **Optional**: Import the earlier data audited by this Identity Server instance into the 10*g* (10.1.4.0.1) database and confirm that it is imported successfully. You will repeat this step for each Identity Server instance.

   The `serverId` field in audit table indicates the ID of the Identity Server that audited that record. Based on the `serverId` field, it is feasible to differentiate the records audited by each Identity Server instance. The same rule applies to the Access Server, as discussed later.

7. Change the DSN (ODBC Data Source Name used by the RDBMS profile of audit & reporting applications) on this machine to refer to the new database instance. For example:

   Note: If you have multiple Identity Servers on the same machine, be sure to upgrade all Identity Server instances on this machine before you change the DSN to refer to the new database. In this case, skip to step 9.

Figure 12–1 Create a New Data Source to SQL Server Window



8. Start the Identity Server service.

   The Identity Server will now audit and store data in the new database instance. However, other Identity Servers (and Access Servers) will continue to audit and store data in the old database instance.

9. Upgrade all other Identity Server instances as follows:

   - Upgrade the next Identity Server instance but do not restart the Identity Server service.
   - Repeat step 6 to import data for this Identity Server instance.

- Repeat step 7 to change the DSN (ODBC Data Source Name used by the RDBMS profile of the audit & reporting applications) on this machine to refer to the new database instance.

  Note: If you have multiple Identity Servers on the same machine, be sure to upgrade all Identity Server instances on this machine before you change the DSN to refer to the new database.

- Repeat step 8 to restart the Identity Server service on this machine.
- Repeat this step (9) for all Identity Servers in your environment.

10. After upgrading all Identity Server instances, upgrade all WebPass instances then complete the rest of the Identity System deployment-specific activities in this chapter before starting to upgrade the Access System.
11. Start auditing, as described in the *Oracle Access Manager Identity and Common Administration Guide*.

Database Record Sizing

For the SQL Server, the maximum length of a database record is 8096 bytes. The Oracle Access Manager Audit table contains 27 columns (23 of which are of type varchar). The previous Oracle Access Manager (release 7.0.4, also available as part of Oracle Application Server 10g Release 2 (10.1.2)) record size was 23 columns * varchar(255) + four additional columns), which equals less than the SQL Server maximum of 8096 bytes for each database record.

To support UTF-8 data in Oracle Access Manager 10*g* (10.1.4.0.1), the column types have changed from varchar(255) to nvarchar(170). When the column data type is nvarchar, the SQL Server stores data in UTF-16 encoding. In this case, 23 columns * nvarchar(170) * 2 + four additional columns equals slightly less than 8096 bytes.

In earlier Oracle Access Manager releases, only values greater than 255 characters were truncated. In 10*g* (10.1.4.0.1), however, any column value that exceeds 170 characters is truncated before inserting the record into the SQL Server audit database.

For the reasons stated earlier, upgrading the existing database could result in permanent data loss. Therefore, with the SQL Server you need to retain the original database as is, create and set up a new database, a new Audit Table for Oracle Access Manager with the 10*g* (10.1.4.0.1) schema definition, then create new auditing and reporting tables.

As stated earlier, you may import data from the original database instance into the new database instance (which may be truncated) in order to query or generate any report that requires data from both the old and new database. You will still have the original database instance and data.

See "Task overview: Ugrading auditing and reporting with a Microsoft SQL Server" on page 12-3 and the *Oracle Access Manager Identity and Common Administration Guide* for more information.

Upgrading Auditing and Reporting with an Oracle Database

As described earlier, to support all the languages available with Oracle Access Manager 10*g* (10.1.4.0.1), the definitions of oblix_audit_events, oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users tables have changed. After upgrading

the first Identity Server, you need to create a new Oracle database instance with AL32UTF8 as character set and UTF-8 as National character set.

> **Note:** Upgrading the database instance and tables is not supported.

You must complete activities outlined in the task overview here because upgrading the database instance and tables is not supported. For more information about the steps, including uploading the audit schema, see the *Oracle Access Manager Identity and Common Administration Guide* and your vendor documentation.

Task overview: Ugrading auditing and reporting with an Oracle database

1. Retain the original database as is, to preserve your original data for import.
2. After upgrading the first Identity Server, set up a new Oracle audit database instance with AL32UTF8 as the character set and UTF8 as the National character set for the Oracle database. See also the *Oracle Access Manager Identity and Common Administration Guide*.
3. Create a new oblix_audit_events table for the 10*g* (10.1.4.0.1) auditing application (which will upload the 10*g* (10.1.4.0.1) schema definition). For information about uploading the audit schema, see the *Oracle Access Manager Identity and Common Administration Guide*.
4. Create new oblix_rpt_as_reports, oblix_rpt_as_resources, and oblix_rpt_as_users. tables for the 10*g* (10.1.4.0.1) reporting application as described in the *Oracle Access Manager Identity and Common Administration Guide*.

   > **Note:** You complete steps 1 through 4 for only the first Identity Server in your environment (even when you have multiple Identity Servers).

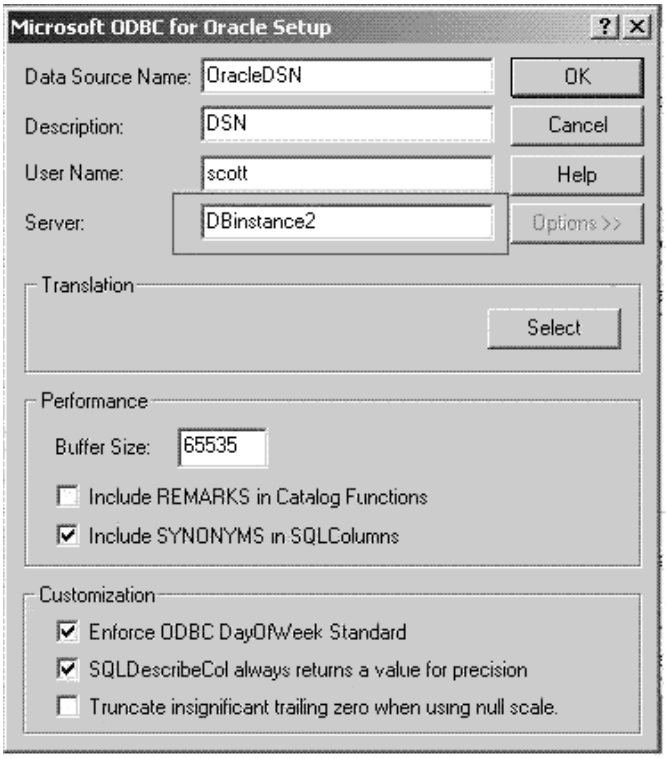
   To query or generate any report that requires data from both the old and new database, you need to import data from the original instance into the new instance before you start auditing with 10*g* (10.1.4.0.1).
5. Import earlier data audited by this Identity Server instance into the 10*g* (10.1.4.0.1) database and confirm that it imported successfully using your Oracle database documentation for details.

   > **Note:** You will repeat this step for each Identity Server instance. There is no truncation of data during the import, because the Audit table column size is 255 characters. If after importing data and before auditing with 10*g* (10.1.4.0.1) you observe any problem in characters (Latin-1), create a new database instance with the proper configuration (including but not limited to AL32UTF8 as character set and UTF-8 as National character set) and import your original again.

6. Change the DSN (ODBC Data Source Name used by the RDBMS profile of audit & reporting applications) on this machine to refer to the new database instance. See your vendor documentation for details about performing this task. For example:

Note: If you have multiple Identity Servers on the same machine, be sure to upgrade all Identity Server instances on this machine before you change the DSN to refer to the new database. In this case, skip to step 7.

Figure 12–2 Microsoft ODBC for Oracle Window



7. Start the Identity Server service.

   The Identity Server will now audit and store data in the new database instance. However, other Identity Servers (and Access Servers) will continue to audit and store data in the old database instance.

8. Upgrade all other Identity Server instances as follows:

   - Upgrade the next Identity Server instance but do not restart the Identity Server service.
   - Repeat step 5 to import data for this Identity Server instance.
   - Repeat step 6 to change the DSN (ODBC Data Source Name used by the RDBMS profile of the audit & reporting applications) on this machine to refer to the new database instance.

     Note: If you have multiple Identity Servers on the same machine, be sure to upgrade all Identity Server instances on this machine before you change the DSN to refer to the new database.

   - Repeat step 7 to restart the Identity Server service on this machine.

- Repeat this step (8) for all Identity Servers in your environment.

9. After upgrading all Identity Server instances, upgrade all WebPass instances then complete the rest of the Identity System deployment-specific activities in this chapter before starting to upgrade the Access System.
10. Start auditing, as described in the *Oracle Access Manager Identity and Common Administration Guide*.

Combining Challenge and Response Attributes on a Panel

In earlier releases, the challenge phrase and response attributes were allowed on different panels of the Profile page of the User Manager, Group Manager, and Organization Manager. In 10*g* (10.1.4.0.1), however, both the challenge phrase and response attributes must be on the *same* panel. In 10*g* (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.

> **Note:** If your original installation included both the challenge phrase and response attribute on a single panel, you may skip this discussion.

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

If challenge and response attributes are present in different panels in Identity System application configuration pages (User Manager Configuration, Group Manager Configuration, or Org. Manager Configuration), you must move these into a single panel.

The next task overview outlines the steps you need to perform to combine the challenge phrase and response attributes on a single panel. For more information about configuring Tab Profile Pages and Panels, configuring attributes, and assigning challenge and response semantic types to attributes for lost password management, see the *Oracle Access Manager Identity and Common Administration Guide*.

The User Manager Configuration page was selected in this example. However, the procedure is similar if you modify panels on Group Manager Configuration or Org. Manager Configuration pages.

Task overview: Combine challenge and response attributes on a single panel

1. Navigate to the Modify Panels page containing the Response attribute. For example:

   Identity System Console, User Manager Configuration
   Tabs *existing_tab_link* (for Response attribute panel)
   View Object Profile, Configure Panels
   panel_name, Modify

   The Modify Panel page appears.

2. **Remove the Response Attribute**: From the list of attributes on the Modify Panel page, locate the Response attribute and select ---- from the list, then click the Save button.

3. **Add the Response Attribute**: Navigate to the Modify Panel page containing the challenge phrase attribute, click the Add button, then select the Response attribute and click the Save button.

   Identity System Console, User Manager Configuration
   Tabs *existing_tab_link* (for Challenge attribute panel)
   View Object Profile, Configure Panels
   panel_name, Modify

IdentityXML changes have also been made for this feature. For details, see the *Oracle Access Manager Developer Guide*. See also "Challenge Response May Not Convert Properly" on page F-4.

Confirming Identity System Failover and Load Balancing

Your earlier implementation may include failover between an Identity Server and the directory server. The Identity Server failover configuration has resided in the directory server profile since release 5.2. As a result, there is **no** migration of parameters from failover configuration files to directory profiles. Although the schema itself has changed, migration of these changes is performed automatically during the upgrade.

There is also no impact on Identity System connection pools. The values for Initial Connections and Maximum Connections specified in the Database Instance profile are retained and will operate as they did previously.

> **Note:** For concurrent authentication requests on NDS directory servers, Oracle recommends that you increase the connection pool size to something higher than the default (1) for the user directory profile using the System Console.

During the upgrade, you do not need to complete any special handling for failover or load balancing. After upgrading Identity System components, simply test to ensure that any failover or load balancing that you had previously configured for the Identity System is still operating as expected.

You can use the following procedure to view details in the Database Instance Profile before testing.

To view failover, load balancing, and connection pool details for the Identity System

1. From the Identity System Console, select System Configuration, Directory Profiles.
2. Under the heading Configure LDAP Directory Server Profiles, select the name of the Profile you want to check.
3. On the Directory Server Profile page, confirm the servers that use the failover information and confirm that the information matches previous settings. For example:
   - Maximum Active Servers
   - Failover Threshold
   - Sleep For (Seconds)
   - Max. Session Time (Min.)

4. Locate the Database Instances list on the Directory Server Profile page and select the name of the Database Instance Profile you want to check.
5. In the Database Instance Profile, verify the values for `Initial Connections` and `Maximum Connections`.
6. Make any changes needed and save the profile.
7. Perform a test to ensure that everything is working as expected.

For more information about configuring failover and load balancing, see the *Oracle Access Manager Deployment Guide*.

Migrating Custom Identity Event Plug-Ins

When your original installation included custom Identity Event Plug-ins, Oracle recommends that you complete this activity immediately after upgrading all Identity System components.

Earlier Identity Event Plug-ins are not copied during the upgrade. At a minimum, you need to move your earlier plug-ins from the renamed source directory to the target directory as indicated in the procedure here.

To send or receive internationalized data you need to re-design plug-ins to use UTF-8 encoding. Also, on Solaris and Linux, plug-ins earlier than release 7.x must be re-compiled using the GCC v3.3.2 C++ compiler. For more information, see "Plug-ins" on page 3-10.

Note: Release 7.0 plug-ins as well as earlier plug-ins implemented as executables or those using a scripting language (such as perl) do not require recompiling after the upgrade. However, to send and receive internationalized data, earlier plug-ins should be redesigned to communicate using UTF-8 encoding.

To use earlier custom Identity Event plug-ins with 10*g* (10.1.4.0.1)

1. Create a folder in the top level of your Identity Event API directory and copy your earlier Identity Event plug-ins in to the new directory.
2. Redesign Identity Event plug-ins to use UTF-8 encoding, if desired.
3. Recompile release 5.2 or 6.x plug-ins on Solaris and Linux platforms using the GCC v3.3.2 compiler.

WARNING: You must use the GCC v3.3.2 compiler, regardless of the compiler that may be provided with the Operating System.

---

4. Complete any testing to ensure your plug-ins are working properly with 10*g* (10.1.4.0.1).
5. When using plug-ins that send and receive data in Latin-1 encoding, ensure that any new Identity Servers added to the upgraded environment are backward compatible as described in Chapter 4, "System Behavior and Backward Compatibility".

Authentication and authorization plug-ins also need to be recompiled or redesigned after the upgrade, as discussed in "Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins" on page 13-5.

Ensuring Compatibility with Earlier Portal Inserts

Oracle Access Manager 10*g* (10.1.4.0.1) cannot detect query string character encoding and assumes it to be UTF-8. An earlier Identity Server that you upgrade to 10*g* (10.1.4.0.1) has backward compatibility enabled to process Latin-1 data from earlier Portal Inserts. Oracle recommends that you change the encoding of the query string in earlier Portal Inserts from Latin-1 to UTF-8.

> **Note:** If you add a 10*g* (10.1.4.0.1) Identity Server to an upgraded environment, you must manually enable backward compatability with older plug-ins by including the Latin-1 encoding tag in *IdentityServer_install_dir*\identity\oblix\apps\common\bin\oblixpppcatalog.lst. For details, see the *Oracle Access Manager Installation Guide*.

10*g* (10.1.4.0.1) supports two encoding formats for IdentityXML requests: ISO-8859-1 (Latin-1) and UTF-8. The response, however, will be in UTF-8 encoding only. Within this required string you can use a tag to select an encoding specification:

- With new 10*g* (10.1.4.0.1) installations, use the UTF-8 encoding tag (`encoding="UTF-8"`), as shown here.

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  ```

- For backward compatability with older plug-ins in an upgraded environment, use the Latin-1 encoding tag (`encoding="ISO-8859-1"`). For example:

  ```
  <?xml version="1.0" encoding="ISO-8859-1" ?>
  ```

For more information about customizing portal inserts, see the *Oracle Access Manager Customization Guide*.

About Custom Items and Upgrades

Customized .XSL style files, images, and JavaScript files are not migrated during the upgrade. If your previous installation includes significant changes to earlier XSL stylesheets, or if you use a style other than the Oracle Access Manager default Classic Style, you need to manually include those changes in 10*g* (10.1.4.0.1) stylesheets, images, and JavaScript files.

> **WARNING: If you simply copy earlier stylesheets, you may receive stylesheet bug reports or experience unpredictable behavior when using new features designed to work with new stylesheets.**

When you view the Customize Styles page in the upgraded 10*g* (10.1.4.0.1), style names are listed to reflect the style definitions maintained in the directory server as Oracle Access Manager configuration data. However, it is important to note that the customized style files themselves are not migrated, which is why the procedures in this chapter are required.

To illustrate this, suppose a system-level change was made to the 10*g* (10.1.4.0.1) basic.xsl stylesheet to accommodate a new feature. In this case, copying an earlier release of basic.xsl to replace the 10*g* (10.1.4.0.1) basic.xsl will not guarantee that the new feature will work (because the new feature requires the 10*g* (10.1.4.0.1) basic.xsl stylesheet).

> **Note:** As discussed earlier, the directory structure has changed starting with Oracle Access Manager release 6.5 and continuing through 10*g* (10.1.4.0.1) to accommodate multiple languages. Of specific interest for activities in this chapter are the differences in the PresentationXML directories and message storage, described in Appendix A, "Oracle Access Manager Directory Structure Changes".

During the upgrade to 10*g* (10.1.4.0.1)

- Files in the earlier \style0 directory are **replaced**, not migrated.
- The original files are saved in the renamed source directory.

  For example:

  ```
  IdentityServer_install_dir_timestamp\identity\oblix\apps\specific_
  app\ui\style0\name.xsl
  ```

- Any style directories that you have created in the earlier installation are **saved**, not migrated, and are stored in the renamed (backup) source directory created during the upgrade.

> **WARNING: Do not attempt to copy earlier stylesheets to upgraded locations. Instead, you must use procedures in this chapter to alter new stylesheets so they reflect changes in earlier stylesheets.**

The process you must complete to include earlier customized styles in the upgraded 10*g* (10.1.4.0.1) environment differ depending on your starting release. For more information, see the appropriate topic for your environment:

- Incorporating Customizations from Release 6.5 and 7.x
- Incorporating Customizations from Releases Earlier than 6.5

Incorporating Customizations from Release 6.5 and 7.x

Including customized styles from release 6.5 or 7.x in the upgraded 10*g* (10.1.4.0.1) environment is a fairly straight forward process.

> **Note:** Oracle recommends that you locate all recorded changes made in the release 6.5 or 7.x environment before starting and track all operations as you complete them in the 10*g* (10.1.4.0.1) environment.

To incorporate styles created with release 6.5 or 7.x

1. Locate any information about changes and customizations made to the release 6.5 or 7.x environment to use those as a guide when completing the following tasks.
2. **Preserving Custom Styles Directories**: Copy your release 6.5 or 7.x custom language-specific style directories from the renamed source to the 10*g* (10.1.4.0.1) Identity Server language directories. For example:

   From: *IdentityServer_timestamp_install_dir*/identity/oblix/lang/*langtag*/*YourStyleName*

   To: *IdentityServer_install_dir*/identity/oblix/lang/*langtag*/*YourStyleName*

3. **Preserving Custom Images**: Copy your release 6.5 or 7.x (or new) custom images from the renamed source directory to the 10*g* (10.1.4.0.1) WebPass directories. For example:

   From: *WebPass_timestamp_install_dir*/identity/oblix/lang/*langtag*/*style0*

   To: *WebPass_install_dir*/identity/oblix/lang/*langtag*/*style0*

4. **Transferring Stylesheet Customizations**: This is a multiple step process where you inspect individual messages from the release 6.5 or 7.x catalog to the 10*g* (10.1.4.0.1) catalog and manually copy these to the 10*g* (10.1.4.0.1) catalog. For example:

   - **Inspect Earlier Message Changes and Additions**: In your release 6.5 or 7.x renamed source directory, manually inspect any changes to messages in msgctlg.xsl (new messages added or original message was changed) in:

     From: *IdentityServer_timestamp_install_dir*/identity/oblix/lang/*langtag*/msgctlg.xsl

   - **Copy Individual Message Changes**: Manually edit the 10*g* (10.1.4.0.1) msgctlg.xsl file to match the release 6.5 or 7.x version.You may copy information from the release 6.5 or 7.x file into the 10*g* (10.1.4.0.1) version:

     To: *IdentityServer_install_dir*/identity/oblix/lang/*langtag*/msgctlg.xsl

   - **Copy Individual Stylesheet Changes**: In the release 6.5 or 7.x stylesheet in the renamed source directory, identify any changes then edit the 10*g* (10.1.4.0.1) stylesheet files to match the earlier version. You may copy any changes to the 10*g* (10.1.4.0.1) version.

5. **Preserving JavaScript Customizations**: This is a two step process that must be performed for each installed language.

   - **Inspect earlier message changes and additions**: In the release 6.5 or 7.x renamed source directory, identify any JavaScript code changes in the msgctlg.js file, then manually copy these to the 10*g* (10.1.4.0.1) version. For example:

     From: *WebPass_timestamp_install_dir*/identity/oblix/lang/*langtag*/msgctlg.js

     To: *WebPass_install_dir*/identity/oblix/lang/*langtag*/msgctlg.js

   - **Copy individual JavaScript customization**: In the release 6.5 or 7.x renamed source directory, identify any Javascript code changes and then manually copy these to 10*g* (10.1.4.0.1).

For complete details about customized styles, see the *Oracle Access Manager Customization Guide*.

Incorporating Customizations from Releases Earlier than 6.5

If you have upgraded from release 6.5 or later (or your earlier installation did not include custom images, styles, or JavaScript that you want to use with 10*g* (10.1.4.0.1)), you may skip this discussion.

For a successful stylesheet upgrade, you must complete all procedures in this chapter. The stylesheet upgrade task has been divided into several functional procedures that you can use as a guide.

Task overview: Incorporating custom styles includes

1. Completing activities in Style Customization Prerequisites.

2. Recreating Custom Style Directories in 10g (10.1.4.0.1)
3. Customizing New Stylesheets
4. Incorporating Custom Images
5. Using New Customized Styles
6. Incorporating JavaScript Customizations
7. Handling Language-Specific Message Catalogs

Style Customization Prerequisites

Before you begin upgrading stylesheets, check Table 12–1 to ensure you have properly prepared the environment for this task. Failure to complete prerequisites may adversely affect your upgrade.

Table 12–1 Identity Customization Prerequisites Checklist

Checklist	Style Customization Prerequisites Checklist
	Finish Upgrading Remaining Identity System Components and confirm that the upgraded system is working properly
	Review "About Custom Items and Upgrades" on page 12-11.

Recreating Custom Style Directories in 10*g* (10.1.4.0.1)

As you re-create (add) custom style directories to Oracle Access Manager in following steps, you must use the same style names and the same style file system locations that were used before the upgrade. See the *Oracle Access Manager Identity and Common Administration Guide* for additional information.

To add custom styles in 10*g* (10.1.4.0.1)

1. Complete tasks in "Style Customization Prerequisites" on page 12-14.
2. Log in to the upgraded Identity System Console and navigate to the Configure Styles page.

   For example:

   Identity System Console, System Configuration, Configure Styles
3. Enable Classic Style as the default stylesheet if this is not currently the default.
4. Delete the placeholders for your customized styles listed on the Customize Styles page.

   Note: You cannot delete the Classic Style maintained in the \style0 file system directory because this style is required by the Identity System Console.

5. From the Customize Styles page, click the Add Style button.

   The Add Styles page appears.
6. On the Add Style page, fill in the Name and (file system) Directory Name fields using the same style name and file system location used before the upgrade.

   For example:

   Name *Pastel*

Directory Name *Pastel*

The next step enables Oracle Access Manager to create the appropriate file system directory structure automatically and copy the upgraded default stylesheets into it.

7. Select Classic Style in the Copy From list, then click the Save button.

   For example:

   Copy From Classic Style

   Save

   Your new custom style directory duplicates \style0 and contains wrapper stylesheets that point to default global stylesheets in the \shared directory (when you selected Copy From Classic Style).

8. Repeat previous steps 4 through 7, to re-create in Oracle Access Manager each customized style from the earlier installation.

   For additional information, see the *Oracle Access Manager Identity and Common Administration Guide*

   The new style name is listed in the Customize Styles page and one or more directories were created to hold the new wrapper stylesheets.

9. Select a 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.

10. Check your file system for the new style directory name you specified.

You are ready to start the next procedure, "Customizing New Stylesheets" on page 12-15, to include earlier customizations in the new stylesheets.

Customizing New Stylesheets

At this point, you must edit a copy of each new-default stylesheet using your own originally customized files as a guide. It is a good idea to take notes about your work as you go.

Locating and selectively copying stylesheets is an iterative process that you complete one stylesheet at a time, as described in the *Oracle Access Manager Customization Guide*, including:

- Base stylesheets
- Stylesheets *included* in base stylesheets
- Specific function-related stylesheets identified for the program in the application's registration file
- Stylesheets *included* in the function-related stylesheet

To verify that a stylesheet has been successfully applied, just launch the page and perform a visual check.

Task overview: Customizing New Stylesheets

1. Complete the procedure "Recreating Custom Style Directories in 10g (10.1.4.0.1)" on page 12-14.
2. Follow the steps in the procedure "To customize new stylesheets" on page 12-16 to:

- Locate, in the renamed source directory, a customized earlier stylesheet to use as a reference.
- Locate, in your new custom directory, the wrapper stylesheet that corresponds to your earlier customized stylesheet.
- Locate, in your upgraded \shared directory, the default stylesheet that corresponds to the new wrapper.
- Overwrite the new wrapper stylesheet in your new custom style directory with the new-default stylesheet from the 10*g* (10.1.4.0.1) \shared directory.
- Edit the new-default stylesheet using your earlier customized file as a guide.

3. Replace messages in stylesheets, as described in "Handling Language-Specific Message Catalogs" on page 12-20.

See the *Oracle Access Manager Customization Guide* for more information about stylesheet structure and content, and how to customize these.

> **WARNING: Do not copy the original customized file into any upgraded directory. Do not copy content from the original customized file into any upgraded file. Do not attempt to copy all new-default stylesheets into your custom directory at once.**

Remember that your new custom style directory duplicates the default \style0 and contains wrapper stylesheets that point to default stylesheets in the \shared directory. You cannot be assured that a wrapper file will be *called* before the actual stylesheet because both the common registration file and the application's own registration file *call* stylesheets according to an internal ordering.

In addition, the stylesheets in the \shared directory are used with all languages and applications and should be retained as is. Eventually your custom directory will contain a copy of all stylesheets, including those identified in the application's registration file and in oblixbasereg.xml. Even if you do not need to edit a stylesheet, it must be copied to your custom directory.

> **Note:** For the Access System, there are only JavaScript changes; no stylesheets. You must update the stylesheet for each "style" directory for each language. Oracle recommends that you perform these steps to retain all customizations for one language first, then simply copy the updated file to other "style" directories and remaining languages.

To customize new stylesheets

1. In the renamed source directory created during the upgrade, select and open one original customized stylesheet file.

   For example:

   ```
   \IdentityServer_install_dir_
   timestamp\identity\oblix\apps\AppName\ui\style0\name.xsl
   ```

2. In your new custom directory, locate and open the wrapper stylesheet that corresponds to your earlier customized stylesheet.

