Oracle® Fusion Middleware

Administrator's Guide for Oracle Access Manager

11g Release 1 (11.1.1)

E15478-02

August 2010

Oracle Fusion Middleware Administrator's Guide for Oracle Access Manager 11g Release 1 (11.1.1)

E15478-02

Primary Author: Gail Flanegin

Contributing Author:

Contributor: Carlos Subi, Harshal X Shaw, Jeremy Banford, Rey Ong, Ramana Turlapati, Deepak Ramakrishnan, Vadim Lander, Vamsi Motokuru, David Goldsmith, Vishal Parashar

Contents

2 Getting Started with OAM Administration and Navigation

Part II OAM 11g System Management

3 Managing Data Sources

4 Managing OAM Server Registration

5 Registering Partners (Agents and Applications) by Using the Console

6 Registering Partners (Agents and Applications) Remotely

Part III Single Sign-on, Policies, and Testing

7 Introduction to the OAM Policy Model, Single Sign-On

8 Managing Policy Components

10 Validating Connectivity and Policies Using the Access Tester

11 Configuring Centralized Logout for OAM 11g

Part IV Session Management and Life Cycle Management

12 Managing Sessions

Part V Logging and Auditing

13 Logging Component Event Messages

14 Auditing OAM Administrative and Run-time Events

Part VI Monitoring OAM Performance

15 Monitoring OAM Metrics by Using Oracle Access Manager

16 Monitoring OAM Performance by Using Fusion Middleware Control

Part VII Using OAM 10g WebGates with OAM 11g

17 Managing OAM 10g WebGates with OAM 11g

B Co-existence Overview: OAM 11g and OSSO 10g

C Integrating Oracle ADF Applications with Oracle Access Manager 11g SSO

D Internationalization and Multibyte Data Support for OAM 10g WebGates

E Securing Communication with OAM 11g

F Introduction to Custom WLST Commands for