   For example:

   ```
   \IdentityServer_install_dir\identity\oblix\lang\en-us\Pastel\name.xsl
   ```

3. In the new wrapper stylesheet, review the "include href=" statements for files that are included and record these names and paths.

   For example:

   ```
   <?xml version="1.0" encoding="ISO-8859-1"?>
   - <!--    Copyright (c) 1996-2005, Oracle Inc. All Rights Reserved. -->
   - <xsl:stylesheet version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:oblix="http://www.oblix.com/">
   <xsl:include href="./name.xsl" />
   <xsl:include href="../name.xsl" />
   <xsl:include href="../../shared/name.xsl" />
   </xsl:stylesheet>
   ```

   Next you need to overwrite the wrapper file in your new custom directory with a copy of the new-default stylesheet you intend to customize.

4. In the 10*g* (10.1.4.0.1) \shared directory, locate and copy the new-default stylesheet that corresponds to your original customized stylesheet, as indicated:

   Copy From \shared

   IdentityServer_install_dir\identity\oblix\lang\shared\name.xsl

   Copy To new custom directory

   IdentityServer_install_dir\identity\oblix\lang\en-us*Pastel*\name.xsl

5. In your new custom directory, locate and edit the copied-default stylesheet to reflect changes made to the earlier customized file, and record your changes.
6. Repeat the steps in this list as you locate and copy each related default stylesheet to your new custom directory, then customize it to match changes in the earlier customized release:
   - Base stylesheets
   - Stylesheets *included* in base stylesheets
   - Specific function-related stylesheets identified for the program in the application's registration file
   - Stylesheets *included* in the function-related stylesheet
7. Ensure that file system access control for your new custom style directories and files is set to match the ownership and permissions of \style0.
8. Restart the Identity Server.
9. To verify that a stylesheet has been successfully applied, just launch the page and perform a visual check.
10. Continue with "Incorporating Custom Images" on page 12-17.

Incorporating Custom Images

If your earlier installation did not include custom images that you want to use with 10*g* (10.1.4.0.1), you may skip this discussion.

In earlier versions of Oracle Access Manager, images were distributed throughout the installation directory and referred to with respect to the application path. From 10*g* (10.1.4.0.1) onward, images are language dependent and are consolidated into a single

directory. When installations include multiple languages, you will have multiple *langTag* directories. For directory details, see Appendix A, "Oracle Access Manager Directory Structure Changes".

- **Identity System images** are in the directory:

  WebPass_install_dir\identity\oblix\lang*langTag*\style0

- **Access System images** are in the following directory:

  install_dir\access\oblix\lang*langTag*\style0

> **Note:** All common images require a copy for each language.

gifPathName and jsPathName Variables

Due to the change in location of all image files, a new *gifPathName* variable is defined in wrapper stylesheet style.xsl. In addition to style.xsl, the msgctlg.js file also includes the gifPathName variable to mention the path for image locations:

IdentityServer_install_dir\oblix\lang*langTag*\style0\style.xsl
IdentityServer_install_dir\oblix\lang*langTag*\msgctlg.js

A language independent stylesheet in the \shared directory picks up the images from the modified image path mentioned by the gifPathName variable. This is important for two reasons:

- It prevents hard-coding of URLs in the stylesheets and makes it easier to reuse the same stylesheet across styles.When customizing stylesheets, you should use this global variable whenever constructing a URL path to a GIF or other image.
- It incorporates the current language and current style tag and generates the correct path.

> **Note:** Stylesheets refer to the gifPathName variable to locate the image directory. JavaScript files refer to the jsPathName variable.

For more information about msgctlg files, see "Handling Language-Specific Message Catalogs" on page 12-20.

Example—style.xsl with variables highlighted

The style.xsl wrapper resides in the \style0 directory and can reside in your custom directory:

IdentityServer_install_dir\identity\oblix\lang\en-us\style0\style.xsl
IdentityServer_install_dir\identity\oblix\lang\en-us*Custom*\style.xsl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <!--    Copyright (c) 1996-2005, Oracle Inc. All Rights Reserved. -->
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:oblix="http://www.oblix.com/">
  <xsl:variable name="styleName">style0</xsl:variable>
  <xsl:variable name="localeName">en-us</xsl:variable>
- <xsl:variable name="gifPathName">
  ../../../lang/
  <xsl:value-of select="$localeName" />
  /
  <xsl:value-of select="$styleName" />
```

```
</xsl:variable>
<xsl:variable name="jsPathName">../../../lang/shared</xsl:variable>
...
</xsl:stylesheet>
```

> **Note:** You need to replace the image path in the 10*g* (10.1.4.0.1) stylesheet you are modifying.

For 10*g* (10.1.4.0.1), you need to reference images using the two variables ($gifPathName and jsPathName) to make your customization language and style independent. To do so, modify your 10*g* (10.1.4.0.1) stylesheet with the corresponding <img src="{$gifPathName}/<custom gif name>.gif"/> reference as described in the next procedure.

To incorporate custom images

1. Copy all custom images from the renamed source directory to the target (repeat this for each language):

   From—*source_directory_timestamp*

   To target—*WebPass_install_dir*\identity\oblix\lang*langTag*\style0

2. Copy all custom images for the Access System from the source directory to the target:

   From—*source_directory_timestamp*

   To target—*install_dir*\access\oblix\lang*langTag*\style2

3. Modify the image source path name in 10*g* (10.1.4.0.1) stylesheets in your custom directory:

   For example:

   IdentityServer_install_dir\identity\oblix\lang*Pastel*

4. **For Releases Before 6.5**: Change image path to use the $gifPathName variable.

   For example, suppose the image source is mentioned in the Oracle Access Manager 6.1 stylesheet:

   install_dir\oblix\apps\common\ui\style0\navbar.xsl

   as:

   ```
   <img src="../../common/ui/style0/T1oblixnetpoint.gif"/>
   ```

   You must change the image path for 10*g* (10.1.4.0.1) as follows:

   ```
   <img src="{$gifPathName}/T1oblixnetpoint.gif"/>
   ```

5. See also:

   - Using New Customized Styles
   - Incorporating JavaScript Customizations
   - Handling Language-Specific Message Catalogs

Using New Customized Styles

Before you can use the new customized style, you need to complete the task here.

Note: Due to extensive coverage in the *Oracle Access Manager Customization Guide*. The specific details are outlined but not repeated here.

Task overview: Using new customized styles

1. Copy images and styles to WebPass to create a custom style directory structure on WebPass and include all images in this structure, as described in the *Oracle Access Manager Customization Guide*.
2. Test your customized style, as described in the *Oracle Access Manager Customization Guide*.
3. Propagate new stylesheets to other Identity Servers and WebPass hosts, as described in the *Oracle Access Manager Customization Guide*.
4. Continue with:
   - Incorporating JavaScript Customizations
   - Handling Language-Specific Message Catalogs

Incorporating JavaScript Customizations

If your earlier installation did not include JavaScript customization that you want to use with 10*g* (10.1.4.0.1), you may skip this discussion.

In 10*g* (10.1.4.0.1), JavaScript files are located in:

WebPass_install_dir\identity\oblix\lang\shared

Like stylesheets, language-specific pop-up messages in JavaScript files are replaced with variables defined in:

install_dir\identity\oblix\lang*langTag*\msgctlg.js

JavaScript files are not present in the \lang*langTag* directory except in the msgctlg.js file. The steps needed to migrate JavaScript files are similar to those you used earlier to migrate stylesheet changes.

To incorporate JavaScript files

1. In the time-stamped source directory created during the upgrade, locate your earlier customized JavaScript files on the machine hosting the upgraded WebPass.
2. In the 10*g* (10.1.4.0.1) \lang\shared directory on the WebPass, locate and copy your new-default JavaScript files to retain for future use.
3. Edit the 10*g* (10.1.4.0.1) JavaScript files in the \lang\shared directory on the WebPass to reflect changes made in the earlier release and record your changes, or see "Handling Language-Specific Message Catalogs" on page 12-20.

Handling Language-Specific Message Catalogs

All custom styles are stored under the *langtag* directories, even when your customized functionality is language independent. The procedure here applies to environments that are upgraded from releases prior to 6.5 because these have embedded message customizations in the stylesheet itself. This only applies to 6.1 customers using single language. If the changes are done for English, then the product will pick up the English message properly.

As discussed elsewhere, multiple languages are available for use with 10*g* (10.1.4.0.1). Messages that were once in stylesheets are language dependent and are now defined separately as variables in message catalogs. See also Appendix A, "Oracle Access Manager Directory Structure Changes".

The new Oracle Access Manager directory structure consolidates all message catalogs for JavaScript files, XSL, and HTML.

- As the name suggests any language-specific files will be located in \lang*langTag*.
- Any non-language specific objects are located within \lang\shared.

All the stylesheets have a language-specific wrapper in \lang\langTag\style0 which includes the main language-neutral release stylesheet in \lang\shared. This new wrapper segregates the main stylesheet functionality, which is language independent, from language-specific messages.

Language-specific messages are referred to through variables in message catalog files, as discussed in:

- Handling XSL Stylesheet Messages
- Handling Messages for JavaScript

Handling XSL Stylesheet Messages

The messages for stylesheets are defined in the message catalog:

IdentityServer_install_dir\identity\oblix\lang*langTag*\msgctlg.xsl

You need to ensure that all displayable strings in your earlier stylesheets are placed in the 10*g* (10.1.4.0.1) stylesheet message catalog.

For example, suppose you have customized a Oracle Access Manager 6.1 stylesheet, navbar.xsl, in:

IdentityServer_install_dir\identity\oblix\apps\common\ui\style0\navbar.xsl

where a message reads as:

```
<xsl:text> &lt;&lt; Click here to return to the previous application(s).
</xsl:text>
```

In the 10*g* (10.1.4.0.1) stylesheet:

IdentityServer_install_dir\identity\oblix\lang\shared\navbar.xsl

you need to modify the message to read:

```
<xsl:text> &lt;&lt; <xsl:value-of select="$MPrevAppln"/> </xsl:text>
```

and ensure that MPrevAppln is defined in the 10*g* (10.1.4.0.1) message catalog:

IdentityServer_install_dir\identity\oblix\lang*langTag*\msgctlg.xsl

as follows:

```
<xsl:variable name="MPrevAppln">Click here to return to the previous
application(s). </xsl:variable>
```

To handle language-specific message catalogs for XSL stylesheets

1. Locate the earlier stylesheet containing the customized message.

   For example:

IdentityServer61_install_dir\identity\oblix\apps\common\ui\style0\navbar.xsl

```
<xsl:text> &lt;&lt; Click here to return to the previous application(s).
</xsl:text>
```

If you have already copied the stylesheet to your custom directory, skip to step 3.

2. If you have not yet overwritten the wrapper file with the corresponding stylesheet, copy the corresponding 10*g* (10.1.4.0.1) stylesheet to your custom directory.

   For example:

   Copy from

   IdentityServer_install_dir\identity\oblix\lang\shared\navbar.xsl

   Copy to

   IdentityServer_install_dir\identity\oblix\lang*langTag**Custom_dir*\navbar.xsl

3. In the 10*g* (10.1.4.0.1) stylesheet in your custom directory, modify the message to use the appropriate message catalog parameter.

   For example:

   ```
   <xsl:text> &lt;&lt; <xsl:value-of select="$MPrevAppln"/> </xsl:text>
   ```

4. In the 10*g* (10.1.4.0.1) message catalog, ensure that the message parameter is defined.

   IdentityServer_install_dir\identity\oblix\lang*langTag*\msgctlg.xsl

   ```
   <xsl:variable name="MPrevAppln">Click here to return to the previous
   application(s). </xsl:variable>
   ```

5. Restart the Identity Server and WebPass so the changes take affect.

Handling Messages for JavaScript

Language-specific pop-up messages in JavaScript are also replaced by variables, which are defined in:

*WebPass_install_dir**install_dir*\identity\oblix\lang*langTag*\msgctlg.js

This message catalog is divided into sections that show the messages for specific JavaScript files, several of which are named:

misc.js
...
atickets.js
wfqs.js
deactivateuser.js
confirm.js
...

You need to ensure that all displayable strings are placed in the message catalog, and the message catalog must be referenced through the I18N_GetMsg function.

For example, the code in the earlier JavaScript file:

install_dir\identity\oblix\apps\admin\bin\admin.js

that pops up a message:

```
alert("Room must have a name.")
```

now appears in:

WebPass_install_dir\identity\oblix\lang\shared\admin.js

as:

```
alert(I18N_GetMsg('MRoomNameReq'))
```

where MRoomName is defined in:

*WebPass_install_dir**install_dir*\identity\oblix\lang*langTag*\msgctlg.js

as:

```
MESSAGE_CATALOG[ 'MRoomNameReq' ] = "Room must have a name.";
```

Note: Oracle recommends that you do not customize files in the \shared directory and, instead, copy files from \shared into your custom directory before customizing.

To handle language-specific message catalogs for JavaScript files

1. Ensure that all displayable strings are placed in the 10*g* (10.1.4.0.1) message catalog:

   WebPass_install_dir\identity\oblix\lang*langTag*\msgctlg.js

2. Ensure that the 10*g* (10.1.4.0.1) message catalog is referenced through the I18N_GetMsg function (which is automatically loaded) located in:

   WebPass_install_dir\identity\oblix\lang\shared\i18n.js

3. Restart the Identity Server and WebPass so changes take affect.

Validating Identity System Customization Upgrades

You need to test your upgraded Identity System customizations in a test or development environment before deploying these in an upgraded production environment.

To validate Identity System customization upgrades

1. Verify that your customizations have been restored properly by performing specific operations that will exercise the upgraded customizations. For example, for workflow PPP plug-ins you need to run appropriate workflows.
2. Verify that auditing and access reporting is working properly.
3. Perform a visual inspection of the user interface if you customized any stylesheets and the like.
4. **Upgrade Not Successful**: Proceed to "Recovering from an Identity System Customization Upgrade Failure" on page 12-24.
5. **Upgrade Successful**: Proceed to "Backing Up Upgraded Identity System Customizations" next.

Backing Up Upgraded Identity System Customizations

As mentioned earlier, Oracle recommends that you finish each upgrade by backing up the appropriate 10*g* (10.1.4.0.1) directory. This will enable you to easily restore your environment to the newly upgraded state should that be a requirement.

To back up Identity System customizations after the upgrade

1. Back up the 10*g* (10.1.4.0.1) directory that contains the upgraded Identity System customizations and store it in a new location.
2. Proceed as described in "Looking Ahead" next.

Recovering from an Identity System Customization Upgrade Failure

If an Identity System customization was not successful, you may perform the following steps to rollback this upgrade, then try again.

To recover from an unsuccessful Identity System component upgrade

1. Restore the earlier customization files or directory that you backed up before the upgrade (to recover the earlier customization), then back it up again. You will retain one as a backup copy and use one in the next step.
2. Using a backup copy of your earlier customization files, restart the upgrade as described in this chapter.

Looking Ahead

After ensuring that your previous Identity System customizations are integrated and operating properly in the upgraded environment, see Chapter 13, "Upgrading Your Access System Customizations".

If you do not have an Access System in your environment, proceed to Chapter 14, "Validating the Entire System Upgrade".

13

Upgrading Your Access System Customizations

If your environment does *not* include Access System components, you may skip this chapter. Activities in this chapter are intended for administrators responsible to upgrade and redeploy earlier Access System customizations. Topics include:

- Prerequisites and Guidelines
- Upgrading Auditing and Reporting for the Access Server
- Confirming Access System Failover and Load Balancing
- Upgrading Forms-based Authentication
- Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins
- Associating Release 6.1.1 Authorization Rules with Access Policies
- Assuring Proper Authorization Failure Re-directs After Upgrading from 6.1.1
- Updating the ObAMMasterAuditRule_getEscapeCharacter in Custom C Code
- Validating Access System Customization Upgrades
- Backing Up Upgraded Access System Customizations
- Recovering from an Access System Customization Upgrade Failure
- Looking Ahead

> **Note:** Your installation may not include all items discussed in this chapter. You may skip any task in this chapter that is not relevant for your environment.

Prerequisites and Guidelines

Before starting to upgrade any Access System customizations, Oracle recommends that you:

- Upgrade and redeploy any Identity System customization upgrades, as described in Chapter 12, "Upgrading Your Identity System Customizations".
- Review information in "Customization Upgrade Planning" on page 1-13.
- Back up the directory containing the earlier customization and store it in a new location to help you if you need to roll back to this later.

After completing and testing each upgraded customization, Oracle recommends that you back up the directory containing the upgraded customization and store it in a new location.

Upgrading Auditing and Reporting for the Access Server

As discussed earlier, you need to complete a few activities to ensure that your auditing and access reporting environment is properly set up for Oracle Access Manager 10*g* (10.1.4.0.1). To complete activities for the Access Server, you will use information available in the file:

AccessServer_install_dir\oblix\reports\crystal\audit.sql

The procedures are similar to those you performed for the Identity Server, as outlined next. For details about individual steps within the task here, including uploading the audit schema, see the *Oracle Access Manager Identity and Common Administration Guide*. For general information about the procedures, see "Upgrading Auditing and Access Reporting for the Identity System" on page 12-2.

Task overview: Ugrading auditing and reporting with a Microsoft SQL Server

1. Retain the original database, as is, to preserve your original data.
2. After upgrading all Policy Managers, upgrade the first Access Server (but do not restart the Access Server Service).
3. If you are using an MS SQL database, review information in "Database Record Sizing" on page 12-5.
4. To query or generate any report that requires data from both the old and new database, you need to import the earlier data audited by each Access Server instance into the 10*g* (10.1.4.0.1) database and confirm that it is imported successfully. You will repeat this step for each Access Server instance that you upgrade.

   The `serverId` field in audit table indicates the ID of the Access Server that audited that record. Based on the `serverId` field, it is feasible to differentiate the records audited by each Access Server. The same rule applies to the Identity Servers.

   Note: With an MS SQL database instance, earlier data may be truncated, as described in "Database Record Sizing" on page 12-5. There is no data truncation with an Oracle database instance.

5. Change the DSN (ODBC Data Source Name used by the RDBMS profile of audit & reporting applications) on this machine to refer to the new database instance.

   Note: If you have multiple Access Servers on the same machine, be sure to upgrade all Access Server instances on this machine before you change the DSN to refer to the new database.

6. Start the Access Server service.

   The Access Server will now audit and store data in the new database instance. However, other Access Servers will continue to audit and store data in the old database instance.

7. Upgrade all other Access Server instances as follows:
   - Upgrade the next Access Server instance but do not restart the Access Server service.
   - Repeat step 4 to import data for this Access Server instance.

     Note: If you have multiple Access Servers on the same machine, be sure to upgrade all Access Server instances on this machine before you change the DSN to refer to the new database.

   - Repeat step 5 to change the DSN (ODBC Data Source Name used by the RDBMS profile of the audit & reporting applications) on this machine to refer to the new database instance.
   - Repeat step 6 to restart the Access Server service on this machine.
   - Repeat this step (7) for all Access Servers in your environment.
8. After upgrading all Access Server instances, you complete the rest of the Access System deployment-specific activities in this chapter. You may upgrade WebGates as described in"Upgrading WebGates" on page 10-9.
9. Start auditing, as described in the *Oracle Access Manager Identity and Common Administration Guide*.

Confirming Access System Failover and Load Balancing

If your previous Access System installation was configured for failover or load balancing, it is a good idea to verify that these configurations are still working properly.

During Policy Manager Upgrades: When creating the Directory Server Profile during the incremental upgrade to release 6.5, directory server credentials are read from:

PolicyManager_install_dir/access/oblix/config/userDB.lst

If the configuration tree is in the user directory server *and* under the user node, then the configuration directory profile is **not** created. Otherwise, a configuration directory profile is created using directory server information from:

PolicyManager_install_dir/oblix/config/ldap/configdb.lst

The configuration directory profile is marked for use only by the Policy Manager. Profiles are not created for Policy Manager failover servers. In the case of release 6.1, if the policy tree was on a separate directory server a profile for policy data existed.

During Access Server Upgrades: Profiles are not created for the configuration or policy trees at this time. Before release 6.5, Access System connection pools values for `Initial Connections` and `Maximum Connections` appeared in the UserDB.lst and UserDBFailover.lst. These may not be retained. Also, many .lst files have been transformed into .xml files as part of the globalization effort.

After upgrading Access System Components, it is a good idea to verify the values for `Initial Connections` and `Maximum Connections` in the Database Instance profile of the newly created Directory Server profile.

> **Note:** For concurrent authentication requests on NDS directory servers, Oracle recommends that you increase the connection pool size to something higher than the default (1) for the user directory profile using the System Console.

To confirm failover, load balancing, and connection pool details after the Access System upgrade

1. From the Access System Console, select System Configuration, Server Settings.
2. Under the heading Configure LDAP Directory Server Profiles, select the name of the Profile you want to check.
3. On the Directory Server Profile page, confirm the servers that use the failover information and confirm that the information matches previous settings. For example:
   - Maximum Active Servers
   - Failover Threshold
   - Sleep For (Seconds)
   - Max. Session Time (Min.)
4. Locate the Database Instances list on the Directory Server Profile page and select the name of the Database Instance Profile you want to check.
5. In the Database Instance Profile, verify the values for `Initial Connections` and `Maximum Connections`.
6. Make any changes needed and save the profile.
7. Perform a test to ensure that everything is working as expected.

For more information about configuring failover and load balancing, see the *Oracle Access Manager Deployment Guide*.

Upgrading Forms-based Authentication

As discussed in Chapter 4, "System Behavior and Backward Compatibility", in 10*g* (10.1.4.0.1), form-based authentication supports non-ASCII login credentials (username/password). When you use form-based authentication with 10*g* (10.1.4.0.1) WebGates, you must ensure that character set encoding for the login form is set to UTF-8.

To set the login form encoding to UTF-8 for 10*g* (10.1.4.0.1)

1. Add the following META tag to the HEAD tag of the login form HTML page.

```
<META http-equiv="Content-Type" content="text/html;charset=utf-8">
```

2. If you upgrade an earlier WebGate to 10*g* (10.1.4.0.1), you must also update the login form HTML page after upgrading.

> **Note:** Basic Authentication fails with non-ASCII login credentials. Use form-based authentication for non-ASCII login credentials. Use Basic Authentication with ASCII login credentials.

Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins

Custom Access System plug-ins are copied into the target directory during the Access Server upgrade. However, as discussed earlier, earlier plug-ins send and receive data in Latin-1 encoding.

To send or receive internationalized data you need to re-design plug-ins to use UTF-8 encoding. Also, on Solaris and Linux, release 5.2 and 6.x plug-ins must be re-compiled using the GCC v3.3.2 C++ compiler. For more information, see "Plug-ins" on page 3-10.

> **Note:** Release 7.0 plug-ins as well as earlier plug-ins implemented as executables or those using a scripting language (such as perl) do not require recompiling after the upgrade. However, to send and receive internationalized data, earlier plug-ins should be redesigned to communicate using UTF-8 encoding.

To use authentication and authorization plug-ins

1. Redesign custom authentication and authorization plug-ins to use UTF-8 encoding, if desired.
2. Recompile release 5.2 or 6.x plug-ins on Solaris and Linux platforms using the GCC v3.3.2 compiler.

> **WARNING: You must use the GCC v3.3.2 compiler, regardless of the compiler that may be provided with the Operating System.**

3. Complete any testing to ensure your plug-ins are working properly with 10*g* (10.1.4.0.1).
4. When using plug-ins that send and receive data in Latin-1 encoding, ensure that any new Access Servers added to the upgraded environment are backward compatible as described in Chapter 4, "System Behavior and Backward Compatibility".

Associating Release 6.1.1 Authorization Rules with Access Policies

If you upgraded from release 6.5 or later, you may skip this discussion.

During an upgrade, the names of any release 6.1.1 Authorization Rules move to the Authorization Rules tab of the corresponding policy domain. In addition, the original rule is renamed with a combination of the name of the Policy to which the rule belongs, followed by the Authorization Rule name: *PolicyName_AuthorizationRuleName*.

For example, suppose your 6.1.1 installation includes a Policy Domain (named *MyPolicyDomain*) with two policies (named P1 and P2). And suppose that you have three Authorization Rules associated with these two policies: rules "A1" and "A2" are associated with policy P1, and rule A3 is associated with policy P2. In this case, after the upgrade you will find the following (under the Authorization Rules tab of MyPolicyDomain):

P1_A1

P1_A2

P2_A3

To confirm Release 6.1.1 Policy Domain Authorization Rule names

1. After upgrading from release 6.1.1, navigate to the Policy Manager/Access System Console using the appropriate URL for your environment. For example:

   ```
   http://hostname:port/access/oblix
   ```

   where *hostname* refers to machine that hosts the Web server; *port* refers to the HTTP port number of the WebPass Web server instance; and /access/oblix connects to the Policy Manager and Access System Console.

2. On the Access System landing page, select the Policy Manager link.
3. On the main Policy Manager page, select My Policy Domains on the left side of the page.
4. On the My Policy Domains page, select the link to one of your earlier Policy Domains: *DomainName*.
5. On the domain page, select the Authorization Rules tab.
6. On the Authorization Rules page, look for the renamed rules which are sorted alphabetically.

Assuring Proper Authorization Failure Re-directs After Upgrading from 6.1.1

Each authorization rule in your Policy Domains may include Allow Access and Deny Access conditions. The Allow Access condition of the rule specifies who is authorized to access a protected resource. The Deny Access condition of an authorization rule specifies the end users and groups of users who are explicitly denied access to a resource protected by the rule. If Allow Access or Deny Access conditions (or both) are specified and they do not apply to a user, the user is not qualified by the rule. If a user is unqualified by a rule, by default the user is denied access to the requested resource.

A new authorization state was introduced in release 7.x (apart from authorization success and failure states). This new state is "inconclusive". To accommodate this new state when your earlier installation included authorization failure redirects, you complete the procedure here to specify an explicit Deny rule and to change `Allow takes precedence` to `Yes` on the General panel of the authorization rule.

To reset your Authorization Rule

1. After upgrading from release 6.1.1, navigate to the Policy Manager/Access System Console using the appropriate URL for your environment. For example:

   ```
   http://hostname:port/access/oblix
   ```

   where *hostname* refers to machine that hosts the Web server; *port* refers to the HTTP port number of the WebPass Web server instance; and /access/oblix connects to the Policy Manager and Access System Console.

2. On the Access System landing page, select the Policy Manager link.
3. On the main Policy Manager page, select My Policy Domains on the left side of the page.
4. On the My Policy Domains page, select the link to one of your earlier Policy Domains: *DomainName*.

5. On the domain page, select the Authorization Rules tab.
6. On the Authorization Rules page, look for the renamed rules which are sorted alphabetically and select the rule you want to modify.
7. On the General panel, confirm that `Allow takes precedence` is set to `Yes`.
8. Select the Deny Access panel, then create or modify a rule to specify the users and groups who are denied access to resources protected by this rule (using the People, Role, Rule, and IP Address controls) as indicated in the *Oracle Access Manager Access Administration Guide*.

Updating the ObAMMasterAuditRule_getEscapeCharacter in Custom C Code

If your earlier installation does not use C code created with the Policy manager, that includes the ObAMMasterAuditRule_getEscapeCharacter you may skip this discussion.

An object of the ObAMMasterAuditRule class represents the master audit rule, which specifies global audit parameters and defaults to be used if there is no audit rule specified for a specific policy. In earlier releases, `ObAMMasterAuditRule_getEscapeCharacter` returned the audit escape character. In 10*g* (10.1.4.0.1), the C language API the `ObAMMasterAuditRule_getEscapeCharacter` remains and you may continue using this. However, the audit escape character must be an ASCII character; otherwise the return value is incorrect.

You may need to modify your C code to use the new `ObAMMasterAuditRule_getUTF8EscapeCharacter`, which returns a pointer to the UTF-8 encoded audit escape character.

For more information, see "Policy Manager API" on page 4-28. For details about using the Policy Manager API, see the *Oracle Access Manager Developer Guide*.

Validating Access System Customization Upgrades

You need to test your upgraded Access System customizations in a test or development environment before deploying these in an upgraded production environment.

To validate Access System customization upgrades

1. Verify that your customizations have been restored properly by performing specific operations that will exercise the upgraded customizations.
2. Verify that auditing and access reporting is working properly.
3. **Upgrade Not Successful**: Proceed to "Recovering from an Access System Customization Upgrade Failure" on page 13-8.
4. **Upgrade Successful**: Proceed to "Backing Up Upgraded Access System Customizations" next.

Backing Up Upgraded Access System Customizations

As mentioned earlier, Oracle recommends that you finish each upgrade by backing up the appropriate 10*g* (10.1.4.0.1) directory. This will enable you to easily restore your environment to the newly upgraded state should that be a requirement.

To back up Access System customizations after upgrading them

1. Back up the 10*g* (10.1.4.0.1) directory that contains the upgraded customizations and store it in a new location.
2. When all customizations are completed and redeployed, proceed to Chapter 14, "Validating the Entire System Upgrade".

Recovering from an Access System Customization Upgrade Failure

If an Access System customization was not successful, you may perform the following steps to rollback this upgrade, then try again.

To recover from an unsuccessful Access System component upgrade

1. Restore the earlier customization files or directory that you backed up before the upgrade (to recover the earlier customization), then back it up again. You will retain one as a backup copy and use one in the next step.
2. Using a backup copy of your earlier customization files, restart the upgrade as described in this chapter.

Looking Ahead

After ensuring that your previous Access System customizations are integrated and operating properly in the upgraded environment, see Chapter 14, "Validating the Entire System Upgrade".

Part V

Validating the Upgrade

This part of the book helps you validate the success of the entire upgrade.

Part V contains the following chapters:

- Chapter 14, "Validating the Entire System Upgrade"

14

Validating the Entire System Upgrade

Activities in this chapter should be completed only after upgrading the schema and data, all Identity System and Access System components, integration components and SDKs, and customizations. Topics include:

- Validating the Identity System Upgrade
- Validating Access System Upgrades
- Deleting the Temporary Directory Server Profile
- Reverting Backward Compatibility

Validating the Identity System Upgrade

It is a good idea to quickly validate that you can perform tasks in the Identity System Console and applications. For additional information, see the *Oracle Access Manager Identity and Common Administration Guide*.

To validate your Identity System upgrade

1. Delete all Web browser caches once the upgrade is complete.
2. Make sure your Identity Server service and WebPass Web server instance are running.
3. Navigate to the Identity System Console from your browser by specifying the appropriate URL. For example:

   ```
   http://hostname:port/identity/oblix
   ```

   where *hostname* refers to machine that hosts the Web server; port refers to the HTTP port number of the WebPass Web server instance; /identity/oblix connects to the Identity System Console.

   The Oracle Access Manager landing page should appear.
4. **Landing Page Does Not Appear**: See Chapter F, "Troubleshooting the Upgrade Process".
5. Perform any of the tasks listed next to verify the operation:
   - View the directory server profile for this Identity Server by selecting Identity System Console, System Configuration, Directory Profiles, *link_to_this_profile*
   - Set up panels in the User Manager, Group Manager, Organization Manager.
   - Set up object-based searchbases in the User Manager.

- Set up access controls in the User Manager, Group Manager, or Organization. Manager.
- Create workflow definitions.
- Configure options such as the mail server and session settings.

Validating Access System Upgrades

You can complete any of the next steps to validate that the Access System schema and data upgrade have been successful. For more information, see *Oracle Access Manager Access Administration Guide*.

To verify a successful Access System upgrade

1. Make sure your Policy Manager Web server and WebPass Web server instance are running.
2. Delete all Web browser caches once the upgrade is complete
3. Navigate to the Access System Console from your browser by specifying the appropriate URL. For example:

   ```
   http://hostname:port/access/oblix
   ```

   where *hostname* refers to machine that hosts the Web server; port refers to the HTTP port number of the WebPass Web server instance; /access/oblix connects to the Access System Console.

   The Oracle Access Manager landing page should appear.
4. **Landing Page Does Not Appear**: See Chapter F, "Troubleshooting the Upgrade Process".
5. Log in to the Policy Manager/Access System Console as a Master Administrator.
6. Complete one or more of the following tasks, as described in the latest (10*g* (10.1.4.0.1)) *Oracle Access Manager Access Administration Guide*. For example:
   - Display configuration details for an authentication scheme by clicking the link that corresponds to the scheme.
   - Define or modify a policy domain.
   - Explore the Access System Console.
   - Access a protected resource to confirm that login is working.
7. Log out, as usual.

Deleting the Temporary Directory Server Profile

After upgrading the master Policy Manager (with the schema and data upgrade), an administrator created a temporary directory profile to grant the Access Server write access to policy data stored in the directory server. This temporary directory profile was required when the Access Server gathered configuration information stored in the WebGatestatic.lst file and updated the directory server during WebGate upgrades.

After upgrading *all* earlier WebGates and confirming proper operation of the upgraded WebGates, you may delete the temporary directory server profile.

Note: Do not perform this task until all earlier WebGates in your environment have been upgraded and verified to be working.

To delete the temporary directory server profile

1. From the Access System Console, click the System Configuration tab.
2. Click Server Settings.
3. In the Configure LDAP Directory Server Profiles section, click the check box for the profile that you want to delete.
4. Click Delete.
5. When all earlier custom plug-ins and WebGates have been successfully upgraded and backward compatibility is no longer needed, proceed to "Reverting Backward Compatibility" next.

Reverting Backward Compatibility

You may recall that backward compatibility with earlier custom plug-ins (and WebGates/AccessGates) was enabled during earlier Identity and Access Server upgrades. If 10*g* (10.1.4.0.1) Identity or Access Servers were installed in the upgraded environment, enabling backward compatibility was a manual task.

After upgrading all older plug-ins, WebGates and AccessGates, and confirming that the entire system upgrade has been successful, you may revert backward compatibility.

The steps you complete to revert backward compatibility are similar to those used to manually enable backward compatibility. For more information, see:

- Reverting Identity Server Backward Compatibility
- Reverting Access Server Backward Compatibility

Reverting Identity Server Backward Compatibility

After extending your custom Identity plug-ins to support UTF-8, you perform the steps in the next procedure on every Identity Server in your environment whether backward compatibility was enabled automatically or manually.

To revert backward compatibility on Identity Servers

1. Upgrade all Identity System customizations as described in Chapter 12, "Upgrading Your Identity System Customizations".
2. Redeploy all upgraded Identity System customizations and verify that all are working as expected.
3. Locate and open the Identity Server oblixpppcatalog.lst file in *IdentityServer_install_dir*\identity\oblix\apps\common\bin\oblixpppcatalog.lst.
4. Set the encoding flag from `Latin-1` to `encoding` after the `ApiVersion` flag (if there is one) to provide backward compatibility for Latin-1 data. For example:

   From:

   ```
   userservcenter_view_pre;lib;;..\..\..\unsupported\ppp\ppp_dll\
   ppp_dll.dll;Publisher_USC_PreProcessingTest_PPP_Automation;;Latin-1
   ```

   To:

```
userservcenter_view_pre;lib;;..\..\..\unsupported\ppp\ppp_dll\
ppp_dll.dll;Publisher_USC_PreProcessingTest_PPP_Automation;;encoding
```

5. Repeat as needed for entries in this file.
6. Save the file.
7. Restart the Identity Server service.
8. Repeat for each Identity Server in the upgraded environment to revert backward compatibility.

Reverting Access Server Backward Compatibility

After verifying that your custom Access System plug-ins were redesigned to support UTF-8, and after upgrading all WebGates/AccessGates successfully, backward compatibility is no longer needed. In this case, Oracle recommends that you manually set `"IsBackwardCompatible" Value="false"` in all Access Server globalparams.xml files.

Whether backward compatibility was enabled automatically or manually, you perform the steps in this procedure on every Access Server in your environment.

To revert backward compatibility on Access Servers

1. Upgrade all Access System customizations as described in Chapter 13, "Upgrading Your Access System Customizations".
2. Redeploy all upgraded Access System customizations and verify that all are working as expected.
3. Locate and open the Access Server globalparams.xml file in *AccessServer_install_dir*\access\oblix\apps\common\bin\globalparams.xml.
4. Set `"IsBackwardCompatible" Value="false"`. For example:

```
<SimpleList
     <NameValPair
           ParamName="IsBackwardCompatible"
            Value="false">
     </NameValPair>
</SimpleList>
```

5. Save the file.
6. Restart the Access Server service.
7. Repeat for each Access Server in the upgraded environment.

Part VI

Appendixes

This part of the book provides useful information that falls outside the scope of the main topics covered elsewhere in this manual.

Part VI contains the following appendixes:

- Appendix A, "Oracle Access Manager Directory Structure Changes"
- Appendix B, "Upgrade Process and Utilities"
- Appendix C, "Manual Schema and Data Upgrades"
- Appendix D, "Upgrading Sun Web Server Version 4 to Version 6 on Windows 2000"
- Appendix E, "Planning Worksheets and Tracking Checklists"
- Appendix F, "Troubleshooting the Upgrade Process"

A

Oracle Access Manager Directory Structure Changes

If you started the upgrade process from Oracle Access Manager release 6.5 or 7.x, you may skip this chapter because the directory structure remains the same. However, if you started the upgrade from a release earlier than 6.5, there are important directory structure changes that you need to be aware of.

The installed product directory structure remained constant from Oracle Access Manager release 5.2 to release 6.5. With the introduction of localization for multi-language environments in release 6.5, new directories were added, some directories moved, and some were eliminated. This new directory structure is carried forward with 10*g* (10.1.4.0.1).

Not all new directories reside on all Oracle Access Manager component hosts. This appendix introduces both the earlier directory structure and the new structure.

- About the 10g (10.1.4.0.1) Directory Structure
- Identity Server Directories
- WebPass Directories
- Directories for Access System Components
- PresentationXML Directories

About the 10*g* (10.1.4.0.1) Directory Structure

Starting with the release 6.5, a new directory structure was introduced to accommodate the addition of Language Packs that enable you to display static information to users in their native language. Oracle Access Manager provides a new directory named \oblix\oracle\nlstrl that is created for each component during with the automatic installation of the Oracle National Language Support Library.

The top level directory structure for 10*g* (10.1.4.0.1) looks like the following:

OracleAccessManager\access

OracleAccessManager\identity

OracleAccessManager\webcomponent

In addition, 10*g* (10.1.4.0.1) provides additional Language Packs and support for multibyte character sets such as Japanese and Chinese.

Note: English language messages require no additional Language Pack. All installations include a \lang directory with an \en-us subdirectory for English language messages.

With release 6.5 through 10*g* (10.1.4.0.1), the location of certain files has changed. For example, the location of message files and stylesheets will differ from earlier releases. See these topics for more information:

- \lang Directory and \langtag Subdirectories
- \logs Directory
- \obsymbols Directory
- \reports Directory
- \scoreboard Directory
- \WebServices Directory

The default directory structure for the latest Oracle Access Manager PresentationXML libraries is summarized in the next list. Information here introduces some of these changes, which are explained in detail in the *Oracle Access Manager Customization Guide*:

- *IdentityServer_install_dir*\identity\oblix\apps*AppName*\bin
- *IdentityServer_install_dir*\identity\oblix\lang*langTag*
- *IdentityServer_install_dir*\identity\oblix\lang*langTag*\style0
- *IdentityServer_install_dir*\identity\oblix\lang\shared

- *WebPass_install_dir*\identity\oblix\lang*langTag*
- *WebPass_install_dir*\identity\oblix\lang*langTag*\style0
- *WebPass_install_dir*\identity\oblix\lang\shared
- *WebPass_install_dir*\identity\oblix\WebServices\XMLSchema

\lang Directory and *langtag* Subdirectories

Starting with release 6.5 and continuing forward, Oracle Access Manager installations include a directory named \lang, which includes a named directory (*langtag*) for each installed language. For example, *langtag* en-us contains English-specific directories and files that is included with every installation by default. When you install a Language Pack a *langtag* directory is included and named with a specific language tag. In the example here, the French Language Pack was installed:

IdentityServer_install_dir\identity\oblix\lang\en-us
IdentityServer_install_dir\identity\oblix\lang\fr-fr
IdentityServer_install_dir\identity\oblix\lang\shared

Note: Your installation will be English only unless Oracle-provided Language Packs were installed. You may install Language Packs independently after installing or upgrading, to 10*g* (10.1.4.0.1) as described in the *Oracle Access Manager Installation Guide*.

Each *langTag* subdirectory contains ..XML message catalog files for various applications, which you may customize, as well as other .HTML files. In addition, each \langTag directory contains a \style0 directory.

The \lang\shared directory provides default global stylesheets in all languages. For more information about stylesheets and PresentationXML directories, see "About Custom Items and Upgrades" on page 12-11 and the *Oracle Access Manager Customization Guide*.

> **Note:** In release 6.5 the \engine directory was removed. Also, in release 6.5 the \orig directory was removed, but returned in release 7.0 and remains in 10*g* (10.1.4.0.1).

\logs Directory

This directory contains Oracle Access Manager log files.

\obsymbols Directory

This directory contains .pdb files used for debugging crashes on Windows systems.

\reports Directory

This directory contains a subdirectory for Crystal Reports that includes samples and templates.

\scoreboard Directory

This directory contains the scoreboard files used by SNMP.

\WebServices Directory

On the machine hosting a WebPass, this directory contains subdirectories for Web Services Description Language files; samples; and XMLSchema. For more information, see the *Oracle Access Manager Developer Guide*.

Identity Server Directories

The Identity Server was formerly known as the NetPoint or COREid Server. There are several new directories for the Identity Server. Some are new starting with release 6.5 and continuing through10*g* (10.1.4.0.1) and some are new as of 10*g* (10.1.4.0.1):

IdentityServer_install_dir\identity\oblix

\lang (contains a named directory (*langtag*) for each installed language and \shared)
langtag (for example, en-us, contains message files in a specific language)
\help
\style0 (default wrapper stylesheets specific to each application)
\shared (default global stylesheets for various applications in all languages)
\obsymbols (.pdb files used for debugging crashes on Windows systems)
\oracle (files for Oracle National Language Support)
\reports (Readme file explaining files and contents)
\crystal (Crystal Reports directory)
\samples (Crystal Reports samples)
\templates (Crystal Reports templates)
\scoreboard (Files used by Oracle Access Manager SNMP)

Table A–1 shows the subdirectories and files that are part of 10*g* (10.1.4.0.1) located in *IdentityServer_install_dir*\identity\oblix.

Table A–1 IdentityServer_install_dir\identity\oblix Subdirectories

6.5 to 10*g* (10.1.4.0.1)	Earlier Subdirectory	File type
\apps	\apps	Application subdirectories with related files
\config	\config	Configuration files
\data	\data	Runtime-related configuration files
\data.ldap	\data.ldap	LDAP-related configuration files
\include	\include	Include files for third-party integration
\lang		Contains the following subdirectories: --\shared directory of default global stylesheets --\en-us (*langtag*)language-specific files/directories: -- message files in a specific language --\help directory in a specific language --\style0 (default wrapper stylesheets)
\lib	\lib	Library files
\logs	\logs	Log files
\mail	\mail	Mail files
\obsymbols		.pdb files used for debugging crashes on Windows systems
\oracle		Files for the Oracle National Support Library
\orig	\orig	Copies of the parameter files with default settings Oracle Customer Care or Professional Services can use these files during troubleshooting to determine if customization of a file is causing a problem.
\reports		Crystal reports samples and templates
\scoreboard		Scoreboard files used by SNMP
\tools	\tools	Utility applications (migration_tools and other directories)
\unsupported	\unsupported	Useful tools, utilities, and code examples that have not been tested by Oracle Quality Assurance

WebPass Directories

There are several new directories for WebPass, starting with release 6.5 and continuing through 10*g* (10.1.4.0.1), as shown:

WebPass_install_dir\identity\oblix

\lang (contains language specific subdirectories as well as \java and \shared)
langtag (is *not* an exact duplicate of the one on the Identity Server)
\style0 (copies of default wrapper stylesheets and image files)
\java (resource properties for specified languages)
\shared (default global files that WebPass uses in response to requests)
\logs
\obsymbols (.pdb files used for debugging crashes on Windows systems)
\oracle (files for Oracle National Language Support)
\WebServices
\ samples (Web Services Description Language samples)
\WSDL (Web Services Description Language files)
\XMLSchema (XML schemas that define elements specific to applications)

Table A–2lists the subdirectories and files that are part of 10*g* (10.1.4.0.1) located in *WebPass_install_dir*\identity\oblix:

Table A–2 WebPass_install_dir\identity\oblix Directories

6.5 to 10*g* (10.1.4.0.1) Subdirectory	**Earlier Subdirectory**	**File type**
\apps	\apps	Application subdirectories and files, including the Identity System Administration files
\config	\config	Configuration files.
\lang		Contains the following: --\en-us and other language-specific subdirectories that are *not* an exact duplicate of those on the Identity Server --\java subdirectory --the \shared directory of default global files that WebPass uses in response to requests
\lib	\lib	Library files
\logs	\logs	Log files
\obsymbols		.pdb files used for debugging crashes on Windows systems
\oracle		Files for the Oracle National Support Library
\orig	\orig	Copies of the parameter files with default settings Oracle Customer Care or Professional Services can use these files during troubleshooting to determine if customization of a file is causing a problem.
\tools	\tools	Utility applications (migration_tools and other directories)
\unsupported	\unsupported	Useful tools, utilities, and code examples that have not been tested by Oracle Quality Assurance
\Webservices		XML schema files for specific applications and more
Release 6.5, 7.0 and 10*g* (10.1.4.0.1) Files	**Previous Files**	**Description**
.apacheconfig	.apacheconfig	Directives for Apache Web servers
.nsconfig	.nsconfig	Directives for Sun (formerly Netscape/iPlanet) Web servers to hide files in the Oracle Access Manager system that should not be viewable from a browser
index.htm	index.htm	Startup Web page with .htm extension
index.html	index.html	Startup Web page with .html extension

Directories for Access System Components

The Access System consists of three components (Policy Manager, Access Server, WebGate). The Access System is optional.

Starting with release 6.5 and continuing through 10*g* (10.1.4.0.1), there are several new directories for the Access System components:

PolicyManager_install_dir\access\oblix
AccessServer_install_dir\access\oblix
WebGate_install_dir\access\oblix

The following subdirectories are included for all Access System components:

\lang (contains language specific subdirectories as well as \shared)
langtag (for example, en-us)
\docs (Web server setup details and other docs)

\style2
\obsymbols

The following additional subdirectories are included on the Policy Manager only:

\lang (contains language specific subdirectories as well as \shared)
langtag (for example, en-us)
\help
\shared (.js files)

The following additional subdirectories are included on the Access Server and WebGate for use with certain third-party integrations:

\lang
langtag
\securid-cgi (files for use when integrating RSA SecurID)
\securid-forms (files for use when integrating RSA SecurID)
\securid-forms-adforest (files for use when integrating RSA SecurID)
\securitybridgeforms (files for use when integrating the security bridge)

Subdirectories for the Policy Manager

The Policy Manager was formerly known as the Access Manager component. Not all directories are available on all Access System components. The following subdirectories and files are part of 10*g* (10.1.4.0.1) located in the *PolicyManager_install_dir*\access\oblix:

Table A–3 Policy Manager_install_dir\access\oblix Directories

6.5 to 10*g* (10.1.4.0.1) Subdirectory	Earlier Subdirectory	File type
\apps	\apps	Application subdirectories with related files
\config	\config	Configuration files
\data	\data	Runtime-related configuration files
\data.ldap	\data.ldap	LDAP-related configuration files
\lang		Contains the following subdirectories: --\en-us and other language-specific subdirectories that contain: --\docs (Web server setup docs) --\help --\style2 --\shared (.js files)
\lib	\lib	Library files
\logs	\logs	Log files
\obsymbols		.pdb files used for debugging crashes on Windows systems
\orig	\orig	A copy of all message and parameter files required for future migration to newer versions of Oracle Access Manager
\tools	\tools	Utility applications (migration_tools and other directories)

Subdirectories for the Access Server

Not all directories are available on all Access System components. The following subdirectories and files in 10*g* (10.1.4.0.1) are located in the *AccessServer_install_dir*\access\oblix as:

Table A–4 Access Server_install_dir\access\oblix Directories

6.5-10*g* (10.1.4.0.1) Subdirectory	Earlier Subdirectory	File type
\apps	\apps	Application subdirectories with related files
\config	\config	Configuration files
\data	\data	Runtime-related configuration files
\data.ldap	\data.ldap	LDAP-related configuration files
\engine	\engine	Files used to create and audit messages
\lang		Contains the following subdirectories (also on the WebGate host): --\en-us and other language-specific subdirectories that contain language-specific message catalogs and: --\docs (Web server setup docs) --\help --\securid subdirectories --\securitybridge subdirectory --\style2
\lib	\lib	Library files
\logs	\logs	Log files
\obsymbols		.pdb files used for debugging crashes on Windows systems
\orig	\orig	Copies of the parameter files with default settings Oracle Customer Care or Professional Services can use these files during troubleshooting to determine if customization of a file is causing a problem.
\reports		Crystal Reports samples and templates (Not on WebGate)
\scoreboard		Files used by Oracle Access Manager SNMP (Not on WebGate)
\sdk	\sdk	Software development kit files (Not on WebGate)
\tools	\tools	Utility applications (migration_tools and other directories)

Subdirectories for WebGate

In addition to the directories described in"Subdirectories for the Access Server" on page A-6, the directories here are included and WebGate information is added to the *WebGate_install_dir*\access\oblix:

- _ivmWebGate
- _uninstWebGate

PresentationXML Directories

The next discussions identify changes for Oracle Access Manager stylesheets and messages as follows:

- PresentationXML Directories with Oracle Access Manager Release 6.5 and Later

- PresentationXML Directories Before Oracle Access Manager 6.5
- Message Storage

PresentationXML Directories with Oracle Access Manager Release 6.5 and Later

If you have upgraded from release 6.5 or later (or your earlier installation did not include custom images, styles, or JavaScript that you want to use with 10*g* (10.1.4.0.1)), you may skip this discussion.

Oracle Access Manager default Classic Style stylesheets and the PresentationXML library are now stored as shown here. For more information, see Appendix A, "Oracle Access Manager Directory Structure Changes" and the *Oracle Access Manager Customization Guide*:

IdentityServer_install_dir\identity\oblix\apps*AppName*\bin
IdentityServer_install_dir\identity\oblix\lang*langTag*
IdentityServer_install_dir\identity\oblix\lang*langTag*\style0
IdentityServer_install_dir\identity\oblix\lang\shared

WebPass_install_dir\identity\oblix\lang*langTag*
WebPass_install_dir\identity\oblix\lang*langTag*\style0
WebPass_install_dir\identity\oblix\lang\shared
WebPass_install_dir\identity\oblix\WebServices\XMLSchema

The contents of the default 10*g* (10.1.4.0.1) directories for the Identity Server are outlined in Table A–5. This directory structure was introduced in Oracle Access Manager release 6.5 and continues through 10*g* (10.1.4.0.1).

Table A–5 Default PresentationXML Libraries Release 6.5 and Later

Default Oracle Access Manager Directories	Contents
IdentityServer_install_dir\identity\oblix\apps*AppName**bin* where *AppName* can be common, groupservcenter, objservcenter, userservcenter, and so on.	Registration and parameter files specific to the application.
IdentityServer_install_dir\identity\oblix\lang*langTag* where *langTag* represents an installed language, such as en-us (English) or fr-fr (French).	Message files for various applications.
IdentityServer_install_dir\identity\oblix\lang*langTag*\style0	▪ Wrapper stylesheets for applications point to templates in \shared ▪ Common Oracle Access Manager images
IdentityServer_install_dir\identity\oblix\lang\shared	XSL stylesheet templates for various applications

The contents of the default 10*g* (10.1.4.0.1) WebPass directories identified earlier are outlined in Table A–6. This directory structure was introduced in Oracle Access Manager 6.5 and continues through 10*g* (10.1.4.0.1).

Table A–6 Default WebPass PresentationXML Libraries Release 6.5 and Later

Default WebPass Directories	Contents
WebPass_install_dir\identity\oblix\lang*langTag*	Contains message files for various applications

Table A–6 (Cont.) Default WebPass PresentationXML Libraries Release 6.5 and Later

Default WebPass Directories	Contents
WebPass_install_dir\identity\oblix\lang*langTag**style0*	▪ Image files used in presenting the page ▪ Copies of style0 stylesheets for client-side processing only
WebPass_install_dir\identity\oblix*lang*\shared	▪ JavaScript files ▪ Copies of stylesheets for reference only
WebPass_install_dir\identity\oblix\WebServices\XMLSchema	Contains XML schema files for specific applications

For more information about directories and their content, see the *Oracle Access Manager Customization Guide*.

PresentationXML Directories Before Oracle Access Manager 6.5

If you have upgraded from release 6.5 or later (or your earlier installation did not include custom images, styles, or JavaScript that you want to use with 10*g* (10.1.4.0.1)), you may skip this discussion.

The PresentationXML library was provided under two directories and distributed depending upon how the files were likely to be used. For example, stylesheets that define the default Oracle Access Manager Classic Style are maintained in flat files in the file system directory:

IdentityServer_install_dir\identity\oblix\apps*AppName*

For example:

IdentityServer_install_dir\identity\oblix\apps\userservcenter\ui\style0*style_name*.xls

The pre-6.5 Identity Server directory structure *IdentityServer_install_dir*\identity\oblix\apps*AppName* (common, groupservcenter, and so on) is summarized in Table A–7.

Table A–7 Pre-6.5 Identity Server PresentationXML Libraries

\bin	\ui	\xmlschema
Dynamically-loadable code for the application, and the registration file, message file(s) and parameter files specific to the application	Stylesheets for the application (in one or more style directories)	XML schema files specific to the application

The pre-6.5 WebPass directory structure is summarized in Table A–8. For example, *WebPass_install_dir*\identity\oblix\apps*AppName* (common, groupservcenter, and so on):

Table A–8 Pre-6.5 WebPass PresentationXML Libraries

\bin	\ui
JavaScript files	Stylesheets and GIFs specific to the application (in one or more style directories).

For more information, see the *Oracle Access Manager Customization Guide* for your earlier release.

Message Storage

Prior to release 6.5, Oracle Access Manager messages were controlled by an XML file for a specific application. For example:

IdentityServer_install_dir/identity/oblix/apps/*appname*/bin/*appnamemsg*.xml

where *IdentityServer_install_dir* is the directory where the Identity Server is installed and *appname* matches a specific application, as follows:

groupservcenter--Group Manager

objservcenter--Organization Manager

userservcenter--User Manager

Each *appnamemsg*.xml file contained multiple paired sets of data, in the form:

```
<Message MsgTag="the tag name">The tag text</Message>
```

In 10*g* (10.1.4.0.1), these message files now reside in specific language directories. For example: *IdentityServer_install_dir*/identity/oblix/lang/*langTag*/oblixbasemsg.xml.

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

B

Upgrade Process and Utilities

This chapter provides information about the utilities that are called into operation during the upgrade process.

Topics in this chapter include:

- About Upgrade Events
- Primary Utility: obmigratenp
- File Upgrade: obmigratefiles
- Message and Parameter Upgrade: obmigrateparamsg
- Schema Upgrade: obmigrateds
- Data Upgrade: obmigratedata
- Web Server Upgrade: obmigratews
- Component-Specific Upgrades

> **Note:** Running the tools manually is not recommended. Oracle strongly recommends that you complete the upgrade as described in Chapter 9, "Upgrading Remaining Identity System Components" and Chapter 10, "Upgrading Access System Components".

About Upgrade Events

As discussed earlier, the term upgrade refers to the process of changing from one major release (X) of Oracle Access Manager to a later major release (Y) of Oracle Access Manager. The term migration is typically used for the process you must complete to push a Oracle Access Manager implementation from a test environment into a production environment. Despite these distinctions, you will see the term migration used in path names, tools (also known as utilities), on-screen messages, and in discussions that follow.

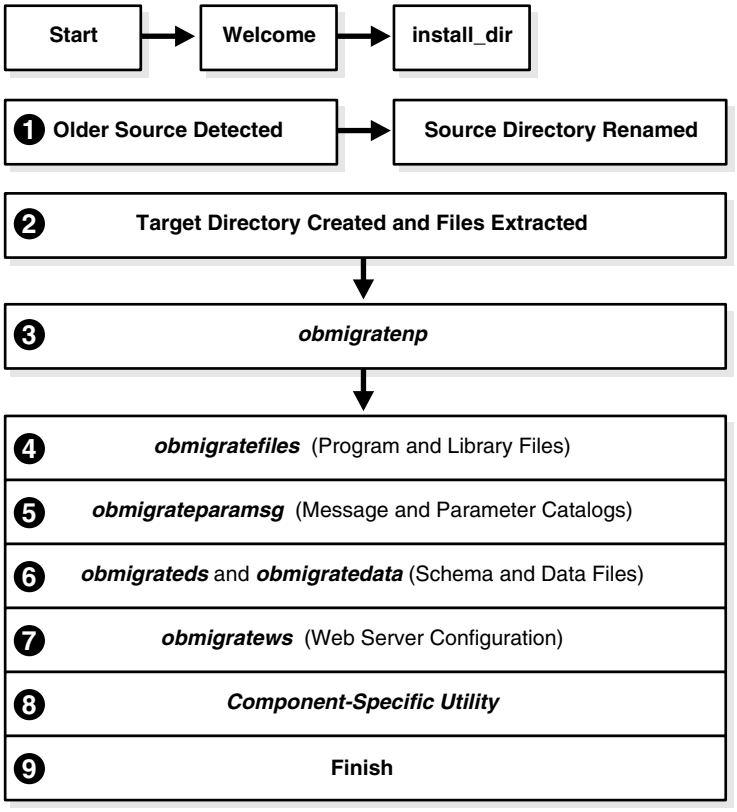
If directory names include spaces, a program may not be invoked properly unless you include quotation marks around each path name in any command you use. For example:

```
obmigratenp.exe -c ois -f 650 -t 700 -s
"C:\Program Files\NetPoint\identity_20060519_134931"
-d "C:\Program Files\NetPoint\identity"
-i "C:\Program Files\NetPoint\identity"
```

WARNING: If your directory names include spaces, be sure to include quotation marks around each path name in any command you use.

Figure B–1 illustrates the process and highlights a number of utilities used and discussed in this chapter. See the process overview that follows the figure for more information.

Figure B–1 Events that Occur When You Initiate an Upgrade



Process overview: When an earlier source is detected and you choose to upgrade

1. The source directory is renamed with a time stamp.
2. The target directory is created and 10*g* (10.1.4.0.1) files are extracted into it. The latest release must be extracted to the original target path, and languages for the previous installation are detected.

   The English language is upgraded automatically. If 10*g* (10.1.4.0.1) Language Packs are available in the source directory you can upgrade earlier languages and add new languages.

> **Note:** If you upgrade a multi-language implementation without 10*g* (10.1.4.0.1) Language Packs, you will lose the multi-language functionality.For more information about multi-language implementations, see Chapter 4, "System Behavior and Backward Compatibility".

3. The obmigratenp utility is called (which determines the release you are migrating from as well as the release you are migrating to) and it internally detects which features need to be upgraded for this particular component and which other utilities to use for those upgrades.

   When your installation includes multiple languages, obmigratenp:

   a. Migrates message catalogs in the default language.

   b. Migrates message catalogs in other selected languages when you have 10*g* (10.1.4.0.1) Language Packs in the component source directory.

   c. Invokes general upgrades, as discussed in "Primary Utility: obmigratenp" on page B-5.

   > **Note:** For details about each utility and the log file generated by each utility, see later discussions in this appendix.

4. The obmigratefiles utility is called to upgrade program and library files.

   a. Required files are extracted to the target directory.

   b. Required configuration and SSL-related files are copied from the renamed source directory to the target directory, and 10*g* (10.1.4.0.1) is installed.

   For more information, see "File Upgrade: obmigratefiles" on page B-5.

5. The obmigrateparamsg utility is called to upgrade message and parameter catalog files.

   a. Required (.xml and .lst) files are identified in renamed source area.

   b. Files are modified for the latest release and written to the target directory. With 10*g* (10.1.4.0.1), .LST files are converted to .XML files and customizations made to the originals are retained in the upgraded files.

   For more information, see "Message and Parameter Upgrade: obmigrateparamsg" on page B-8.

6. The obmigrateds and obmigratedata utilities are called to initiate schema or data upgrades when needed:

   Schema Upgrade Utility: obmigrateds

   a. Determines the type of directory servers configured for user, configuration, and policy data, and bind information.

   b. Uploads new attributes, object classes, and user-related data based on directory server type and release.

   c. Updates user entries for the ChallengeResponsePhrase value with RC6 encryption scheme, if needed.

   For more information, see "Message and Parameter Upgrade: obmigrateparamsg" on page B-8.

Data Upgrade Utility: obmigratedata

a. Connects to the target directory server.

b. Reads and processes existing configuration and policy data using mapping information in the map files.

c. Applies the mapping data and upgrades object class and attribute mapping based on existing data in the old release.

For more information, see "Data Upgrade: obmigratedata" on page B-13

7. The obmigratews utility is called to perform a selective Web server configuration file and filter upgrade, if needed, to accommodate changes for newer versions of Policy Manager, WebPass, and WebGate.

   Changes are added to the Web server configuration file.

   For more information, see "Web Server Upgrade: obmigratews" on page B-15.

8. A component-specific utility is used to make changes to related registry entries for Windows, plug-ins, and other files.

   For example:

   a. **Identity Server:** obMigrateNetPointOis upgrades existing registry entry for the Identity Server to reflect the newer release; modifies PPP catalog if needed; modifies password from password.xml and .lst, if needed; re-creates proper uninstall_info.txt. For details, see "Identity Server: obMigrateNetPointOis" on page B-16.

   Note: The password written in the Oracle Access Manager 5.2, password.xml and password.lst files is not encrypted; however, later versions encrypt this. Encryption occurs automatically during an upgrade.

   b. **WebPass:** obMigrateNetPointWP upgrades existing registry entry for the WebPass to reflect the newer release; modifies password from password.xml and password.lst, if needed. For details, see "WebPass: obMigrateNetPointWP" on page B-17

   c. **Policy Manager:** obMigrateNetPointAM upgrades registry entry for Policy Manager; modifies password encryption from password.lst, if needed; copies your custom plug-ins to the target installation directory from your renamed source directory. For details, see "Policy Manager: obMigrateNetPointAM" on page B-18.

   d. **Access Server:** obMigrateNetPointAAA upgrades registry entry for Access Server; modifies password encryption, if needed; copies your custom plug-ins from your renamed source directory to the target installation directory; upgrades the following failover files:

      AppDB.lst—converted to .xml

      ConfigDB.lst—converted to .xml

      Group.lst—if present, converted to .xml

      UserDB.lst—if present, converted to .xml

      WebResrcDB.lst—converted to .xml

   For more information, see "Upgraded Items" on page 3-5.

e. **WebGate:** obMigrateNetPointWG upgrades registry entry for the WebGate; modifies password encryption, if needed. For details, see "WebGate: obMigrateNetPointWG" on page B-19.

f. **SDK:** obMigrateNetPointASDK is called by obmigratenp to accomplish an Access Manager SDK upgrade.

The SDK upgrade will be invoked automatically as the last step when upgrading components bundled with SDK (Identity Server and the Oracle Access Manager Connector for WebSphere).

Note: If you decline the automatic SDK upgrade, current SDK configuration settings are not preserved and you must reconfigure SDK using the configureAccessGate tool, as described in the *Oracle Access Manager Access Administration Guide*

For details, see "Software Developer Kit (SDK): obMigrateNetPointASDK" on page B-19.

9. Finish as you would any installation.

For details about what must be handled manually, see "Items that You Must Manually Upgrade" on page 3-8.

Note: Upgrades occur incrementally. This means that the sequence of earlier processes begins and the earlier release is upgraded to the next-major release. Following the component-specific upgrade, the process may repeat automatically until all changes between your original release and the latest release are completed.

If you cancel an upgrade after being informed that the component has been installed, you need to complete the following steps to restore your Oracle Access Manager configuration to the original state.

Primary Utility: obmigratenp

The main utility driving an upgrade is obmigratenp. This utility orchestrates the entire upgrade process for a component from a given major release X to a given major release Y, as described in Table B–1.

Table B–1 The Upgrade Driver obmigratenp

Description	Function
obmigratenp.exe	▪ Decides and executes any intermediate incremental steps needed to reach a given target release for the component. ▪ Invokes other utilities to carry out functions to upgrade the specific component from major release X to major release Y
Path	*Component_install_dir*\identity|access\oblix\tools\migration_tools\obmigratenp
Command Line	Run `obmigratenp.exe` without any parameters. This command prints usage along with the meaning of all input parameters.

Table B–1 (Cont.) The Upgrade Driver obmigratenp

Description	Function

Other Files Used	■ Invokes Language Pack extraction to upgrade the default language, determine which additional languages to upgrade and which will not be upgraded.
	■ Reads the message catalog, to print messages to the console or while writing to a log file:
	`_install_dir\identity|access\oblix\tools\migration_tools\obmigratenpmsg.xml`
	■ Reads the parameter file, which includes a section for every component and every x to y upgrade:
	`_install_dir\identity|access\oblix\tools\migration_tools\obmigratenpparams`
	You can specify whether you want to have a certain type of upgrade for that component by setting flags to "true" or "false" to invoke or skip that function, respectively. When a flag is absent in this file, its value is presumed to be false.
	Flags include:
	■ kMigrateWS decides whether `obmigratews.exe` is executed.
	■ kMigrateData and kMigrateSchema determines if `obmigrateds.exe`. Setting either value to true invokes `obmigrateds.exe`.
	■ kMigrateASDK decides whether re-invocation of `obmigratenp.exe` is called for the Access Manager SDK upgrade.

Output	This utility drives the overall upgrade process by invoking various utilities and generating the log files described next.
Log File	Generates the log file: `install_dir\identity|access\oblix\tools\migration_tools\obmigratenp.log`, which typically contains: ■ Component name, source, and target directory ■ Command line used to invoke each upgrade utility ■ Return status of each upgrade utility ■ Other error and informative messages

File Upgrade: obmigratefiles

obmigratefiles is called by the obmigratenp multiple times to carry out file and folder related upgrades.

File upgrades involve copying required files from the renamed source directory to the target installation directory. The obmigratenp tool calls the obmigratefiles tool, which works on a given map file that specifies:

- The files to be copied
- From which source
- To which target

For more information, see the next process overview.

Process overview: obmigratenp calls obmigratefiles

1. obmigratefiles creates a folder of original files in the source directory in the two following circumstances, because the release 5.2.0 installer (and the 6.0.0 installer on Solaris) does not create a folder of original files:
   - When you are upgrading from Oracle Access Manager 5.2.0
   - When you are upgrading on Solaris from release 6.0.0

The map file that is used is *component_Version*_orig_files.lst. For example:

```
ois_520_orig_files.lst
```

or

```
ois_600_orig_files.lst
```

This folder is further used for the message and parameter upgrade.

2. obmigratefiles creates a folder of original files for the current release in the current installation directory using the map file

   *component_Version*_orig_files.lst.

   For example:

   ```
   ois_600_orig_files.lst
   ```

   The next time an upgrade occurs from this release to a newer release, the folder of original files for this release will be available to use during the message and parameter upgrade.

3. obmigratefiles copies config files, SSL setup-related files, and the like from the renamed source directory to the target installation directory.

   In this case, obmigratefiles works on a given *component*_base_files. For example:

   ```
   ois_base_files.lst
   am_base_files.lst
   ```
   and so on

   Base files contain the list of configuration files required for the upgrade. Typically, configuration files do not change. Files and directories in the base file are copied during the upgrade, including failover-related files. Any file and directory clean up occurs as needed. For example, if a particular Oracle Access Manager release deletes a file or introduces additional files, these will be treated appropriately. Suppose a file added in Oracle Access Manager 6.0 is not required in 6.5. In this case, the file will be deleted from the later installation.

   a. Upgrade all files listed in the base file.

   b. For all source versions from the base file release to the current source release, upgrade files listed in *component_source-version_files*.lst files.

   For example, consider upgrading from Oracle Access Manager 5.2. release-specific files exist for Oracle Access Manager 6.0, 6.5, and 7.0. In this case, step 2 copies files listed in ois_610_files.lst, ois_650_files.lst, and ois_700_files.lst. However, when upgrading from Oracle Access Manager 7.0 to 10*g* (10.1.4.0.1), the current source release is the base file release, therefore step 2 doesn't occur.

   Note: In case a deleted file exists in the base file, the deleted file will be removed from the base file list itself. Even if it existed in the earlier Oracle Access Manager release, it is no longer needed in the later release.

Additionally, release-specific files contain changes specific to only a particular *component_source-version*_files.lst; information that needs to be copied if you are upgrading from that release and there are some deviations. For example, suppose a file is added in Oracle Access Manager 6.0 and 6.5. In this case, you need files for:

```
ois_600_files.lst
```

```
ois_650_files.lst
ois_700_files.lst
```

> **Note:** If Oracle Access Manager release 6.1 did not require any changes, ois_610_files.lst is not required.

Table B–2 provides more information

Table B–2 File Upgrades with obmigratefiles

Description	Function
obmigratefiles.exe	▪ Reads a given file for a specific component. ▪ Copies files from the source directory to the target directory according to the list specified in the file. ▪ Processes release-specific files as needed.
Path	*Component_install_dir*\identity|access\oblix\tools\migration_tools\obmigratefiles
Command Line	Run obmigratefiles.exe without any parameters. This command prints usage along with the meaning of all input parameters. **Options include:** -m Specifies name of map file to use -s [*source_dir*] Specifies the source directory. -d [*target_dir*] Specifies the target directory. -i Specifies the install directory. -l Specifies the language for Message migration. -p Specifies the flag for Language Packs.
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *_install_dir*\identity|access\oblix\tools\migration_tools\obmigratefilesmsg.xml
Output	This utility copies files based on input parameters.
Log File	Generates the log file: install_dir\identity|access\oblix\tools\migration_tools\obmigratefiles.log, which typically contains: ▪ Parameters passed to this utility. ▪ Status of every copy-instruction mentioned in used map file. ▪ Error messages, if any

Message and Parameter Upgrade: obmigrateparamsg

The obmigratenp utility calls the obmigrateparamsg utility with the required file for a specific component.

The Message Upgrade Process: Allows you to upgrade earlier messages with new messages and add new messages for the later release. A customized message will be retained. However, if the number of parameters in the message has changed, only the new message is retained. For example:

Original Message—`"Cannot copy file %1"`

Customized Message—`"Failed copy operation for file %1"`

New Message—`"Cannot copy file %1 from %2 to %3"`

Note: In the examples shown here, the new message is retained.

The Parameter Upgrade Process: Parameter upgrades occur in parameter files. When you have modified parameters in your earlier release of Oracle Access Manager and the new release has modified the same parameter, the obmigrateparamsg utility overwrites the earlier changes. See the log file for changes.

Earlier Oracle Access Manager Versions: Include files named as *component_Fromrelease_to_Torelease_msg* | param.lst in the directory *component_install_dir*\identity | access\oblix\tools\migration_tools. For example:

```
ois_520_to_600_msg.lst
ois_520_to_600_param.lst
```

Use of obmigrateparamsg was an iterative process with upgrades occurring for each incremental release.

Note: As mentioned previously, earlier during the upgrade from release 7.0 to 10*g* (10.1.4.0.1), .LST files are converted to .XML files and stored in the target directory. Customizations made in earlier .LST catalogs are preserved and appear in the .XML version of the file. No separate manual step is required to preserve customizations.

Oracle Access Manager 10*g* (10.1.4.0.1): The migration of parameter and message catalogs is performed in a single process. 10*g* (10.1.4.0.1) includes files named *component_release*_param_files.lst and *component_release*_msg_files.lst. For example:

```
am_700_param_files.lst
am_700_msg_files.lst
```

Optional hidden parameters from the earlier release are copied into the target. Hidden parameters are those which Oracle Access Manager supports and which you may want to add.

With Oracle Access Manager 10*g* (10.1.4.0.1), a path within the _param | msg_files.lst file includes a language ID to handle the multi-language feature available as of Oracle Access Manager 6.5. This looks like the example here:

```
file:/oblix/lang/%lang%/frontpagemsg.xml
```

When you specify the -p option, the obmigrateparamsg tool upgrades only message catalogs of the specified languages. The installer detects the language/ language of earlier release according to the following decision and pass it to obmigratenp:

- **For 5.2, 6.0 & 6.1**–Look into globalparams.xml file for the language tag. For example, language:En_US.
- **For 6.5 and Later**–Look in obnls.xml for the list of languages.

Table B–3 provides additional information about obmigrateparamsg.

Table B–3 Message and Parameter Upgrades with obmigrateparamsg

Description	Function
obmigrateparamsg.exe	▪ Reads a given file for a specific component. ▪ Processes given message/parameter files in the .xml file. For every message/parameter file, obmigrateparamsg: ▪ Reads the old release message/parameter file from the renamed source directory. ▪ Modifies the message/parameter file as needed. ▪ Writes the modified file to the target directory where the new installer has extracted files.
Path	*Component_install_dir*\identity|access\oblix\tools\migration_tools\obmigrateparamsg
Command Line	Run obmigrateparamsg.exe with the following parameters: obmigratparamsg -s source_dir -d *target_install_dir* -f *component_oldversion*_param_files.lst -t *component_newversion*_param_files.lst -l <langids> [-p] **Where** -s *source_dir* identifies the installation directory of the earlier Oracle Access Manager release. -d *target_install_dir* identifies installation directory of latest release of Oracle Access Manager. **Note:** This command is executed twice, first for the message catalogs (with the -l option), then for the parameter catalogs (*without* the -l option). The *target_install_dir* can be on a different machine from *source_dir*. -f component_oldversion_param_files.lst -t component_newversion_param_files.lst -l <*language*> (-l is to be specified only for message migration) [-p] Signifies the message catalog upgrade is to happen only for files under the /lang/*langTag* folder. To facilitate the migration of only message catalogs of the specified languages, this is used by Language Pack installers.
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *install_dir*\identity|access\oblix\tools\migration_tools\obmigrateparamsgmsg.lst
Output	This utility upgrades message/parameter files. The log contains all parameters forcefully overwritten/retained.
Log File	Generates the log file: *install_dir*\identity|access\oblix\tools\migration_tools\obmigrateparamsg.log, which typically contains: ▪ Parameters passed to this utility ▪ Name of every parameter/message file mentioned in input file and actions taken on this file. For example, replaced the existing parameter/message value with new value, added/deleted parameter/message, and so on. ▪ Error messages, if any. ▪ Note: When you modify a parameter/message in the earlier release and the current release includes a new parameter/message, you may want to look at these values because obmigrateparamsg has made decisions that you should be aware of.

As discussed in "Mime_types -related Customizations Not Retained" on page F-8, when upgrading from Oracle Access Manager 6.0, multiple entries with the same ParamName in mime_types (.xml and .lst) files are *not* upgraded:

IdentityServer_install_dir/identity/oblix/apps/admin/bin/mime_types.xml
IdentityServer_install_dir/identity/oblix/apps/admin/bin/mime_types.lst

WebPass_install_dir/identity/oblix/apps/admin/bin/mime_types.xml
WebPass_install_dir/identity/oblix/apps/admin/bin/mime_types.lst

Note: Both versions of the file are needed. You may remove MIME types that are no longer needed or add new MIME types to be associated with the particular attribute for further use. Simply edit the mime_types.lst and .xml files for the Identity Server, then copy these into the *WebPass_install_dir* to replace the earlier version.

Schema Upgrade: obmigrateds

Typically, the Oracle Access Manager schema is enhanced for each major Oracle Access Manager release. For example, when Identity Server functionality is enhanced it may refer to a greater number of schema attributes and object classes than previous versions.

During your upgrade, any differences between an earlier schema release and the next release are uploaded to your directory server using the required schema ldif file for your specific directory server. Every schema ldif file includes entries to modify the schema based on the difference between two versions. Schema ldif files use the following naming convention.

```
DataType_fromrelease_to_torelease_schema_DsType.ldif
```

For example:

```
osd_650_to_700_schema_ad.ldif
policy_650_to_700_schema_nds.ldif
```
and so on.

and reside in the directory with various upgrade map files:

```
Component_install_dir\identity|access\oblix\tools\migration_tools
```

During the upgrade, the obmigratenp utility reads the file obmigratenpparams.lst and calls obmigrateds to internally upload schema files when the kMigrateData kMigrateSchema flag is set to true in:

```
Component_install_dir\identity|access\oblix\tools\migration_tools
\obmigratenpparams.lst
```

Schema upgrades occur incrementally. This means that the earlier release is upgraded to the next-latest release, the resulting schema is upgraded to the next-latest release, and so on until all interim schema changes between your original release and the latest release are completed. Obsolete schema elements are deleted during the upgrade.

A schema upgrade may occur only with Oracle Access Manager components that interface with the directory server: Identity Server, Policy Manager, and Access Server. Table B–4 provides more information about obmigrateds.

Table B–4 Schema Upgrades with obmigrateds

Description	Function

obmigrateds.exe	▪ Reads configuration files, assesses schema data (OSD), and determines possible directory servers with which Oracle Access Manager is communicating. For example, the directory server containing configuration data, the directory server containing user data, and the directory server containing policy data.
	▪ Gathers the information required to connect and bind to those directory servers.
	▪ Locates the schema file for the specific data type, directory type, and the from and to versions, then uploads the appropriate ldif file to the directory server using the ds_conf_update.exe utility
	▪ Using information read from the OSD (for example, 'o=oblix, ..'node) and configuration files, obmigrateds creates an input map file to be passed to obmigratedata.exe.
	For example:
	`data_fromrelease_to_torelease_osd.lst -- for osd, policy, and workflow upgrades`
	`data_fromrelease_to_torelease_user.lst -- for user data upgrade`
	▪ obmigrateds upgrades configuration data using obmigratedata, which creates an output data file, then deletes the Oracle Access Manager configuration tree from the directory and uploads this output data file to the directory server. For more information, see "Data Upgrade: obmigratedata"
	▪ obmigrateds upgrades user data using obmigratedata.
	Note: Starting with release 6.0, and later, the upgrade includes user entries for the "ChallengeResponsePhrase" value with an RC6 encryption scheme. Earlier Oracle Access Manager versions used an RC4 encryption scheme for the same purpose.

Path	Component_install_dir\identity|access\oblix\tools\migration_ tools\obmigrateds
Command Line	'Run obmigrateds.exe without any parameters. This command prints usage along with the meaning of all input parameters
Other Files Used	▪ Reads the message catalog here to print messages to the console or while writing to a log file: install_ dir\identity|access\oblix\tools\migration_ tools\obmigratedsmsg.lst ▪ Reads the parameter file obmigratedsparams.lst, gathers data, and determine which flags are set and which type of upgrade to perform: `install_dir\identity|access\oblix\tools\migration_ tools\obmigratedsparams.lst.` Note: obmigratedsparams includes a section for every component. Within every section is a subsection for the upgrade from x to y. ▪ For example, the obmigratedsparams section for 'ois' contains a subsection '520_to_600' that contains flags that determine which of the following upgrades to complete: – osd/user schema upgrade – osd/policy/user data upgrade Each subsection also includes path and filenames (LST or XML) from which obmigrateds can get details about directory servers with OSD, policy data, or user data.
Output	This utility carries out schema and data migration by invoking appropriate utilities.
Log File	Generates the log file: *install_dir*\identity|access\oblix\tools\migration_ tools\obmigrateds.log

Data Upgrade: obmigratedata

When the newer release of Oracle Access Manager includes a new Oracle Access Manager-specific data organization or values, data upgrades occur in much the same way as the schema upgrade. The delta between the old and new versions is determined and the appropriate data ldifs are provided so they can be uploaded to the directory server. An incremental upgrade is performed between each major release and the next major release until you have completed the upgrade. After the upgrade, Oracle Access Manager can identify and use the data present in the directory and run smoothly.

A data upgrade may occur only with Oracle Access Manager components that interface with the directory server: Identity Server, Policy Manager, and Access Server.

Files that contain both object-class and attribute mappings are provided. The object and attribute mapping files reside in:

```
install_dir\identity|access\oblix\tools\migration_tools\obmigratedata
```

The object-class mapping filename is oc_*Fromrelease*_to_*Torelease*_map.lst. For example:

```
oc_520_to_600_map.lst
oc_610_to_650_map.lst
oc_650_to_700_map.lst
```

Note: There is no data migration from Oracle Access Manager 6.0.0 release 6.1.0. For this reason, there is no oc_600_to_610_map.lst file.

The attribute mapping filenames are appear as:

at_*Fromrelease*_to_*Torelease*_map_DataType.lst. For example:

```
at_520_to_600_map_osd.lst—Oblix schema data
at_520_to_600_map_policy.lst—NetPoint policy data
at_520_to_600_map_user.lst—User data
at_520_to_600_map_wf.lst—Workflow data
```

as well as files for 600 to 650 and 650 to 700 and 700 to 10*g* (10.1.4.0.1). For example:

```
at_600_to_650_map_item.lst
at_650_to_700_map_item.lst
```

where item refers to osd, policy, user, or workflow attribute mapping files.

The obmigrateds utility invokes obmigratedata for data upgrading and passes a map file with initial information—OSD directory, bind DN, password, personoc, groupoc, and the like—to obmigratedata. This map file uses the naming convention:

```
data_Fromrelease_to_Torelease_osd.lst
data_Fromrelease_to_Torelease_user.lst
```

For example:

```
data_520_to_600_osd.lst
data_520_to_600_psc.lst
data_610_to_650_osd.lst
data_610_to_650_psc.lst
data_650_to_700_osd.lst
data_650_to_700_psc.lst
data_700_to_1014_osd.lst
```

```
data_700_to_1014_psc.lst
```

This is because the upgrade is carried out in steps. For example, if you start from release 520, data is first upgraded from 520 to 600, then from 610 to 650, from 650 to 700, and finally from 700 to 10*g* (10.1.4.0.1).

> **Note:** There is no data upgrade between release 600 and 610. Starting from release 5.2, you upgrade first to release 6.1.1 using 6.1.1 installers; then you complete the upgrade from 6.1.1 using 10*g* (10.1.4.0.1).

See Table B–5 for more information.

Table B–5 Data Upgrades with obmigratedata

Description	Function
obmigratedata.exe	▪ Accepts a file giving basic required information as input for the target directory server (connectivity details, person and group object classes, and so on). The input file instructs the utility about the file used to obtain object-class mapping. **Note:** The object class mapping file identifies the file to be used for attribute-level mapping. ▪ Reads mapping files, connects to given directory, reads existing data, processes this data based on instructions in the object-class and attribute-mapping files, and creates an output ldif. **Note:** All mappings file must be present in *install_dir*\identity|access\oblix\tools\migration_tools\obmigratedata
Path	Component_install_dir\identity|access\oblix\tools\migration_tools\obmigratedata
Command Line	Run obmigratedata.exe with the following parameters. obmigratedata -f ConfigFileName -i install_dir where ConfigFileName is the full path of a file that provides all initially required information so this utility can connect to a given directory server. Additionally, this file contains other information such as the object-class mapping file name, log file name, name of the file giving a list of binary attributes, and so on. Also: *install_dir* is the target installation directory for this component.
Other Files Used	▪ The object class mapping file defined in the input config file ▪ The attribute mapping file(s) as mentioned in the object class mapping file ▪ The file listing binary attributes (file name is mentioned in the input config file) ▪ The message catalog obmigratedatamsg.lst, while printing to the console or writing to a file: *install_dir*\identity|access\oblix\tools\migration_tools\obmigratedata\obmigratedatamsg.lst
Output	This utility creates an output ldif file whose name is mentioned in the input config file.

Table B–5 (Cont.) Data Upgrades with obmigratedata

Description	Function
Log File	Generates the log file: *install_dir*\identity|access\oblix\tools\migration_tools\obmigratedata The name of the log file is mentioned in the input config file. For example, migration_log_file.lst, which typically contains: ▪ Old and new DNs of entries migrated by this utility ▪ Success/failure messages for selected migration ▪ Other error messages if any

Web Server Upgrade: obmigratews

With Policy Manager, WebPass, and WebGate enhancements may come the need for changes in the supporting Web server configuration file. During the upgrade process, you are asked about automatic Web server configuration file updates. Oracle recommends that you update the Web server configuration file automatically, though you may do this manually following the upgrade.

The obmigratenp utility calls obmigratews to complete the Web server configuration update by passing a map file and other parameters to obmigratews. The map file is named and located as follows:

*Component_fromrelease_to_torelease*_ws_WebserverType.lst
install_dir\identity|access\oblix\tools\migration_tools

For example:

am_520_to_600_ws_nsapi.lst

Also, obmigratenp copies the file generated by obmigratews to the original Web server configuration file. Thus the Web server configuration file gets the required changes for the newer release of the component. See Table B–6 for more information.

Table B–6 Web Server Configuration Upgrades with obmigratews

Description	Function
obmigratews.exe	▪ Reads the input map file, modifies content of this file using input values of old and new installation directories. ▪ Modifies the given input Web server configuration file according to the content created using the map file. ▪ Writes the Web server configuration to a new output file whose name is passed to this utility as one of the parameters.
Path	Component_install_dir\identity|access\oblix\tools\migration_tools\obmigratews
Command Line	Run obmigratews without any parameters. This command prints usage along with the meaning of all input parameters.
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *install_dir*\identity|access\oblix\oblix\tools\migration_tools\obmigratewsmsg.lst
Output	This utility creates a modified release of the Web server configuration file.

Table B–6 (Cont.) Web Server Configuration Upgrades with obmigratews

Description	Function
Log File	Generates the log file: *install_dir*\identity|access\oblix\tools\migration_tools\obmigratews.log

Component-Specific Upgrades

Every component needs special treatment during the upgrade to accommodate specific registry changes, modifying specific files, and the like. As a result, `obmigratenp.exe` calls the appropriate utility depending upon the component selected for the upgrade. Typical actions taken during this sequence include:

- Copy/modify specific files
- Modify existing component specific registry entry
- Copy specific plug-ins

Each component-specific utility is described as follows:

- Identity Server: obMigrateNetPointOis
- WebPass: obMigrateNetPointWP
- Policy Manager: obMigrateNetPointAM
- Access Server: obMigrateNetPointAAA
- WebGate: obMigrateNetPointWG
- Software Developer Kit (SDK): obMigrateNetPointASDK

Identity Server: obMigrateNetPointOis

To accomplish a Identity Server upgrade, the obmigratenp tool calls the obMigrateNetPointOis tool. See Table B–7 for more information.

Table B–7 Identity Server Upgrade with obMigrateNetPointOis

Description	Function
obMigrateNetPointOis.exe	▪ Upgrades the existing registry entry for the Identity Server to reflect the newer Oracle Access Manager release. ▪ Modifies the PPP catalog file, if required, to ensure it is usable with the newer Oracle Access Manager release. ▪ Encrypts the password written in password.xml, when upgrading from Oracle Access Manager 5.2. ▪ Deletes install_dir\identity\oblix\tools\setup\uninstall_info.txt, if present, and creates it again with proper information on Windows systems.
Path	install_dir\identity\oblix\tools\migration_tools\obMigrateNetPointOis
Command Line	Run obMigrateNetPointOis without any parameters. This command prints usage along with the meaning of all input parameters.
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *install_dir*\identity\oblix\oblix\tools\migration_tools\obMigrateNetPointOismsg.lst

Table B–7 (Cont.) Identity Server Upgrade with obMigrateNetPointOis

Description	Function
Output	■ A new flag (`encoding`) is added to the oblixpppcatalog.lst file automatically to ensure backward compatability with earlier plug-ins. A backward-compatible Identity Server continues to send data to earlier plug-ins in Latin-1 encoding (earlier plug-ins will set data in Latin-1 encoding; new plug-ins will set data in UTF-8 encoding). ■ Modifies the registry entry for the Identity Server. ■ Modifies the PPP catalog file. ■ Modifies password.xml. ■ Creates proper uninstall_info.txt.
Log File	Generates the log file: *install_dir*\identity\oblix\tools\migration_tools\obMigrateNetPointOis.log. Typically this file contains: ■ Parameters passed to this utility. ■ Status of each action taken by this utility.

WebPass: obMigrateNetPointWP

To accomplish a WebPass upgrade, obmigratenp calls obMigrateNetPointWP. See Table B–8 for more information.

Table B–8 WebPass Upgrade with obMigrateNetPointWP

Description	Function
obMigrateNetPointWP.exe	■ Upgrades the existing registry entry for the WebPass to reflect the newer Oracle Access Manager release. ■ When upgrading from Oracle Access Manager 5.2, encrypts the password written in password.xml. ■ On Windows systems, deletes the file uninstall_info.txt, if present, and creates it again with proper information: *install_dir*\identity\oblix\tools\setup\uninstall_info.txt
Path	*install_dir*\identity\oblix\tools\migration_tools\obMigrateNetPointWP
Command Line	Run obMigrateNetPointWP without any parameters to print usage and the meaning of all input parameters.
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *install_dir*\identity\oblix\tools\migration_tools\obMigrateNetPointWPmsg.lst
Output	■ Modifies the registry entry for the WebPass. ■ Modifies password.xml.
Log File	Generates the log file: *install_dir*\identity\oblix\tools\migration_tools\obMigrateNetPointWP.log

Policy Manager: obMigrateNetPointAM

To accomplish an Policy Manager (formerly known as the Access Manager component) upgrade, obmigratenp calls obMigrateNetPointAM. See Table B–9 for more information.

Table B–9 Policy Manager Upgrade with obMigrateNetPointAM

Description	Function
obMigrateNetPointAM.exe	▪ Upgrades the existing registry entry for the Policy Manager to reflect the newer release. ▪ When upgrading from release 5.2, encrypts the password written in password.lst. ▪ Modifies install_dir\access\oblix\data\common\ldapuserdbparams.lst if required. ▪ Copies custom plug-ins from renamed source directory to target directory
Path	*install_dir*\access\oblix\tools\migration_tools\obMigrateNetPointAM
Command Line	Run obMigrateNetPointAM without parameters to print usage.
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *install_dir*\access\oblix\oblix\tools\migration_tools\obMigrateNetPointAMmsg.lst
Output	▪ Modifies the registry entry for the Policy Manager. ▪ Modifies password.xml. ▪ Copies custom plug-ins to target directory.
Log File	Generates the log file: install_dir\access\oblix\tools\migration_tools\obMigrateNetPointAM.log

Access Server: obMigrateNetPointAAA

To accomplish an Access Server upgrade, the obmigratenp utility calls the obMigrateNetPointAAA utility. See Table B–10 for more information.

Table B–10 Access Server Upgrade with obMigrateNetPointAAA

Description	Function
obMigrateNetPointAAA.exe	▪ Upgrades the existing registry entry for the Policy Manager to reflect the newer release. ▪ When upgrading from release 5.2, encrypts the password written in password.lst. ▪ Modifies install_dir\access\oblix\data\common\ldapuserdbparams.lst if required. ▪ Copies custom plug-ins from renamed source directory to target directory
Path	*install_dir*\access\oblix\tools\migration_tools\obMigrateNetPointAAA
Command Line	Run obMigrateNetPointAAA without parameters to print usage.
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *install_dir*\access\oblix\oblix\tools\migration_tools\obMigrateNetPointAAA.lst

Table B–10 (Cont.) Access Server Upgrade with obMigrateNetPointAAA

Description	Function
Output	▪ A new parameter "IsBackwardCompatible" Value="true" is set in the Access Server globalparams.xml file automatically. A backward-compatible Access Server continues to send (and receive) data to earlier custom authentication and authorization plug-ins in Latin-1 encoding (earlier custom plug-ins will set data in Latin-1 encoding; new plug-ins will set data in UTF-8 encoding). ▪ Modifies the registry entry for the Access Server. ▪ Modifies password.xml. ▪ Copies custom plug-ins to target directory.
Log File	Generates the log file: `install_dir\access\oblix\tools\migration_tools\obMigrateNetPointAAA.log`

WebGate: obMigrateNetPointWG

To accomplish an WebGate upgrade, obmigratenp calls obMigrateNetPointWG. See Table B–11 for more information

Table B–11 WebGate Upgrade with obMigrateNetPointWG

Description	Function
obMigrateNetPointWG.exe	▪ Upgrades the existing registry entry for the WebGate to reflect the newer Oracle Access Manager release. ▪ Encrypts the password written in password.lst, when upgrading from Oracle Access Manager 5.2.
Path	*install_dir*`\access\oblix\tools\migration_tools\obMigrateNetPointWG`
Command Line	Run obMigrateNetPointWG without parameters to print usage.
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *install_dir*`\access\oblix\tools\migration_tools\obMigrateNetPointWGmsg.lst`
Output	▪ Modifies the registry entry for the WebGate. ▪ Modifies password.lst.
Log File	Generates the log file: *install_dir*`\access\oblix\tools\migration_tools\obMigrateNetPointWG.log`

Software Developer Kit (SDK): obMigrateNetPointASDK

To accomplish a Software Developer Kit upgrade, obmigratenp calls obMigrateNetPointASDK. See Table B–12 for more information.

Table B–12 SDK Upgrade with obMigrateNetPointASDK

Description	Function
obMigrateNetPointASDK.exe	▪ Upgrades the existing registry entry for the Access Manager SDK to reflect the newer release. ▪ Encrypts the password written in password.lst, when upgrading from release 5.2.
Path	*install_dir*`\access\oblix\tools\migration_tools\obMigrateNetPointASDK`

Table B–12 (Cont.) SDK Upgrade with obMigrateNetPointASDK

Description	Function
Command Line	Run as `obmigrateAccessSDK -fromver <oldVer> -tover <newVer> -srcdir <dir> -dstdir <dir>`
Other Files Used	To print messages to the console or while writing to a log file, reads the message catalog: *install_dir*`\oblix\tools\migration_tools\obMigrateNetPointASDKmsg.lst`
Output	▪ Modifies the registry entry for the Access Manager SDK. ▪ Modifies password.lst.
Log File	Generates the log file: *install_dir*`\oblix\tools\migration_tools\obMigrateNetPointASDK.log`

C

Manual Schema and Data Upgrades

This chapter provides information about the tools and utilities that are called into operation during the upgrade process.

Topics in this chapter include:

- About Upgrading Schema and Data Manually
- About Upgrading Schema and Data Manually
- About Upgrading Data Manually
- Upgrading Data Manually
- Sample Default obmigratenpparams.lst File
- Sample data_520_to_600_xxx.lst

> **See also:** Appendix B, "Upgrade Process and Utilities"

About Upgrading Schema and Data Manually

Oracle recommends that you upgrade the schema and data automatically, as described in Part II, "Upgrading the Schema and Data". However, errors occur during the upgrade process, you may need to use these instructions to perform an upgrade from one release to another manually.

Completing manual upgrades of the schema and data includes the use of specific utilities provided by Oracle Access Manager. For more information about these utilities, see Appendix B, "Upgrade Process and Utilities".

In examples in this chapter, release 5.2 is mentioned for illustration only. If you are starting the upgrade from a release earlier than 6.1.1, see "Indirect Upgrade Paths" on page 1-26.

Upgrading the Schema Manually

When upgrading your schema manually, you need to select the appropriate schema file for your directory server and the specific release you are upgrading from and to, as shown in Table C–1.

Note: You may notice gaps in the from and to releases in Table C–1. This occurs when there is no schema or data update for the specific release. For example, there were no schema changes from release 6.0.0 to 6.1.0 and as a result there are no files named ...600_to_610_ schema_ ....

Table C–1 Schema Files

Directory Type	Schema Files
Active Directory	osd_520_to_600_schema_ad.ldif policy_520_to_600_schema_ad.ldif user_520_to_600_schema_ad.ldif osd_610_to_650_schema_ad.ldif policy_610_to_650_schema_ad.ldif user_610_to_650_schema_ad.ldif osd_650_to_700_schema_ad.ldif policy_650_to_700_schema_ad.ldif user_650_to_700_schema_ad.ldif osd_700_to_1014_schema_ad.ldif policy_700_to_1014_schema_ad.ldif user_700_to_1014_schema_ad.ldif
ADAM	osd_650_to_700_schema_adam.ldif policy_650_to_700_schema_adam.ldif user_650_to_700_schema_adam.ldif osd_700_to_1014_schema_adam.ldif policy_700_to_1014_schema_adam.ldif user_700_to_1014_schema_adam.ldif
IBM SecureWay	osd_520_to_600_schema_ibm.ldif policy_520_to_600_schema_ibm.ldif user_520_to_600_schema_ibm.ldif osd_610_to_650_schema_ibm.ldif policy_610_to_650_schema_ibm.ldif osd_650_to_700_schema_ibm.ldif policy_650_to_700_schema_ibm.ldif user_650_to_700_schema_ibm.ldif osd_700_to_1014_schema_ibm.ldif policy_700_to_1014_schema_ibm.ldif user_700_to_1014_schema_ibm.ldif
Novell e-Directory	osd_520_to_600_schema_nds.ldif policy_520_to_600_schema_nds.ldif user_520_to_600_schema_nds.ldif osd_610_to_650_schema_nds.ldif policy_610_to_650_schema_nds.ldif osd_650_to_700_schema_nds.ldif policy_650_to_700_schema_nds.ldif user_650_to_700_schema_nds.ldif osd_700_to_1014_schema_nds.ldif policy_700_to_1014_schema_nds.ldif user_700_to_1014_schema_nds.ldif
Oracle Internet Directory	osd_700_to_1014_schema_oid.ldif policy_700_to_1014_schema_oid.ldif user_700_to_1014_schema_oid.ldif
Siemens DirX	osd_700_to_1014_schema_dirx.ldif policy_700_to_1014_schema_dirx.ldif user_700_to_1014_schema_dirx.ldif

Table C–1 (Cont.) Schema Files

Directory Type	Schema Files
Sun 4.x and 5.x	osd_520_to_600_schema_ns.ldif policy_520_to_600_schema_ns.ldif user_520_to_600_schema_ns.ldif osd_610_to_650_schema_ns.ldif policy_610_to_650_schema_ns.ldif osd_650_to_700_schema_ns.ldif policy_650_to_700_schema_ns.ldif user_650_to_700_schema_ns.ldif osd_700_to_1014_schema_ns.ldif policy_700_to_1014_schema_ns.ldif user_700_to_1014_schema_ns.ldif
Oracle Virtual Directory Server (VDS)	user_700_to_1014_schema_vde.ldif

To upgrade the schema manually

1. Navigate to the upgrade directory:

   ```
   IdentityServer_install_dir\identity\oblix\tools\migration_tools
   ```

2. Locate the appropriate schema file for the directory server and specific *fromrelease_to_release*.

3. Invoke the schema upgrade tool `ds_conf_update` using the command here.

   For example:

   ```
   IdentityServer_install_dir\identity\oblix\tools\ldap_tools
   \ds_conf_update -f schema_file
   ```

   where *schema_file* is the name of the schema file as shown in the samples in Table C–1.

4. Respond to prompts for various options, such as host, port, userid, and password.

   Any errors produced while running this tool are printed to stdout. This tool can be used for any directory server that is compatible with Oracle Access Manager 10*g* (10.1.4.0.1).

About Upgrading Data Manually

Oracle Access Manager includes several files that are called and used during the upgrade from Oracle Access Manager 5.2 to 10*g* (10.1.4.0.1). You may copy and use these files as a template to display or suppress the prompts you see and respond to during the upgrade. For example, you can allow the prompts to enable automatic data upgrades or you can suppress the prompts to enable manual upgrade.

The template in "Sample Default obmigratenpparams.lst File" on page 12 determines which data upgrade prompts you see during the upgrade from Oracle Access Manager 5.2 to 10*g* (10.1.4.0.1).

```
\IdentityServer_install_dir\identity\oblix\tools
\migration_tools\obmigratenpparams.lst
```

Included in this file are sections for upgrades from each major release to the next major release starting with release 5.2 and continuing through release 10*g* (10.1.4.0.1). A

complete, ***annotated*** version of the obmigratenpparams.lst file is shown in "Sample Default obmigratenpparams.lst File" on page 12.

WARNING: The order of parameters in the file may not indicate the upgrade order.

The appropriate value must be supplied for each parameter in the file. For True/False values:

- A value of True triggers the automatic data upgrade prompt and program.
- A value of False suppresses the data upgrade prompt and program and can be used when you want a manual data upgrade.

WARNING: Although not recommended, you may suppress automatic upgrades of the Oracle Access Manager configuration data and user data, as described in "Suppressing Automatic Data Upgrades" on page 5. In that case, you must manually upgrade the configuration data and user data as described in "Upgrading User Data Manually" on page 10.

Upgrading Data Manually

Oracle recommends that you upgrade the schema and data automatically. However, when you must upgrade the schema and data manually use the overview here as a guide.

Task overview: Upgrading data manually includes

1. Suppressing Automatic Data Upgrades.
2. Upgrading the Configuration Tree Manually.
3. Removing Obsolete Schema Elements for Release 6.5 and 7.0, if you want.
4. Uploading the Generated LDIF.
5. Upgrading User Data Manually.

WARNING: Oracle recommends that you upgrade the schema and data automatically.

When you choose manual data upgrades, two files provide parameters for the configuration data and user data in the directory:

install_dir\identity\oblix\tools\migration_tools\obmigratedata

data_520_to_600_osd.lst—drives the manual upgrade of the Oracle Access Manager configuration data, for example:

```
osd_migration:true
policy_migration:true
wf_migration:true
user_migration:false
```

Note: A value of True upgrades data. A value of False suppresses the upgrade.

data_520_to_600_user.lst—drives the upgrade of user data, which occurs only during the upgrade from Oracle Access Manager 5.2.x to 6.0.0, for example:

```
osd_migration:false
policy_migration:false
wf_migration:false
user_migration:true
```

Both the osd.lst and user.lst files contain similar information. However, procedures must be completed for both user data and configuration data and a specific value must be provided for each parameter in the files. See Table B–7 on page B-16. An annotated example is shown in "Sample data_520_to_600_xxx.lst" on page C-16. See also, "Data Upgrade: obmigratedata" on page B-13.

In addition to the two .lst files mentioned earlier, Oracle Access Manager 10*g* (10.1.4.0.1) provides the following files:

- **data_610_to_650_multi_lang.lst**—multi_lang_migration:true
- **data_610_to_650_osd.lst**—osd_migration:true
- **data_610_to_650_psc.lst**—psc_migration:*true*
- **data_650_to_700_osd.lst**—osd_migration:true and wf_migration:true
- **data_650_to_700_psc.lst**—psc_migration:*true*
- **data_700_to_1014_osd.lst**—osd_migration:true
- **data_700_to_1014_psc.lst**—psc_migration:true

Suppressing Automatic Data Upgrades

If you intend to upgrade data manually, you need to suppress automatic data upgrades.

To suppress automatic data upgrades

1. Copy the file obmigratenpparams.lst and rename the original to retain it.

   For example:

   ```
   IdentityServer_install_dir\identity\oblix\tools
   \migration_tools\obmigratenpparams.lst
   ```

2. Edit the `ois` section of the copy and set the values shown in bold, next, to "false" to prohibit automatic data upgrades.

   For example:

   ```
   ois
   BEGIN:vCompoundList
    520_to_600:
    BEGIN:vNameList:
     kMigrateData:false
     kMigrateSchema:false
     kMigratePublisher:false
   ```

Note: See "Sample Default obmigratenpparams.lst File" on page 12 for sections on individual upgrades from one major release to the next. For example, from 520_to_600, and so on.

3. Complete a manual data upgrade as described in "Upgrading the Configuration Tree Manually" on page 6.

Upgrading the Configuration Tree Manually

You begin upgrading data manually by upgrading the configuration data tree. The following commands provide an example only. Your environment may vary.

To upgrade the configuration tree manually

1. Copy the data_fromrelease_to_release_osd.lst file and rename it to retain the original.

   For example:

   From

   ```
   \install_dir\identity\oblix\tools\migration_tools\obmigratedata
   \data_520_to_600_osd.lst
   ```

   To

   ```
   \install_dir\identity\oblix\tools\migration_tools\obmigratedata
   \config_data_520to600_osd.lst
   ```

2. Edit the file to provide the information for your environment based on the annotated sample in "Sample data_520_to_600_xxx.lst" on page 16.

3. Confirm that the configuration tree and data parameter values are true.

   ```
   osd_migration:true
   policy_migration:true
   wf_migration:true
   ```

4. Confirm that the user data migration parameter value is false.

   ```
   user_migration:false
   ```

5. Back up (export) the old configuration tree to an LDIF.

6. Locate the obmigratedata tool, then upgrade the configuration tree by running obmigratedata and specifying the name of your updated file. For example:

   ```
   \IdentityServer_install_dir\identity\oblix\tools
   \migration_tools\obmigratedata\obmigratedata.exe
   run obmigratedata.exe -f config_data_520to600_osd.lst -I install_dir
   ```

   This program generates an LDIF based on the options selected in FunctionalityTBMigrated. The resulting LDIF file will be named as specified by the outputFileName parameter.

7. Delete the existing configuration tree after the OSD upgrade completes.

8. **From 6.5 to 10*g* (10.1.4.0.1)**—Before continuing with step 9 when upgrading the Identity Server and Policy Manager, you may proceed to "Removing Obsolete Schema Elements for Release 6.5 and 7.0" next.

9. **All Upgrades**—Complete the upgrade of configuration data with:

- Uploading the Generated LDIF
- Upgrading User Data Manually

Removing Obsolete Schema Elements for Release 6.5 and 7.0

Oracle provides cleanup files for use during the incremental upgrade sequence between release 6.5 and 7.0. There are no cleanup files for any directory servers for the incremental upgrade sequence between release 7.x and 10*g* (10.1.4.0.1).

You may skip this procedure if your starting release is 7.0 or if you do not want to remove the obsolete schema from the directory server.

During an upgrade from release 6.5 to 10*g* (10.1.4.0.1), you can use the following procedures to clean up the obsolete schema elements from the directory server. If you choose to do this, it must be done after the configuration tree is deleted in the manual upgrade flow:

- During Identity Server upgrades
- During Policy Manager upgrades

After schema cleanup, LDIF files are available for you to upload to the directory server. Schema files for the configuration schema and the policy schema are separate. As a result, there are two scenarios to consider when you clean up the obsolete schema:

- Are configuration and policy data stored in same directory server?
- Are configuration and policy data stored in different directory servers?

Table C–2 shows configuration data cleanup files for specific directory server types.

Table C–2 Configuration Data Cleanup Files

Directory Type	Schema Cleanup Files
Active Directory	osd_650_to_700_schema_delete_ad.ldif
ADAM	osd_650_to_700_schema_delete_adam.ldif
IBM SecureWay	osd_650_to_700_schema_delete_ibm.ldif
Novell e-Directory	osd_650_to_700_schema_delete_nds.ldif
Oracle Internet Directory	Support for Oracle Internet Directory was introduced with Oracle COREid release 7.0.4 (also available as part of Oracle Application Server 10g Release 2 (10.1.2)). Therefore, there are no obsolete schema entries to be deleted and no such files for Oracle Internet Directory.
Sun 4.x 5.x **Note**: 10*g* (10.1.4.0.1) does *not* support Sun 4.x directory servers. See "Sun Directory Server Considerations and Preparation" on page 5-14.	osd_650_to_700_schema_delete_ns.ldif

Table C–3 shows the policy data cleanup files for specific directory server types.

Table C–3 Policy Data Cleanup Files

Directory Type	Schema Cleanup Files
Active Directory	policy_650_to_700_schema_delete_ad.ldif
ADAM	policy_650_to_700_schema_delete_adam.ldif
IBM SecureWay	policy_650_to_700_schema_delete_ibm.ldif

Table C–3 (Cont.) Policy Data Cleanup Files

Directory Type	Schema Cleanup Files
Novell e-Directory	policy_650_to_700_schema_delete_nds.ldif
Oracle Internet Directory	Support for Oracle Internet Directory was introduced with Oracle COREid release 7.0.4 (also available as part of Oracle Application Server 10g Release 2 (10.1.2)). Therefore, there are no obsolete schema entries to be deleted and no such files for Oracle Internet Directory.
Sun 4.x 5.x **Note**: Oracle Access Manager 10*g* (10.1.4.0.1) does *not* support Sun 4.x directory servers. See "Sun Directory Server Considerations and Preparation" on page 5-14.	policy_650_to_700_schema_delete_ns.ldif

Depending upon your directory server and starting release, you may complete the procedures here more than once:

- Cleaning Up Obsolete Elements During Identity Server Upgrades
- Cleaning Up Obsolete Elements During Policy Manager Upgrades

Cleaning Up Obsolete Elements During Identity Server Upgrades

Use the following procedure to remove obsolete elements during Identity Server upgrades from release 6.5 to 7.0.

To remove obsolete elements during Identity Server upgrades

1. Navigate to the upgrade directory:

   ```
   IdentityServer_install_dir/identity/oblix/tools/migration_tools
   ```

2. Locate the appropriate file for your directory server, as specified in Table C–2, so that you can provide this name in step 3.

3. Take the appropriate action for your environment:

   - **Same Directory Server**—If configuration data and policy data are in the same directory server, invoke the schema upgrade tool `ds_conf_update` using the command:

     ```
     IdentityServer_install_dir/identity/oblix/tools/ldap_tools
     /ds_conf_update -f schema_file
     ```

     where *schema_file* is the name of the schema cleanup file from Table C–2.

   - **Different Directory Server**—If configuration and policy data are in different directory servers, invoke `ds_conf_update` twice: once for the configuration schema and a second time for the policy schema as follows:

     ```
     IdentityServer_install_dir/identity/oblix/tools/ldap_tools
     /ds_conf_update -f identity/oblix/tools/migration_tools
     ```

     where *schema_file* is the name of the appropriate schema cleanup file from Table C–2.

4. Respond to prompts for various options, such as host, port, userid, and password.

5. Continue the upgrade with "Uploading the Generated LDIF" on page 9.

Cleaning Up Obsolete Elements During Policy Manager Upgrades

Use the next procedure to remove obsolete elements during Policy Manager upgrades from release 6.5 to 7.0.

To remove obsolete elements during Policy Manager upgrades

1. Navigate to the upgrade directory:

   ```
   PolicyManager_install_dir/identity/oblix/tools/migration_tools
   ```

2. Locate the appropriate schema file for the directory server as specified in Table C–3.

3. Take the appropriate action for your environment:

   - **Same Directory Server**—If configuration data and policy data are in the same directory server, invoke the schema upgrade tool `ds_conf_update` using the command:

     ```
     PolicyManager_install_dir/identity/oblix/tools/ldap_tools
     /ds_conf_update -f schema_file
     ```

     where *schema_file* is the name of the policy schema cleanup file from Table C–3.

   - **Different Directory Server**—If configuration and policy data are in different directory servers, invoke the schema upgrade tool `ds_conf_update` twice: once for the configuration schema and a second time for the policy schema as follows.

     ```
     PolicyManager_install_dir/identity/oblix/tools/ldap_tools
     /ds_conf_update -f schema_file
     ```

     where *schema_file* is the name of the configuration/policy schema cleanup file from Table C–3 or Table C–2.

4. Respond to prompts for various options such as host, port, userid, and password.

Note: The tool may generate errors of type Attribute/Object does not exist. These may occur when LDIF files contain a list of all obsolete schemas from Oracle Access Manager release 3.6 which might not be present in your directory server.

Uploading the Generated LDIF

After upgrading the configuration tree and optionally removing obsolete schema elements, you are ready to upload the generated LDIF.

To upload the generated LDIF

1. Run ldapmodify to upload the generated LDIF. For example:

   ```
   \IdentityServer_install_dir\identity\oblix\tools\ldap_tools\ldapmodify
   run ldapmodify.exe -f generated_ldif
   ```

   This program prompts for various options, such as host, port, userid, and password.

2. Upgrade user data, as described in "Upgrading User Data Manually" on page 10.

3. Repeat earlier steps using the next highest data_*fromrelease*_to_*release*_osd.lst file, until you have upgraded all data to release 10*g* (10.1.4.0.1).

Upgrading User Data Manually

Release numbers used here are simply for illustration. If you have an earlier release than 6.1.1, be sure to contact Oracle Support before upgrading:
`http://www.oracle.com/support/contact.html`

User data upgrades are required *only* while making a move, either a direct or an intermediate move, from Oracle Access Manager 5.2.x to Oracle Access Manager 6.x.

To upgrade the user data manually

1. Copy the data_*fromrelease*_to_*release*_user.lst file and rename it to retain the original.

   For example:

   From

   ```
   \IdentityServer_install_dir\identity\oblix\tools
   \migration_tools\obmigratedata\data_520_to_600_user.lst
   ```

   To

   ```
   \install_dir\identity\oblix\tools\migration_tools\obmigratedata
   \config_data_520to600_user.lst
   ```

2. See Table C–4 as you edit the keys in the file to provide the information for your environment based on the annotated samples here.

   A complete, ***annotated*** version of the data_520_to_600_xxx.lst file is shown in "Sample data_520_to_600_xxx.lst" on page 16. Both the osd.lst and user.lst files contain similar information.

3. Confirm that the Oracle Access Manager configuration tree and data parameter values are false.

   ```
   osd_migration:false
   policy_migration:false
   wf_migration:false
   ```

4. Confirm that the user data upgrade parameter value is true.

   ```
   user_migration:true
   ```

   Note: Although a user-data upgrade does not do anything with the configuration data tree, it is a good idea to complete step 5.

5. Back up the configuration tree.

6. Upgrade user data by running obmigratedata.exe and specifying the name of your updated file. For example:

   ```
   run obmigratedata.exe -f config_data_520to600_user.lst
   ```

Table C–4 Keys to Add or Edit

Key	Description
hostname: *host name*	Directory server host
portNo: *port number*	Directory server port
bindDN: *DS credentials*	The DN of the directory server administrator account. This can be found in *installdir*/oblix/config/ldap/AppDB.xml.
password: *encrypted password*	Password of the directory server administrator account. This can be found in *installdir*/oblix/config/ldap/AppDB.xml.
directoryType: NS | AD | NDS | IBM	The type of directory server you are running
directoryMode: SSL | OPEN	The mode of the directory server you are using. Values can be either SSL or Open.
Oblixnode: ou=Oblix | o=Oblix	RDN of the Oracle Access Manager configuration tree
groupOC: *name of Group object class*	Example: group or groupOfUniqueNames
PersonOC: *name of Person object class*	Example: user or inetOrgPerson
oldVersion: *release number*	Exact release number of the Oracle Access Manager 5.2 system. This can be found in ./oblix/config/np52_is.txt. Example: 5.2, 5.2.1.12
oldVersionSearchBase: *searchbase*	The searchbase to use. Typically, this is the global searchbase.
binAttrFileName:at_520_to_600_binary.lst	Accept this file as shown. It contains important information for the upgrade program.
objclassMapFileName:oc_520_to_600_map.lst	Accept this file as shown. It contains important information for the upgrade program.
logFileName: *filename*	Name of the file to receive logging information during the conversion process.
outputFileName: *filename*	Name of the LDIF file to receive the converted data.
missedSuppliedAttrsDetailsFileName: *filename*	Name of the file to receive workflows containing Provisioned attributes in Oracle Access Manager 5.2. These workflows must be modified manually in the applet to associate the appropriate subflow with them.
WFDefContainer:	DN of the workflow definitions container. Example: obcontainerID=workflowDefinitions,OU= oblix,DC=company,DC=com
WFInstanceContainer	DN of the workflow instance container. Example: obcontainerId=workflowInstances, OU=oblix,DC=company,DC =com
FunctionalityTBMigrated BEGIN:vNameList	
osd_migration: true | false	Perform data upgrades on the configuration tree. **Note:** If you are running data upgrades because of errors in an earlier run: ▪ During the Identity Server upgrade, ensure that values of osd_migration and wf_migration match (true). policy_migration value should be "false". ▪ During Policy Manager upgrades, ensure that values of osd_migration and wf_migration match (false) while policy_migration value is "true".
policy_migration: true | false	Refer to details for osd_migration.

Table C–4 (Cont.) Keys to Add or Edit

Key	Description
wf_migration: true | false	Perform data upgrades on workflow containers specified earlier. **Note:** If you are running data upgrades manually because workflows are on a separate directory server and you want to upgrade them: ▪ During Identity Server, ensure that osd_migration value is "false"; wf_migration value is "true"; policy_migration value is "false". An output LDIF file is generated with workflow definition and instance containers, and migrated definitions and instances. ▪ After Identity Server upgrade, remove the Workflow definition and instance containers from the directory server where they are present and upload the generated output LDIF file there. ▪ During the Policy Manager upgrade, there is nothing to do because the directory server contains only workflows and instances.
user_migration: true | false	Perform data upgrades on User entries (non-Oracle data). Between release 5.2 and 10*g* (10.1.4.0.1), the Challenge Phrase Response encryption format is changed from RC-4 to RC-6. User data upgrade is performed inline by default. The user entries are modified directly without generating an intermediate LDIF.
END:vNameList	
Additional_DS_Info:	If you are upgrading user data and you have multiple user directories, you may specify the additional directory servers in this section.
BEGIN:vCompoundList	
user_migration_ds_1:	For additional user directory servers, the format is user_migration_ds_n Where n is 1, 2, 3, and so on.
BEGIN:vCompoundList	
hostname: *host name*	Directory server host
portNo: *port number*	Directory server port
bindDN: *DS credentials*	The DN of the user data directory server administrator account, which can be found in *installdir*/oblix/config/ldap/AppDB.xml. **Note:** If you have user data stored separately from configuration data, the AppDB.xml file may not contain the appropriate bind DN and encrypted password for the user data directory.
password: encrypted password	The Password of the user data directory server administrator account, which can be found in *installdir*/oblix/config/ldap/AppDB.xml.
directoryType: NS | AD | NDS | IBM	The type of directory server you are running
directoryMode: SSL | OPEN	The mode of the directory server you are using: Values can be either SSL or Open.
oldVersionSearchBase: *searchbase*	The searchbase to use. Typically, this is the global searchbase. **Note:** Every directory server in "Additional DS" in the config.lst file once had the searchbase given with keyword 'oldVersionSearchbase' . However, with 10*g* (10.1.4.0.1), there is no 'oldVersionSearchBase' keyword present in additional directory server sections. Instead, the keyword is 'searchbase'. The value of this keyword need not always be the global searchbase. One additional directory server (for example, user_migration_ds_1) represents one directory-profile from the configuration tree. The 'searchbase' keyword from this section will give you the 'Namespace' covered by this directory-profile.
END:vCompoundList	
END:vCompoundList	

Sample Default obmigratenpparams.lst File

```
BEGIN:vCompoundList
```

```
am: Policy Manager Parameters
BEGIN:vCompoundList
  kMigrateLicense:false
  520_to_600:
  BEGIN:vNameList:
    kMigrateWS:true
    kMigrateData:false
    kMigrateSchema:true
  END:vNameList
  600_to_610:
  BEGIN:vNameList:
    kMigrateWS:false
    kMigrateData:false
    kMigrateSchema:false
  END:vNameList
  610_to_650:
  BEGIN:vNameList:
    kMigrateWS:true
    kMigrateData:true
    kMigrateSchema:true
  END:vNameList
       650_to_700:
       BEGIN:vNameList:
          kMigrateWS:true
          kMigrateData:true
          kMigrateSchema:true
       END:vNameListr
              700_to_1014:
       BEGIN:vNameList:
          kMigrateWS:false
          kMigrateData:true
          kMigrateSchema:true
       END:vNameListr
     END:vCompoundList

     wg: (WebGate)
     BEGIN:vCompoundList
       520_to_600:
       BEGIN:vNameList:
         kMigrateWS:true
       END:vNameList
       600_to_610:
       BEGIN:vNameList:
         kMigrateWS:false
       END:vNameList
       650_to_700:
       BEGIN:vNameList:
         kMigrateWS:true
       END:vNameList
              700_to_1014:
       BEGIN:vNameList:
                kMigrateWS:true
       END:vNameList
     END:vCompoundList

     wp: (webPass)
     BEGIN:vCompoundList
       520_to_600:
       BEGIN:vNameList:
         kMigratePublisher:true
```

```
      END:vNameList
      600_to_610:
      BEGIN:vNameList:
        kMigratePublisher:false
      END:vNameList
      610_to_650:
      BEGIN:vNameList:
        kMigratePublisher:false
      END:vNameList
      650_to_700:
      BEGIN:vNameList:
        kMigrateWS:true
      END:vNameList
            700_to_1014:
      BEGIN:vNameList:
      END:vNameList
    END:vCompoundList

    aaa: (Access Server) Authentication, Authorization, Auditing Parameters
    BEGIN:vCompoundList
      520_to_600:
      BEGIN:vNameList:
        kMigrateData:true
        kMigrateSchema:true
      END:vNameList
      600_to_610:
      BEGIN:vNameList:
        kMigrateData:false
        kMigrateSchema:false
      END:vNameList
      610_to_650:
      BEGIN:vNameList:
        kMigrateSchema:false
                  kMigrateData:true
      END:vNameList
            700_to_1014:
      BEGIN:vNameList:
      END:vNameList
    END:vCompoundList

    bea:
    BEGIN:vCompoundList
      600_to_610:
      BEGIN:vNameList:
        kMigrateASDK:true
      END:vNameList
      650_to_700:
      BEGIN:vNameList:
        kMigrateASDK:true
      END:vNameList
    END:vCompoundList

idlk:
BEGIN:vCompoundList
    kMigrateLicense:false
END:vCompoundList

ois: Identity Server Parameters: True triggers automatic data migration
BEGIN:vCompoundList
    kMigrateLicense:false
```

```
  kMigrateASDK:true
520_to_600:
BEGIN:vNameList:
  kMigrateData:true
  kMigrateSchema:true
  kMigratePublisher:true
END:vNameList

600_to_610:
BEGIN:vNameList:
  kMigrateASDK:true
  kMigrateData:false
  kMigrateSchema:false
  kMigratePublisher:false

  kASDKSubDir:/AccessServerSDK

END:vNameList

610_to_650:
BEGIN:vNameList:
             kMigrateASDK:true
  kMigrateSchema:true
  kMigrateData:true
  kMigratePublisher:false

  kASDKSubDir:/AccessServerSDK

END:vNameList

650_to_700:
BEGIN:vNameList:
  kMigrateASDK:true
  kMigrateData:true
  kMigrateSchema:true
  kMigratePublisher:false

  kASDKSubDir:/AccessServerSDK

 END:vNameList

       700_to_1014:
   BEGIN:vNameList:
   kMigrateASDK:true
   kMigrateData:true
   kMigrateSchema:true
   kMigratePublisher:false

   kASDKSubDir:/AccessServerSDK
   END:vNameList

  END:vCompoundList

     was:
     BEGIN:vCompoundList
       kMigrateASDK:true
       600_to_610:
       BEGIN:vNameList:
         kMigrateASDK:true
       END:vNameList
```

```
          650_to_700:
          BEGIN:vNameList:
            kMigrateASDK:true
          END:vNameList
        END:vCompoundList

END:vCompoundList
```

Sample data_520_to_600_xxx.lst

Again, release numbers shown here are for illustration only. If your starting release is earlier than 6.1.1, be sure to contact Oracle Support before upgrading: `http://www.oracle.com/support/contact.html`

Both the osd.lst and user.lst files contain similar information. For additional details, see Table B–7. User migration occurs only during the upgrade from Oracle Access Manager 5.2.x to Oracle Access Manager 6.0.0. When you have ADAM as the directory server with Oracle Access Manager 6.5.1 or later, the directory type to be used is ADAM.

```
BEGIN:vCompoundList

  hostName:<hostName>
  portNo:<portNo>
  bindDN:<bindDN>
  password:<password>  Copy encrypted credentials from
\COREid\identity\oblix\config\ldap\AppDB.xml
  directoryMode:<directoryMode> Open or SSL
  directoryType:<directoryType> NS or AD or NDS or IBM

  oldConfigDN:<oldConfigDN> o=company,c=us
  oldVersionSearchbase:<oldVersionSearchbase>Global Searchbase

  binAttrFileName:at_520_to_600_binary.lst
  objclassMapFileName:oc_520_to_600_map.lst

  logFileName:output_520_to_600_osd.log Okay to change
  outputFileName:output_520_to_600_osd.ldif Okay to change
  missedSuppliedAttrsDetailsFileName:output_520_to_600_supplied_osd.txt

  oblixnode:o=oblix For AD use ou=oblix
  groupOC:<groupOC> Your Group
  personOC:<personOC>
  doUTFConversion:<doUTFConversion> True or False

  oldVersion:5.2.1.7 Must be specific
  newVersion:6.0.0 Use proper Release
  encryptionType:Oblix Changes the encryption scheme from RC4 to RC6; use Oblix
unless you have a customized encryption scheme

  #We want to know wf-containers names Workflow definition containers
  WFDefContainer:<wfdefcontainer>
  WFInstanceContainer:<wfdefcontainer>

  FunctionalityTBMigrated: In the following parameters, True migrates
                           automatically; False does not
  BEGIN:vNameList

  osd_migration:true Configuration tree migration (True or False)
  policy_migration:true
```

```
 wf_migration:true

 user_migration:false (True migrates data, False does not)

END:vNameList

oblixdeletedobjects:
BEGIN:vCompoundList
      ad:
      BEGIN:vList
           0ADEL:
           CN=Deleted Objects
      adam:
      BEGIN:vList
           0ADEL:
           CN=Deleted Objects
      END:vList
END:vCompoundList

END:vCompoundList
```

D

Upgrading Sun Web Server Version 4 to Version 6 on Windows 2000

If your earlier Oracle Access Manager environment includes Sun (previously iPlanet) Web server that has been discontinued, you may use the example here to upgrade to a supported Sun Web server release.

- Upgrading Sun Web Server version 4.x to version 6
- Configuring the New Web Server Instance
- Troubleshooting

> **Note:** Specific release numbers are used only to illustrate the sequence of tasks. Refer to your vendor documentation for complete details about administering your directory server release. Specific details of the intermediate upgrade from earlier Oracle Access Manager releases to release 6.1.1 are outside the scope of this manual. Before you start upgrading from an Oracle Access Manager release *earlier* than 6.1.1, contact Oracle Support at `http://www.oracle.com/support/contact.html`.

Upgrading Sun Web Server version 4.x to version 6

The new Web server release should be in a separate area of the Windows 2000 file system. Be sure that the Web server 4.1 service is stopped and that the service control window is closed. Otherwise, the 4.1 service will be disabled and marked for deletion, and the new service will not be created. In this state, neither the 4.1 nor the 6.0 Admin Console can operate the server.

> **Note:** This procedure provides details for only Windows 2000 and Sun Web server version 4.1 to 6.0 upgrades. This is an example only. For complete support information, see the Certify tab at `https://metalink.oracle.com`

To upgrade Sun (iPlanet) version 4.x Web Server to Sun version 6

1. Install a Sun Web server release 6 on the same Windows 2000 machine that is currently hosting iPlanet 4.1 and the earlier release of Oracle Access Manager.
2. Make a copy of the earlier magnus.conf and obj.conf files for future reference.
3. Stop Sun Web server 4 using either the Services Window or the Sun Web Server Administration Console, then close the service control window.

4. Open the Sun Web server 6 Administration Console, then click the Migrate Server link.
5. In the Sun 6.0 Admin Console, enter the Server Root for the Sun Web server 4 and click the Search button.

   The list of server instances under the root node appears.
6. Select the instance you want to migrate, then click the Migrate button.
7. Select the Document Root option, then click Migrate. For example:

   Use the new server's document root, Migrate

   Note: If you choose the same document root as the old server, migration will create some incorrect entries in your configuration file. For example, if your Web server contains entries for WebGate, the server's document root will be changed to the WebGate directory and accessing the root using `http://server:port/` will show you a directory listing with webgate.dll as the only file. More details about correcting the document root are given later.

   The migration starts and status messages appear. The old configuration is assimilated into a newly created instance for Sun 6.0.
8. Close this browser window and continue with "Configuring the New Web Server Instance".

Configuring the New Web Server Instance

The following things need to be done manually:

- Configuring magnus.conf
- Configuring obj.conf

Configuring magnus.conf

You need to define the logs/access path in magnus.conf for the newly installed Web server instance, as described in the procedure here.

To configure the new Web server instance in magnus.conf

1. In the config directory of the migrated instance (6.0 instance area), locate the magnus.conf file.
2. Search for logs/access in magnus.conf; the path still refers to the old area.
3. In the magnus.conf, update the logs/access path appropriately.

   For example, suppose your 4.1 area is D:\NSWS\Server4, and the 6.0 area is G:\iPlanet6WS, you need to make the following change:

   From

```
Init fn=flex-init access="D:/NSWS/Server4/https-hostname/logs/access"
format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
\"%Req->reqpb.clf-request%\" %Req->srvhdrs.clf-status%
%Req->srvhdrs.content-length%"
```

   To

```
Init fn=flex-init access="G:/iPlanet6WS/https-hostname/logs/access"
format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
\"%Req->reqpb.clf-request%\" %Req->srvhdrs.clf-status%
%Req->srvhdrs.content-length%"
```

Note: In this newly created 6.0 Web server, the access log location is defined as a variable in server.xml. For example: <VARS accesslog="G:/iPlanet6WS/https-hostname/logs/access" />.

If you use this method to add this variable to server.xml, you would have to replace the line in magnus.conf as follows:

From

```
Init fn=flex-init access="D:/NSWS/Server4/https-hostname/logs/access"
format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
\"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

To

```
Init fn=flex-init access="$accesslog"
format.access="%Ses->client.ip% - %Req->vars.auth-user%
[%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

4. Continue with "Configuring obj.conf".

Configuring obj.conf

In the migrated obj.conf file (in the config directory), the document-root directives are all set improperly. For example, suppose the following section appears in the obj.conf file for your Web server 4.x instance before the migration. Note that the values of the various document-root(s) differ. For example, there is a document root for the Web server (D:/NSWS/Server4/docs) and others for individual objects, shown in bold:

Sample obj.conf 4.x before Migration to 6.0

```
NameTrans fn="NSServletNameTrans" name="servlet"
NameTrans fn="pfx2dir" from="/servlet" dir="D:/NSWS/Server4/docs/servlet"
name="ServletByExt"
NameTrans fn=pfx2dir from=/ns-icons dir="D:/NSWS/Server4/ns-icons"
name="es-internal"
NameTrans fn=pfx2dir from=/mc-icons dir="D:/NSWS/Server4/ns-icons"
name="es-internal"
NameTrans fn="pfx2dir" from="/help" dir="D:/NSWS/Server4/manual/https/ug"
name="es-internal"
NameTrans fn="pfx2dir" from="/manual" dir="D:/NSWS/Server4/manual/https"
name="es-internal"
NameTrans fn=document-root root="D:/NSWS/Server4/docs"
PathCheck fn=nt-uri-clean
PathCheck fn="check-acl" acl="default"
...
...
<Object name="access_lost_pwd_mgmt">
NameTrans fn="document-root" root="G:/52/webpass/access/oblix/apps/lost_pwd_
mgmt/bin"ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/html"
Service fn="OBlost_pwd_mgmt_Service" method="(POST|GET)"
```

```
</Object>
# Oblix Access Manager Objects #AMOBJECTS
# Oblix WebGate Objects start #WGOBJECTS
<Object name="access_web_gate">
NameTrans fn="document-root"
root="G:/52/webpass/access/oblix/apps/webgate/bin"ObjectType
fn="type-by-extension"
ObjectType fn="force-type" type="text/html"
Service fn="OBWebGate_Control" method="(POST|GET)"
</Object>
```

During migration you were asked to choose to continue with the old document root or a new Web server document root. When you chose a new document root, a new location (for example, G:/iPlanet6WS/docs) is assigned to each document-root in the obj.conf file. Thus the section of the obj.conf file shown earlier would look something like this in the 6.0 release after migration.

Sample obj.conf after Migration to 6.0

```
NameTrans fn="NSServletNameTrans" name="servlet"
NameTrans fn="pfx2dir" from="/servlet" dir="D:/NSWS/Server4/docs/servlet"
name="ServletByExt"
NameTrans fn=pfx2dir from=/ns-icons dir="D:/NSWS/Server4/ns-icons"
name="es-internal"
NameTrans fn=pfx2dir from=/mc-icons dir="D:/NSWS/Server4/ns-icons"
name="es-internal"
NameTrans fn="pfx2dir" from="/help" dir="D:/NSWS/Server4/manual/https/ug"
name="es-internal"
NameTrans fn="pfx2dir" from="/manual" dir="D:/NSWS/Server4/manual/https"
name="es-internal"
NameTrans fn=document-root root="G:/iPlanet6WS/docs"PathCheck fn=nt-uri-clean
PathCheck fn="check-acl" acl="default"
…
…
<Object name="access_lost_pwd_mgmt">
NameTrans fn="document-root" root="G:/iPlanet6WS/docs"ObjectType
fn="type-by-extension"
ObjectType fn="force-type" type="text/html"
Service fn="OBlost_pwd_mgmt_Service" method="(POST|GET)"
</Object>
# Oblix Access Manager Objects #AMOBJECTS
# Oblix WebGate Objects start #WGOBJECTS
<Object name="access_web_gate">
NameTrans fn="document-root" root="G:/iPlanet6WS/docs"
ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/html"
Service fn="OBWebGate_Control" method="(POST|GET)"
</Object>
```

You use the next procedure to validate or correct these.

To configure the new Web server instance in obj.conf

1. In the config directory of the migrated instance (6.0 instance area), locate the obj.conf file.
2. Locate and verify the document roots point to the release 6 Web server instance.

Note: The following line in obj.conf may still refer to the old location:

```
NameTrans fn="pfx2dir" from="/servlet"
dir="D:/NSWS/Server4/docs/servlet" name="ServletByExt"
```

In the newly created release 6.0 Web server, this entry is as follows:

```
NameTrans fn="pfx2dir" from="/servlet" dir="$docroot/servlet"
name="ServletByExt"
```

when the variable document root is defined in server.xml, as shown in step 3.

3. Locate and verify that the variable document root is defined in server.xml as follows.

   For example

```
-<VSCLASS id="defaultclass" objectfile="obj.conf" rootobject="default"
acceptlanguage="off">
  <VARS docroot="g:/iPlanet6WS/docs" />
-<VS id="https-hostname" connections="group1" mime="mime1"
urlhosts="lucerne.persistent.co.in" aclids="acl1">
  <VARS webapps_file="web-apps.xml" webapps_enable="on" />
  <USERDB id="default" database="default" />
  </VS>
  </VSCLASS>
```

   Alternatively, you can change obj.conf without updating server.xml by replacing the line from obj.conf by the following one -

```
NameTrans fn="pfx2dir" from="/servlet" dir="G:/iPlanet6WS/docs/servlet"
name="ServletByExt"
```

4. Search the migrated obj.conf for any mention of the old install area (in this example, D:/NSWS/Server4) to ensure that the v 6.0 instance of the Web server does not refer to the old install area in any way.

Troubleshooting

For information about troubleshooting this process, see "Troubleshooting Sun Web Server Upgrades" on page F-9.

E

Planning Worksheets and Tracking Checklists

As discussed in Chapter 1, "Upgrade Overview and Planning", planning deliverables include a document where you define and record a detailed plan that identifies how the upgrade process is to be performed within each of your installed environments. The details that you need to include for each component and the environment are described in this chapter, as follows:

- About Completing Planning Worksheets and Checklists
- Worksheet for Your Overall Deployment
- Worksheet for Directory Instances
- Worksheet for DIT and Object Definition Details
- Worksheet for Directory Server/RDBMS Profiles
- Worksheet for Database Instance Profiles
- Worksheet for Earlier Identity Servers
- Worksheet for Earlier WebPass Instances
- Worksheet for Earlier Policy Manager Instances
- Worksheet for Earlier Access Servers
- Worksheet for Earlier WebGates/AccessGates
- Worksheet for Integration Components and Independently Installed SDKs
- Worksheet for Customizations
- Checklist for Schema and Data Preparation
- Checklist for the Schema and Data Upgrade
- Checklist for Component Preparation
- Checklist for Component Upgrades
- Checklist for Integration Connector/SDK Upgrades
- Checklist for Customization Upgrades
- Checklist for Validating the Entire Upgrade

About Completing Planning Worksheets and Checklists

As part of your planning deliverables, a filled in worksheet is needed for each installed component. The worksheets in this appendix provide space where you can document details. You may copy worksheets in this appendix and fill them in for each component and customization. You may use earlier installation worksheets as a starting point.

Any details that you can access and print in your earlier installation will save you time and eliminate the possibility of errors. For example, consider printing directory server profiles and DB instance profiles, as well as WebPass, Access Server, and WebGate configuration pages.

Note: Be sure to store worksheets, printed copies, and other recorded details about your installation in a secure location.

For more information, see "Upgrade Planning and Deliverables"page 1-10.

The checklists in this appendix are provided to help you track the progress of tasks that are completed as you and your team perform the preparation and upgrade activities in your enterprise. You will find information about how to perform each task in chapters within this manual. Most items in the checklists are links to more information.

Worksheet for Your Overall Deployment

Table E–1 provides space for you to record general information when planning to upgrade Oracle Access Manager. Other worksheets in this appendix provide space for specific details related to each component.

Table E–1 Details for Your Overall Deployment

Task	Subtask	Overall Deployment Worksheet
0	0.1	**Deployment Name**: _____ **Deployment Type** (circle all that apply): Identity System Only Joint Identity and Access System Intranet Deployment Extranet Deployment Development Test/Demo QA Production Other Master Administrator for this deployment:_____ Deterministic test script developed by:_____ Date of the last validation of system operation:_____
	0.2	**Total number of each component in this environment**: Identity Servers:_____ WebPass Instances:_____ Independently installed SDKs:_____ Identity customizations:_____ **If Joint Identity and Access System, enter, total number of**: Policy Managers (formerly known as Access Manager component):_____ Access Servers:_____ WebGates:_____ Custom AccessGates:_____ Access customizations:_____ Integration connectors:_____ _____
	0.3	**Total number of (and potential downtime windows for)**: Directory Instances for Identity Servers only:_____ Potential downtime windows:_____ **If Joint Identity and Access System**: Directory Instances for Policy Managers only:_____ Potential downtime windows:_____ Directory Instances used by both Identity Servers and Policy Managers:_____ Potential downtime windows:_____
	0.4	**Applications that depend on this deployment, owners, and potential downtime windows**: _____ _____ _____ _____ _____ _____ _____ _____

Table E–1 (Cont.) Details for Your Overall Deployment

Task	Subtask	Overall Deployment Worksheet
	0.5	Change control procedures:_____ _____ Scheduled maintenance windows:_____ _____ Off hours operation windows:_____ _____
	0.6	**Potential Identity System Downtime Estimates**: Preparing for the Identity Schema and Data Upgrade:_____ Directory Server Backups:_____ File System Backups:_____ Schema Upgrade:_____ Data Upgrade:_____ Identity Server Component Upgrades:_____ WebPass Instance Upgrades:_____ Identity System Customization Upgrades:_____ Identity System Customization Redeployment:_____ Identity System Customization After Upgrading:_____ Identity System Upgrade Validation:_____
	0.7	**Potential Access System Downtime Estimates**: Preparing for the Access Schema and Data Upgrade:_____ Directory Server Backups:_____ File System Backups:_____ Schema Upgrade:_____ Data Upgrade:_____ Policy Manager Component Upgrades:_____ Access Server Component Upgrades:_____ WebGate Component Upgrades:_____ Access System Customization Upgrades:_____ Access System Customization Redeployment:_____ Access System Customization After Upgrading:_____ Access System Upgrade Validation:_____

Worksheet for Directory Instances

Table E–2 provides space for the information you need for each directory instance in your existing Oracle Access Manager installation.

Table E–2 Details for Directory Instances

Task	Subtask	Directory Instance Details
1	**1.1**	Directory server type:_____ Directory server version:_____ Directory server patch level:_____
	1.2	Directory Server Details Directory server DNS hostname or IP address:_____ Directory server port #: _____ Root bind DN for Oracle Access Manager:_____ Root password_____ Searchbase_____ Configuration base_____ Directory server security mode Open SSL Disjoint searchbase_____
	1.3	Directory Server/RDBMS Profiles (for more information, see specific worksheets for each) _____ _____ _____ _____ _____ _____
	1.4	Master/replica configuration details: _____ _____ _____ _____
	1.5	Types of data in the directory server (circle all that apply): User Data Configuration Data Policy Data
	1.6	Person Object Class_____ Group Object Class_____ User full name attribute:_____ User login ID attribute:_____ Password attribute:_____
	1.7	User full name attribute:
	1.8	User login ID attribute:
	1.9	Password attribute:

Worksheet for DIT and Object Definition Details

The following worksheet provides space where you can record information about the DIT and object definitions. Table E–3 provides space for the information you need for each directory instance.

Table E–3 DIT and Object Definition Details

Task	Subtask	DIT and Object Definition Details
2	2.1	Directory server DNS hostname or IP address:_____ Directory server port #: _____
	2.2	DIT and schema objects used in Oracle Access Manager_____
	2.3	Object definition details for all objects managed through Oracle Access Manager:

Worksheet for Directory Server/RDBMS Profiles

Table E–4 provides space where you can record information about each directory server/RDBMS profile. Consider printing this information from your existing installation.

Table E–4 Details for Directory Server/RDBMS Profiles for Oracle Access Manager

Task	Subtask	Directory Server/RDBMS Profile Details
3	3.1	Directory server DNS hostname or IP address:_____ Directory server port #: _____
	3.2	Directory Server Profile Profile Name_____:_____ Namespace (searchbase): _____ Directory Type:_____ Dynamic Auxiliary Classes
	3.3	Operations (circle all that apply) Search Operations: Search Entries Authenticate Users Read Operations: Read Entry Write Operations: Create Entry Modify Entry Delete Entry Change Password
	3.4	Used by components (record all that apply) All Identity Servers:_____ _____ _____ _____ Access Servers_____ _____ _____ _____ Policy Managers (formerly Access Managers)_____ _____ _____ _____
	3.5	Write Operations: Create Entry Modify Entry Delete Entry Change Password
	3.6	Database Instances (for more information, see specific worksheets for each) _____ _____ _____ _____ _____ _____ _____ _____
	3.7	Maximum Active Servers:_____ Failover Threshold:_____ Sleep for seconds:_____ Max. Session Time (minutes):_____

Worksheet for Database Instance Profiles

Table E–5 provides space for the information you need for each database instance profile associated with a directory server instance. Consider printing this information from your existing installation.

Table E–5 Details for DB Instance Profiles

Task	Subtask	DB Instance Profile Details
4	**4.1**	Directory Server Instance Name_____ Machine Name hosting the directory instance_____ Port Number: _____ Root DN:_____ Root DN Password:_____ Time Limit:_____ Size Limit:_____ Flags: SSL Referral Fast Bind (AD only) Secure Port Number_____ Initial Connections:_____ Maximum Connections:_____

Worksheet for Earlier Identity Servers

Table E–6 provides space for the information you need for each Identity Server.

Table E–6 Details for Existing Identity Servers

Task	Subtask	Existing Identity Server Details
		Prepare for Identity Server Upgrade in Environment: Total Number of Identity Servers in this environment:
5		**Identity Server Details** Installation directory of this Identity Server _____ Exact Patch Level_____ Operating System and Patch Level_____ Installation directory for the associated WebPass_____
	5.1	Default Locale (Administrator Language) Languages Language Packs
	5.2	Transport security mode between the Identity Server and WebPass: Open Simple Cert
	5.3	Unique Identity Server ID of this instance:_____ Host name of the machine where the Identity Server is installed_____ Port number for Identity Server/WebPass communication_____
	5.4	Is this the master Identity Server? (There can be only one installed to update the schema/data) Directory server type_____ _____ For more information for this Directory Instance, see worksheet____ _____
	5.5	Security mode between directory server and Identity Server: SSL Open
		If SSL, path to the Root CA certificate:
		Simple mode only Global Access Protocol pass phrase
		Cert Mode Only Certificate PEM pass phrase:_____ Path of the certificate request file:_____ Path of the certificate file:_____ Path of the key file:_____ Path of the chain file:_____
	5.6	(**Windows only**) Unique Identity Server service name that will differentiate this instance in the Services window if you install several instances of Identity Server):
	5.7	Auditing configuration _____ _____

Table E–6 (Cont.) Details for Existing Identity Servers

Task	Subtask	Existing Identity Server Details
	5.8	Password policy configuration _____
	5.9	Any customizations (Identity Event plug-ins, styles, Portal Inserts and the like)? See worksheets: _____ _____
	5.10	File-based changes (globalparams.xml, and the like)? _____ _____

Worksheet for Earlier WebPass Instances

Table E–7 provides space for the information you need for each WebPass, some of which may be printed from the Identity System Console.

Table E–7 Details for existing WebPass Instances

Task	Subtask	Existing WebPass Details
6		**Prepare for WebPass Instances Upgrade in Environment:** Total Number of WebPass Instances in this environment:
	6.1	**WebPass Instance Details** Installation directory of this WebPass Instance _____ Exact Patch Level_____ Operating System and Patch Level_____ WebPass hostname:_____ ▪ Installed for Web server instance:_____ ▪ Web Server Type:_____ ▪ Web Server Release:_____ ▪ Exact Web Server Patch Level_____ ▪ Absolute path to the Web server configuration file_____ ▪ User name (Unix only):_____ ▪ Group (Unix only):_____
	6.2	Default Locale (Administrator Language) Languages Language Packs Same Language Packs as the Identity Server
	6.3	Transport security mode between the Identity Server and WebPass: Open Simple Cert
		Simple mode only Global Access Protocol pass phrase
		Cert mode only Certificate PEM phrase:_____ Path of the certificate request file:_____ Path of the certificate file:_____ Path of the key file:_____ Path of the chain file:_____
	6.4	WebPass ID used by Oracle Access Manager to identify the instance:
	6.5	DNS hostname of the Identity Server with which this WebPass communicates:_____ Installation directory for the associated Identity Server_____ Identity Server Port # for communication with WebPass:
	6.6	Any customizations? _____ _____
	6.7	File-based changes? _____

Worksheet for Earlier Policy Manager Instances

Table E–8 provides space for the information you need for each existing Policy Manager (formerly known as the Access Manager component).

Table E–8 Details for Existing Policy Managers

Task	Subtask	Existing Policy Manager Details
7		**Prepare for Policy Manager Upgrade in Environment:** Total Number of Policy Managers in this environment:
	7.1	**Policy Manager Instance Details** Installation directory of this Policy Manager Instance _____ Exact Patch Level_____ Operating System and Patch Level_____ Policy Manager hostname:_____ ▪ Installed for Web server instance:_____ ▪ Web Server Type:_____ ▪ Web Server Release:_____ ▪ Exact Web Server Patch Level_____ ▪ Absolute path to the Web server configuration file_____ ▪ Web server user name (Unix only):_____ ▪ Web server group (Unix only):_____
	7.2	Default Locale (Administrator Language) Languages Language Packs
	7.3	Transport security mode between the Policy Manager and Access Servers: Open Simple Cert
		Simple mode only Global Access Protocol pass phrase:
		Cert mode only Certificate PEM phrase:_____ Path of the certificate request file:_____ Path of the certificate file:_____ Path of the key file:_____ Path of the chain file:_____
	7.4	Is this the master Policy Manager for the schema/data upgrade? Yes No Where is policy data stored? - User data directory server - Configuration data directory server - Separate directory server Directory server type_____ _____ Searchbase where user data is stored:_____ Configuration DN:_____ Policy base:_____ For more information for this Directory Instance, see worksheet____ _____

Table E–8 (Cont.) Details for Existing Policy Managers

Task	Subtask	Existing Policy Manager Details
		If the security mode between the directory server and the Policy Manager is SSL, the path to the SSL certificate is:_____
	7.5	Person object class name:
	7.6	Policy Manager policy domain root:
	7.7	Configured authentication schemes? Yes No If Yes, select authentication scheme or schemes: **Authentication Schemes** - Basic Over LDAP - Client Certificate - Anonymous - Oracle Access and Identity - Oracle Access and Identity for AD Forests - Others _____ _____ _____ _____ _____ _____
	7.8	Configure Oracle Access Manager-related policy domains? Yes No If Yes, select policy domains: **Policy Domains** - Identity Domain (a default) - Access Domain (a default) Others _____ _____ _____ _____ _____ _____ _____
	7.9	Configured policies to protect Oracle Access Manager-related URLs? Yes No Details_____ _____ _____ _____ _____ _____ _____ .
	7.10	Any customizations? _____ _____ _____
	7.11	File-based changes? _____ _____ _____

Worksheet for Earlier Access Servers

Table E–9 provides a space for information you need to record for each earlier Access Server. Consider printing some of this information from the Access System Console.

Table E–9 Details for Existing Access Servers

Task	Subtask	Access Server Details
8		**Access Server Details** Total number of Access Servers
	8.1	**Access Server Instance Details** Installation directory of this Access Server Instance _____ Exact Patch Level_____ Operating System and Patch Level_____
	8.2	**Access Server Details in the System Console** Access Server name_____ Access Server host name_____ Port # the Access Server listens to_____ Transport security between Access Server and associated WebGate: Open Simple Cert Associated WebGate ID_____ Access Management flag On Off
	8.3	Default Locale (Administrator Language) Languages Language Packs
	8.4	Which directory server stores the configuration data? Same as Policy Manager directory server? Yes No Configuration DN_____ If no, see worksheet for directory server instance_____ Host machine_____ Port number_____ Root DN_____ Root DN password_____ Directory type_____ Security mode between the configuration data directory server and the Access Server: Open SSL
	8.5	Which directory server stores the policy data?_____ Policy base_____ For more information about the directory server instance, see the worksheet for_____
	8.6	Save PEM phrase in a password file? (Simple and Cert modes only): Yes No
		Simple mode only Global Access Protocol pass phrase:_____ Password file_____

Table E–9 (Cont.) Details for Existing Access Servers

Task	Subtask	Access Server Details
		Cert mode only Certificate PEM phrase:_____ Password file_____ Path of the certificate request file:_____ Path of the certificate file:_____ Path of the key file:_____ Path of the chain file:_____
	8.7	Auditing configuration _____ _____
	8.8	Any customizations (plug-ins, AccessGates, and the like), see the worksheets: _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____
	8.9	File-based changes? _____ _____

Worksheet for Earlier WebGates/AccessGates

Table E–10 provides space for information you need for each WebGate/AccessGate. Consider printing some of this information from the Access System Console.

Table E–10 WebGate/AccessGate Details

Task	Subtask	WebGate/AccessGate Details
9		**Prepare for WebGate/AccessGate Upgrade in Environment:** Total Number of WebGates in this environment:_____ Total number of custom AccessGates in this environment:_____
	9.1	**WebGate/AccessGate Instance and Web Server Details** Installation directory of this Instance _____ Exact Patch Level_____ Operating System and Patch Level_____ ▪ Installed for Web server instance:_____ ▪ Web Server Type:_____ ▪ Web Server Release:_____ ▪ Exact Web Server Patch Level_____ ▪ Absolute path to the Web server configuration file_____ ▪ Web server user name (Unix only):_____ ▪ Web server group (Unix only):_____
	9.2	**WebGate/AccessGate Details in the Access System Console** WebGate ID_____ WebGate hostname:_____ WebGate port:_____ WebGate password_____ Transport security between the Access Server and WebGate: Open Simple Cert Preferred http host_____ HTTP cookie domain:_____ Cache timeout_____
	9.3	Associated with Access Server ID_____ Access Server DNS hostname_____ Port number on which Access Server listens_____ Priority_____ Number of connections_____
	9.4	Default Locale (Administrator Language) Languages Language Packs
	9.5	Transport security mode between the Access Server and WebGate/AccessGate: Open Simple Cert
		Simple mode only Global Access Protocol pass phrase_____

Table E–10 (Cont.) WebGate/AccessGate Details

Task	Subtask	WebGate/AccessGate Details
		Cert mode only Certificate PEM phrase:_____ Path of the certificate request file:_____ Path of the certificate file:_____ Path of the key file:_____ Path of the chain file:_____
	9.6	Virtual IP and DNS aliases used to reference the WebPass or Web server farm protected with WebGate_____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____
	9.7	Any customizations? _____ _____
	9.9	File-based changes? _____ _____

Worksheet for Integration Components and Independently Installed SDKs

Table E–11 provides the information you need for Oracle Access Manager integration connectors for third-party products as well as independently installed software developer kits (SDKs).

Table E–11 Details for Integration Connectors and Independently Installed SDKs

Task	Subtask	Integration Connectors and Independently Installed SDK Details
10		**Prepare for Upgrade in Environment:** Total Number of Integration Connectors in this environment:_____ Types of Integration Connectors in this environment:_____ _____ _____ Total number of independently installed SDKs in this environment:_____
	10.1	**Integration Connector/SDK Instance and Web/App/Portal Server Details** Installation directory of this Connector/SDK _____ Exact Patch Level_____ Operating System and Patch Level_____ ▪ Installed for Web/App/Portal server instance:_____ ▪ Web/App/Portal server Type:_____ ▪ Web/App/Portal server Release:_____ ▪ Exact Web/App/Portal server Patch Level_____ ▪ Absolute path to the Web/App/Portal server configuration file_____ ▪ User name (Unix only):_____ ▪ Group (Unix only):_____
	10.2	Default Locale (Administrator Language) Languages Language Packs

Worksheet for Customizations

Table E–12 provides the information you need for each customization.

Table E–12 Details for Existing Customizations

Task	Subtask	Details of Existing Customizations
11	**11.1**	Installation directory of the Customization _____ Operating System and Patch Level_____ Other Oracle Access Manager components on this machine? Yes No Identity Server WebPass Policy Manager Access Server WebGate
	11.2	Workflows _____ _____ _____ _____ _____ _____ _____ _____ _____
	11.3	Access Control Lists (ACLs) _____ _____ _____ _____ _____
	11.4	Custom Identity Event plug-ins:_____ _____ _____ _____ _____ _____
	11.5	PresentationXML customizations _____ _____ _____ _____
	11.6	Styles and XSL stylesheet customizations: _____ _____ _____ _____ _____
	11.7	IdentityXML clients and applications: _____ _____ _____ _____ _____

Table E–12 (Cont.) Details for Existing Customizations

Task	Subtask	Details of Existing Customizations
	11.8	Portal Inserts: _____
	11.9	Customized Authentication plug-ins: _____
	11.10	_____ Customized Authorization plug-ins: _____
	11.11	_____ Access Manager API clients: _____

Checklist for Schema and Data Preparation

The checklist in Table E–13 may help you track the progress of preparing for the schema and data upgrade. The checklist includes links to information in this manual. Most tasks are described in Chapter 5, "Preparing for Schema and Data Upgrades". However, general procedures to prepare host machines are described in Chapter 8, "Preparing Components for the Upgrade".

Table E–13 Checklist for Schema and Data Preparation

Done	Checklist for Schema and Data Preparation	Details
	Deployment Name:_____ **Task owner**:_____	
	Developing Strategies for Upgrading in a Replicated Environment	page 5-4
	Configuring the Challenge/Response Phrase at the Object Class Level	page 5-6
	Configuring Unique Namespaces for Directory Connection Information	page 5-7
	Directory instances involved are described on Worksheet _____ _____	
	Preparing Your Directory Instances for the Schema and Data Upgrade	page 5-8
	■ Preparing a Directory Server when Its Release is Deprecated	page 5-9
	■ Changing the Directory Server Search Size Limit Parameter	page 5-9
	■ Directory-specific procedures in Preparing Your Directory Instances for the Schema and Data Upgrade	page 5-8
	Backing Up Existing Oracle Access Manager Data:	page 5-15
	■ Backing up the Earlier Oracle Access Manager Schema	page 5-16
	■ Backing up Oracle Access Manager Configuration and Policy Data	page 5-16
	■ Backing Up User and Group Data	page 5-16
	■ Backing Up Workflow Data	page 5-17
	■ Archiving Processed Workflow Instances	page 5-17
	Backing Up Existing Directory Instances	page 5-18
	Preparing Host Machines for Master Components	page 5-18
	Adding An Earlier Identity System to Use as a Master	page 5-18
	■ Defining Additional Instances in the Existing System Console	page 5-19
	■ Installing the Master COREid Server Instance	page 5-21
	■ Installing the Master WebPass	page 5-22
	■ Setting Up the Master Identity System for the Schema and Data Upgrade	page 5-23
	Joint Identity and Access System Deployments Only After performing all Identity System schema and data preparation tasks described in this table and in Chapter 5, "Preparing for Schema and Data Upgrades", perform remaining tasks in this table.	
	Adding an Earlier Access Manager to Use as a Master	page 5-24
	■ Installing the Master Access Manager for the Schema and Data Upgrade	page 5-25
	■ Setting Up the Master Access Manager	page 5-26

Table E–13 (Cont.) Checklist for Schema and Data Preparation

Done	Checklist for Schema and Data Preparation	Details
	Finishing Preparation includes topics in Chapter 8, "Preparing Components for the Upgrade"	page 5-29
	■ Preparing Release 6.x Environments	page 8-6
	■ Preparing Multi-Language Installations	page 8-6
	■ Backing Up the Existing Installed Directory	page 8-8
	■ Backing Up the Existing Web Server Configuration File	page 8-8
	■ Backing Up Windows Registry Data	page 8-9
	■ Stopping Servers and Services	page 8-9
	■ Logging in with Appropriate Administrative Rights	page 8-10

Checklist for the Schema and Data Upgrade

The checklist in Table E–14 is provided to help you track the progress of upgrading the schema and data. Identity System details are described in Chapter 6, "Upgrading Identity System Schema and Data". If you have a joint Identity and Access System deployment, procedures for the Access System are described in Chapter 7, "Upgrading Access System Schema and Data".

Table E–14 Checklist for Schema and Data Upgrade

Done	Checklist for the Schema and Data Upgrade	Details
	Deployment Name:_____ **Task owner**:_____	
	Prerequisites, all preparation tasks in Checklist for Schema and Data Preparation	page 5-1
	Upgrading Identity System Schema and Data	page 6-1
	■ Upgrading the Schema and Data with the Master Identity Server	page 6-3
	■ Upgrading the Master WebPass	page 6-13
	■ Verifying the Identity System Schema and Data Upgrade	page 6-17
	■ Uploading Directory Server Index Files	page 6-17
	■ Backing Up Upgraded Identity Data	page 6-21
	Joint Identity and Access System Deployments Only After performing all Identity System schema and data upgrade tasks described in this table and in Chapter 6, perform remaining tasks in this table as described in Chapter 7, "Upgrading Access System Schema and Data".	
	Upgrading Access System Schema and Data	page 7-1
	■ Upgrading the Schema and Data with the Master Access Manager Component	page 7-3
	■ Uploading Directory Server Index Files	page 7-9
	■ Verifying the Access Schema and Data Upgrade	page 7-9
	■ Creating a Temporary Directory Profile For Access System Upgrades	page 7-10
	■ Backing Up Upgraded Policy Data	page 7-12

Checklist for Component Preparation

The checklist in Table E–15 may help you track the progress of activities that you and your team perform when preparing for the component upgrade. Procedures are described in Chapter 8, "Preparing Components for the Upgrade". Most procedures apply equally to Identity System-only deployments and to joint Identity and Access System deployments.

Table E–15 Checklist for Component Preparation

Done	Checklist for Component Preparation	Details
	Deployment Name:_____ **Task owner**:_____	
	Checking Compatibility with Previous Releases	page 8-1
	Copying Custom Identity Event Plug-ins	page 8-2
	Preparing Earlier Customizations	page 8-2
	Preparing the Default Logout in the Policy Manager	page 8-3
	Preparing Host Machines Changing Read Permissions on Password Files	page 8-3 page 8-3
	Preparing Release 6.x Environments	page 8-4
	Preparing Multi-Language Installations	page 8-6
	Backing Up Directories, Web Server Configurations, and Registry Details ▪ Backing Up the Existing Installed Directory ▪ Backing Up the Existing Web Server Configuration File ▪ Backing Up Windows Registry Data	page 8-8 page 8-8 page 8-8 page 8-9
	Stopping Servers and Services	page 8-9
	Logging in with Appropriate Administrative Rights	page 8-10

Checklist for Component Upgrades

The checklist in Table E–16 may help you track the progress of your component upgrades. Identity System procedures are described in Chapter 9, "Upgrading Remaining Identity System Components". Access System procedures are described in Chapter 10, "Upgrading Access System Components".

Table E–16 Checklist for Component Upgrades

Done	Checklist for Component Upgrades	Details
	Deployment Name:_____ **Task owner**:_____	
	Prerequisites, all tasks in Checklist for Component Preparation	
	Upgrading Remaining Identity Servers	page 9-3
	Upgrading Remaining WebPass Instances	page 9-8
	Validating the Identity System Upgrade	page 9-11
	Backing Up Upgraded Identity Component Information	page 9-12
	Joint Identity and Access System Deployments Only Include After performing all Identity System upgrade tasks described in this table and in Chapter 9, perform remaining tasks in this table as described in Chapter 10, "Upgrading Access System Components".	
	Upgrading Remaining Policy Managers	page 10-2
	Upgrading Access Servers	page 10-6
	Upgrading WebGates	page 10-9
	Backing Up Upgraded Access System Component Directories	page 10-13

Checklist for Integration Connector/SDK Upgrades

The checklist in Table E–17 may help you track the progress your integration connector or independently installed SDK upgrades (or both). The procedures are described in Chapter 11, "Upgrading Integration Components and an Independently Installed SDK".

Note: In an Identity System-only deployment, there will be no integration connectors to upgrade. When you have a joint Identity and Access System deployment, you must upgrade integration connectors before independently installed SDKs for the Access System.

Table E–17 Checklist for Integration Connector/Independently Installed SDK Upgrades

Done	Checklist for Integration Connector/Independently Installed SDK Upgrades	Details
	Deployment Name:_____ **Task owner**:_____	
	Prerequisites, all tasks in Checklist for Component Upgrades	
	Identity System-Only Deployments	
	Upgrading Independently Installed Software Developer Kits	page 11-4
	Backing Up Upgraded Integration Connector or SDK Data	page 11-6
	Joint Identity and Access System Deployments Only	
	Upgrading Third-Party Integration Connectors	page 11-4
	Upgrading Independently Installed Software Developer Kits	page 11-4
	Backing Up Upgraded Integration Connector or SDK Data	page 11-6

Checklist for Customization Upgrades

The checklist in Table E–18 may help you track the progress of customization upgrades in your environment. Specific Identity System procedures are described in Chapter 12, "Upgrading Your Identity System Customizations". Access System procedures are described in Chapter 13, "Upgrading Your Access System Customizations".

Table E–18 Checklist for Customization Upgrades

Done	Checklist for Customization Upgrades	Details
	Deployment Name:_____ **Task owner**:_____	
	Prerequisites, all tasks in: ■ Checklist for Component Upgrades ■ Checklist for Integration Connector/SDK Upgrades	
	Identity System-Only Deployments	
	Upgrading Auditing and Access Reporting for the Identity System	page 12-2
	Combining Challenge and Response Attributes on a Panel	page 12-8
	Confirming Identity System Failover and Load Balancing	page 12-9
	Migrating Custom Identity Event Plug-Ins	page 12-10
	Ensuring Compatibility with Earlier Portal Inserts	page 12-11
	Incorporating Customizations from Release 6.5 and 7.x	page 12-12
	Incorporating Customizations from Releases Earlier than 6.5	page 12-13
	Validating Identity System Customization Upgrades	page 12-23
	Other Customizations (see worksheet)_____ _____ _____ _____	
	Backing Up Upgraded Identity System Customizations	page 12-24
	Access System Customizations Only	
	Upgrading Auditing and Reporting for the Access Server	page 13-2
	Confirming Access System Failover and Load Balancing	page 13-3
	Upgrading Forms-based Authentication	page 13-4
	Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins	page 13-5
	Associating Release 6.1.1 Authorization Rules with Access Policies	page 13-5
	Assuring Proper Authorization Failure Re-directs After Upgrading from 6.1.1	page 13-6
	Updating the ObAMMasterAuditRule_getEscapeCharacter in Custom C Code	page 13-7
	Validating Access System Customization Upgrades	page 13-7
	Other Customizations (see worksheet)_____ _____ _____ _____	
	Backing Up Upgraded Access System Customizations	page 13-7

Checklist for Validating the Entire Upgrade

The checklist in Table E–19 may help you track the progress your customization upgrades. Specific procedures are described in Chapter 14, "Validating the Entire System Upgrade".

Table E–19 Checklist for Validating All Upgrades

Done	Checklist for Validating All Upgrades	Details
	Deployment Name:_____ **Task owner**:_____	
	Prerequisites, all tasks in: ▪ Checklist for Component Upgrades ▪ Checklist for Integration Connector/SDK Upgrades ▪ Checklist for Customization Upgrades	
	Identity System-Only Deployments	
	Validating the Identity System Upgrade	page 14-1
	Reverting Identity Server Backward Compatibility	page 14-3
	Joint Identity and Access System Deployments After performing all Identity System upgrade tasks described in this table, perform remaining tasks in this table to validate the upgraded Access System upgrade.	
	Validating Access System Upgrades	page 14-2
	Deleting the Temporary Directory Server Profile	page 14-2
	Reverting Access Server Backward Compatibility	page 14-4

F

Troubleshooting the Upgrade Process

In addition to the guidelines and techniques presented throughout this guide, this chapter provides troubleshooting details that you can employ during or after the upgrade process. Topics include:

- Accessing Log Files
- Access Server Not Processing Earlier WebGate Data Properly
- Auditing and Access Reporting Issues
- Authentication Failures
- Authorization Failure Re-direct Problems After Upgrading from 6.1.1
- Challenge and Response Phrase Issues
- Challenge Response May Not Convert Properly
- Compatibility of Earlier Plug-ins in the Upgraded Environment
- Customized Styles, Images, and JavaScript
- Deleting the vpd.properties File
- Ensuring Compatibility with Earlier Portal Inserts
- Failover and Load Balancing Issues in Upgraded Environments
- Identity Server Not Processing Data from Earlier Plug-ins
- IdentityXML Calls Fail After WebGate Install
- LDAP Add Errors in a Replicated Environment
- Manual Schema Upload Fails
- Mime_types -related Customizations Not Retained
- Searches Are Slow
- Troubleshooting Sun Web Server Upgrades
- Users Cannot Log In
- WebSphere Application Server and Portal Server Upgrades

Accessing Log Files

During each component upgrade, one or more log files may be produced to inform you if any problem should arise. If a log file is created, a message during the upgrade process indicates the name and location of each log file created. In general, you can find upgrade log files in:

Log File Path:

Component_install_dir\identity|access\oblix\tools\migration_tools*toolname*.log

where *Component_install_dir* is the directory where the specific component is installed; identity|access represents the system to which the component belongs (Identity System or Access System, respectively); and *toolname* represents the name of the utility that produced the log. For example, the following log files may be generated:

General Log Files

obmigratenp.log (generated by the main tool obmigratep which calls the other tools)

obmigratefiles.log (generated by the obmigratefiles tool that reads a given map-file and copies files from source directory to target directory based upon the mapping list)

obmigrateparamsg.log (generated by the obmigrateparamsg tool which performs the parameter and message catalog upgrade)

obmigrateds.log (generated by the obmigrateds tool which performs the schema and data upgrade)

Component-specific Log Files

obmigrateNetPointOIS.log (Identity Server)
obmigrateNetPointWP.log (Webpass)
obmigrateNetPointAM.log (Policy Manager)
obmigrateNetPointAAA.log (Access Server)
obmigrateNetPointWG.log (WebGate)

Each log file contains information about a particular activity that occurs during the component upgrade. For example, a separate log file may be generated for file upgrades, or message and parameter upgrades, or the Oracle Access Manager schema upgrade to name a few.

In general, log files include the following information that you can use to troubleshoot specific problems:

- A snapshot of the steps being executed by the respective tool is recorded to help you identify any failure points.
- Any argument details passed while the tool is executing the tool is logged to help you detect any incorrect values that were passed. This can also help if you need to execute the tool manually.
- Return code details are logged to help you identify any error being returned. You can communicate the specific error to Oracle Support for analysis.
- During parameter and message catalog upgrades (performed by the obmigrateparamsg tool) a corresponding log file (obmigrateparamsg.log) shows all files that have got upgraded. This helps you identify any missing files to detect any loss of customizations.
- Component-specific log files show the changes that were completed for that specific component. Changes include any component-specific configuration file and registry changes occurring during the upgrade. This helps you identify any upgrade failures for the respective component.

For information about specific log files, their content, and the tools that generate them see Appendix B, "Upgrade Process and Utilities".

Additionally, the following files are created to log any ldap specific errors:

- During Identity Server data migration, error_output_*fromversion*_to_*toversion*_osd.ldif file is created in the *IdentityServer_install_dir*\identity\oblix\tools\migration_tools\obmigratedata directory.
- During Policy Manager data migration, error_output_*fromversion*_to_*toversion*_psc.ldif file is created in the *PolicyManager_install_dir*\access\oblix\tools\migration_tools\obmigratedata directory

Access Server Not Processing Earlier WebGate Data Properly

If you have a newly installed Access Server that does not appear to process information from an earlier WebGate, there may be a backward compatibility problem.

Upgraded Access Servers are automatically enabled for backward compatibility with earlier WebGates. However, if you install a new 10*g* (10.1.4.0.1) Access Server in an environment that includes earlier WebGates, you must manually set the "`IsBackwardCompatible" Value="true`" in the newly installed Access Server globalparams.xml to enable communication with earlier plug-ins and interfaces, as well as earlier WebGates and custom AccessGates. See also "Access Server Backward Compatibility" on page 4-24.

Auditing and Access Reporting Issues

If you had auditing and access reporting configured in your earlier environment, you need to perform specific steps to ensure you can continue using this function. Otherwise, you may notice that some language characters (for example, Chinese or Japanese) in audit records are not being inserted correctly.

The steps you need to take to accommodate globalization changes, even when you have an English only environment, depends on the type of database you are using.

> **Note:** Simply upgrading or altering existing database instances and tables is not supported and could result in permanent truncation and loss of existing data.

Task overview: Correcting auditing and access reporting issues

1. If your environment includes an Oracle database instance for auditing, you can check to ensure that your database character set is AL32UTF8.
2. Review and complete all steps in "Upgrading Auditing and Access Reporting for the Identity System" on page 12-2.
3. Review and complete all steps in "Upgrading Auditing and Reporting for the Access Server" on page 13-2.

Authentication Failures

Users with non Latin-1 login IDs may not be able to log in successfully when using a custom form. This problem can occur when you have internationalized data in your custom login forms, but you have not updated the login HTML encoding to UTF-8.

As discussed in Chapter 4, "System Behavior and Backward Compatibility", in 10*g* (10.1.4.0.1), form-based authentication supports non-ASCII login credentials (username/password). When you use form-based authentication with 10*g* (10.1.4.0.1) WebGates, you must ensure that character set encoding for the login form is set to UTF-8.

Note: Basic Authentication fails with non-ASCII login credentials. Use form-based authentication for non-ASCII login credentials. Use Basic Authentication with ASCII login credentials.

To correct problems with form-based authentication, see "Upgrading Forms-based Authentication" on page 13-4.

Authorization Failure Re-direct Problems After Upgrading from 6.1.1

Problem: Authorization failure redirects may not work as expected after upgrading from release 6.1.1.

Cause: A new authorization inconclusive state was introduced in release 7.x (apart from authorization success and failure states).

Solution: In the Authorization Rule, be sure to specify an explicit Deny rule and change `Allow takes precedence` to `Yes` under the General panel. For more information, see "Assuring Proper Authorization Failure Re-directs After Upgrading from 6.1.1" on page 13-6.

Challenge and Response Phrase Issues

Problem: If your earlier environment has the challenge phrase and response attributes on separate panels, then the response attribute will not be displayed in the Profile page.

Cause: In earlier releases, the challenge phrase and response attributes were allowed on different panels of the Profile page of the User Manager, Group Manager, and Organization Manager. In 10*g* (10.1.4.0.1), however, both the challenge phrase and response attributes must be on the *same* panel.

Solution: You need to update your panel definitions to include the response attribute on the same panel as the challenge attribute. For more information, see "Combining Challenge and Response Attributes on a Panel" on page 12-8.

Challenge Response May Not Convert Properly

If you choose to manually migrate data during an upgrade (exporting user data from old instance into new instance which is not recommended), the Challenge Response for lost password management may not convert properly. As a result, some users will not be able to use the lost password feature. For example, when providing a correct response on the Lost Password Management page the user cannot reset the password. Also some users might not be able to set new responses, basically it will complain that the old response is not correct.

The Challenge Response value is encrypted with the shared secret in the CPResponseEncryptionKey node, using the RC6 encryption algorithm. The Challenge Response encryption key contains the attributes:

DN:

```
cn=CPResponseEncryptionKey,obcontainerId=encryptionKey,o=Oblix,<container>

Attribute: obSecretSize
Attribute: obSharedSecret
```

where Attribute: obSharedSecret is a binary attribute.

Note: If the configuration tree moved or a different directory server is used in the upgraded environment, the shared secrets may not match.

To ensure the Challenge Phrase Response is properly converted

1. Use caution with the shared secret, which cannot be copied and pasted.
2. Manually document the shared secret in your original configuration tree and add it to the 10*g* (10.1.4.0.1) configuration tree.

For more information, see "Encryption Schemes and the Shared Secret" on page 3-8 and "Checking Compatibility with Previous Releases" on page 8-1.

Compatibility of Earlier Plug-ins in the Upgraded Environment

If your earlier customized plug-ins are not operating as expected after the upgrade when working with internationalized data (that is, non latin-1 data), you need to redesign these to ensure they can process UTF-8 encoded data. To send or receive internationalized data, earlier custom plug-ins must be redesigned to use UTF-8 encoding.

Also, on Solaris and Linux, plug-ins earlier than release 7.x must be re-compiled using the GCC v3.3.2 C++ compiler, regardless of the compiler that may be provided with the Operating System.

Note: Release 7.0 plug-ins as well as earlier plug-ins implemented as executables or those using a scripting language (such as perl) do not require recompiling after the upgrade. However, to send and receive internationalized data, earlier plug-ins should be redesigned to communicate using UTF-8 encoding.

To ensure compatibility of your earlier plug-ins in the upgraded environment, see:

- Migrating Custom Identity Event Plug-Ins
- Recompiling and Redesigning Custom Authentication and Authorization Plug-Ins

Customized Styles, Images, and JavaScript

Broken images or the incorrect appearance of a customized Graphical User Interface (GUI) or JavaScript errors is an indication that earlier customizations have not been manually incorporated into the upgraded environment.

As discussed earlier, customized .XSL style files, images, and JavaScript files are not migrated during the upgrade. If your previous installation includes significant changes to earlier XSL stylesheets, or if you use a style other than the Oracle Access Manager default Classic Style, you need to manually include those changes in 10*g* (10.1.4.0.1) stylesheets, images, and JavaScript files.

WARNING: If you simply copy earlier stylesheets, you may receive stylesheet bug reports or experience unpredictable behavior when using new features designed to work with new stylesheets.

For details about migrating customized styles, images, and JavaScript (including message handling), see details in Chapter 12, "Upgrading Your Identity System Customizations".

Deleting the vpd.properties File

If previous installations and upgrades have left behind a vpd.properties file, you may have trouble when you specify the installation directory. This may occur if a component installation terminates (or is terminated by you) after component files were extracted to the designated installation directory and you simply remove the installation directory without running the Uninstaller. In this case, you are left with an inconsistent vpd.properties file.

Before starting an upgrade you need to remove this file.

To remove the vpd.properties file

1. Locate the vpd.properties file. For example:
   - **On Windows NT**: vpd.properties file is located in c:\WINNT.
   - **On Unix**: The vpd.properties file is located in the home directory of the user running the installer
2. Delete it.

Ensuring Compatibility with Earlier Portal Inserts

After the upgrade if you notice that your portal inserts are not working as expected, you need to ensure that your portal insert URLs have been manually updated for UTF-8 encoding. Also, to use internationalized data in PresentationXML requests, these requests must indicate UTF-8 encoding.

Oracle Access Manager 10*g* (10.1.4.0.1) cannot detect query string character encoding and assumes it to be UTF-8. The 10*g* (10.1.4.0.1) Identity Server cannot process Latin-1 data from earlier Portal Inserts. After upgrading to 10*g* (10.1.4.0.1)), you must change the encoding of the query string in earlier Portal Inserts from Latin-1 to UTF-8.

To ensure compatibility with portal inserts in your environment, see "Ensuring Compatibility with Earlier Portal Inserts" on page 12-11.

Failover and Load Balancing Issues in Upgraded Environments

You should not experience any problems in this area following an upgrade. Refer to following discussions for more information:

- Confirming Identity System Failover and Load Balancing
- Confirming Access System Failover and Load Balancing

Identity Server Not Processing Data from Earlier Plug-ins

If you have a newly installed Identity Server that does not appear to process information from an plug-in, there may be a backward compatibility problem.

Upgraded Identity Servers are automatically enabled for backward compatibility with earlier plug-ins. However, if you install a new 10*g* (10.1.4.0.1) Identity Server in an upgraded environment you must manually set the `encoding` flag in the Identity Server oblixpppcatalog.lst to enable communication with earlier plug-ins and

interfaces. See also "Identity Server Backward Compatability" on page 4-18 on page 4-24.

IdentityXML Calls Fail After WebGate Install

IdentityXML calls require authentication credentials. If there is no WebGate protecting WebPass, then the basic credential mechanism is used. This takes the form of username and password embedded in the SOAP request itself. However, when a WebGate is installed later, then the IdentityXML calls must be changed to use a SSO token-based authentication.

The IdentityXML calls need to be changed to first obtain an OBSSOCookie, and then pass that token into all the subsequent calls. An example of how to do this is shown in the *Oracle Access Manager Developer Guide*. Look for details on code examples of deployed IdentityXML functions, and the ObSSOCookie Example.

LDAP Add Errors in a Replicated Environment

If you are upgrading from Oracle Access Manager 5.2.x on Windows 2000 SP3, you may receive LDAP add errors during the upgrade. If you receive these errors, you may need to set the replication agreements for the machines.

To you receive LDAP add errors in your Windows environment

1. Install the Support Tools from the Windows 2000 CD.
2. Run the `dcdiag` program, optionally using the `/v` command-line option.
3. Under the Replication test, check for failures.
4. If failures are reported, use the next procedure to troubleshoot LDAP add errors.

To troubleshoot LDAP add errors in a forest

1. Confirm that the clocks are synchronized for the domain controllers.
2. On the command line for all domain controllers in the forest, enter the following:

   ```
   net time /setsntp:machine name
   ```

   Use the same machine name so there is minimal clock skew.
3. Set the group policy for replication:
   a. Open the Users and Computers tool.
   b. Go to Domain Controllers, right click, and select Properties.
   c. Under the Group Policy tab, select Default Domain Controllers Policy, select Computer Configuration, then click Windows Settings.
   d. Select Security Settings, select Local policies, then click User Rights Assignment.
   e. From the right hand side, select Access this computer from the network.
   f. Add ENTERPRISE DOMAIN CONTROLLERS to the access list.
4. Do a replication using the Sites and Services tool:
   1. Go to Sites, select Default-First-site-name, then select Servers.
   2. Select the server name.

3. Select NTDS settings.
4. Right click <automatically generated> on the right hand side, and select replicate now.
5. Enter the `dcdiag` program again on the command line to see if the replication test is now working.

After performing these steps, the schema migration should work properly.

Manual Schema Upload Fails

If you attempt to upload and use the full schema installation files for any directory during the upgrade (instead of using release-specific delta files for an existing schema), the operation will fail. This is because the schema already exists and the files used to upgrade an existing schema provide only the difference between the existing schema and the next release.

For example, suppose you have an installation with ADAM as the directory. If you attempt to upload and install the complete (new installation) schema file (ADAM_oblix_schema_add.ldif andADAM_user_schema_add.ldif) rather than release-specific delta-content files (osd_650_to_700_schema_adam.ldif and policy_650_to_700_schema_adam.ldif) the process will fail. This is because the schema already exists.

Guidelines

- Oracle recommends that you accept automatic schema and data upgrades.
- If you must manually update the schema and data (for ADAM, for example), use only those release-specific delta-content files provided to upgrade your directory schema.

Mime_types -related Customizations Not Retained

You may notice that your mime_types-related customizations are not reflected in the attribute configuration applet in the System Console.

When upgrading, multiple entries with the same ParamName in mime_types (.xml and .lst) files are not retained:

IdentityServer_install_dir/identity/oblix/apps/admin/bin/mime_types.xml
IdentityServer_install_dir/identity/oblix/apps/admin/bin/mime_types.lst

WebPass_install_dir/identity/oblix/apps/admin/bin/mime_types.xml
WebPass_install_dir/identity/oblix/apps/admin/bin/mime_types.lst

The .xml version of the file is used by the Identity Server. The .lst version of the file is used by the WebPass Java applet. Both versions of the file must match. Both versions of the file must reside in the *IdentityServer_install_dir* and in the *WebPass_install_dir*.

For example, if your original mime_types.xml file in *IdentityServer_install_dir*/identity/oblix/apps/admin/bin/mime_types.xml contains the following NameValPair ParamNames:

```
<NameValPair ParamName="application/postscript" Value="ai1"/>
<NameValPair ParamName="application/postscript" Value="eps1"/>
<NameValPair ParamName="application/postscript" Value="ps1"/>
```

the following entries will occur in the newly upgraded file:

```
<NameValPair ParamName="application/postscript" Value="ai1"/>
```

```
(CORRECT)
<NameValPair ParamName="application/postscript" Value="eps"/>
(INCORRECT)
<NameValPair ParamName="application/postscript" Value="ps"/>
(INCORRECT)
```

For existing user entries, the MIME type is stored along with the user entry in the directory. As a result, there is no impact on existing user entries and Oracle Access Manager installations after the upgrade.

> **Note:** Both .lst and .xml versions of the file are needed. You may remove MIME types that are no longer needed or add new MIME types to be associated with the particular attribute for further use. Simply edit the mime_types.lst and .xml files for the Identity Server, then copy these into the *WebPass_install_dir* to replace the earlier version.

To ensure MIME type files are accurate and available in the upgraded environment

1. Edit the Identity Server mime_types.lst file if needed to remove MIME types that are no longer needed or add new MIME types to be associated with the particular attribute for further use.
2. Edit the Identity Server mime_types.xml file to match your edited mime_types.lst file.
3. Copy both Identity Server mime_types files (.lst and .xml) in to the *WebPass_install_dir* to replace the earlier version.

Searches Are Slow

If you do not upload the indexes for iPlanet and NDS directories, the product will work. However, searching will be inefficient and impact performance.

For details, see "Uploading Directory Server Index Files" on page 6-17.

Troubleshooting Sun Web Server Upgrades

The release numbers in this discussion are for illustration only and related to the information in Appendix D, "Upgrading Sun Web Server Version 4 to Version 6 on Windows 2000". Specific details of the intermediate upgrade from earlier Oracle Access Manager releases to release 6.1.1 are outside the scope of this manual. Before you start upgrading from a release *earlier* than Oracle Access Manager 6.1.1, contact Oracle Support at `http://www.oracle.com/support/contact.html`.

There are several potential issues that may occur after upgrading to Sun release 6.0 Web server and upgrading to the Oracle Access Manager 7.0 Identity System.

During the upgrade, the following entries are added to obj.conf:

```
NameTrans fn="pfx2dir" from="/identity/oblix" dir="G:/70/webpass/identity/oblix"
name="idoblix"
…
…
<Object name="idoblix">
PathCheck file=".nsconfig" fn="load-config" descend="1"
</Object>
```

However, in the Oracle Access Manager 5.2 Identity System installed on a version 4.1 Web server, the obj.conf does *not* contain the entries underlined earlier. Even when the version 4 Web server is migrated to release 6 and Oracle Access Manager is upgraded to release 7.0, these entries are not included in obj.conf.

- **If you have enabled cgi for the 4.1 iPlanet instance**, the URL prefix and script directory settings are carried over exactly during migration. If this directory is under the version 4.1 document root, you might want to change this directory by:
  - Either using the release 6.0 Web server Admin Console (Class manager > Programs)
  - Or by hand editing the appropriate line in obj.conf

  For example, an example line in the 6.0 obj.conf is shown next:

  ```
  NameTrans fn="pfx2dir" from="/cgi-bin" dir="D:/NSWS/Server4/docs/cgi-bin"
  name="shellcgi"
  ```

  Note: It is a good idea to search the migrated obj.conf for the old install area (in this case D:/NSWS/Server4) to make sure that the 6.0 instance of the server does not refer to the old install area in any way.

- I**n the file jvm12.conf in the Web server config directory**, the following line can be found after migration. This property contains references to the old (4.1) bits and is not correctly migrated:

  ```
  jvm.classpath=D:/NSWS/Server4/plugins/samples/servlets/beans.10/SDKBeans10.jar;
  D:/NSWS/Server4/plugins/samples/servlets/beans/SDKBeans.jar;D:/NSWS/Server4/bin
  /https/jar/Bugbase.jar;D:/NSWS/Server4/bin/https/jar/Calljsac.jar
  ```

  This line should be replaced by the following:

  ```
  jvm.classpath=G:/iPlanet6WS/plugins/servlets/examples/legacy/beans.10/SDKBeans
  10.jar
  ```

  Note: The other jars are not to be included in the release 6.0 Web server configuration and have been intentionally left out.

- **Any files or folders that were in your old document-root** need to be copied manually to the same structure in the new document-root. This is important if you want the new Web server instance to behave exactly as the old one.
- **As noted earlier, both Admin Consoles (version 4.1 and version 6.0) can operate the server**, which work using the 6.0 binaries. The Admin Consoles simply use the Windows NT service to start/stop the server.

  From the 4.1 Console, if you delete the old instance the result is deleting the service and the version 4.1 files. If this happens, even the 6.0 Console cannot operate the instance, because the service has been deleted. Since this is not desirable, the following steps are required:

  a. Stop the server (from either the version 4.1 or the release 6.0 console or using the NT service).

  b. If you want to preserve the logs and the like, back up the old logs directory manually.

c. Delete the version 4.1 instance directory manually.

d. Restart the version 4.1 Admin Console.

Note: The upgraded server is no longer available for the version 4.1 Admin Console to manage.

This completes the process.

Users Cannot Log In

If you do not upload appropriate indexes for Oracle Internet Directory after the schema and data upgrade, users will not be able to login.

For details, see "Uploading Directory Server Index Files" on page 6-17.

WebSphere Application Server and Portal Server Upgrades

During installation of the integration connector for WebSphere Application Server and Portal Server, you are asked to provide the WebSphere classes directory path so the following files are added:

jobaccess.jar and NetPointWASRegistry.jar

However, during the upgrade of the integration connector for WebSphere Application Server and Portal Server, you are not asked for this directory. Instead, you need to manually copy the three files (listed in the following example) into the directory following the upgrade, then restart the Websphere Application Server and the Portal Server.

jobaccess.jar
NetPointWASRegistry.jar
NetPointCMR.jar

There is no NetPointCMR.jar in a release 6.5 installation.

For more information, see "Upgrading Third-Party Integration Connectors" on page 11-1.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X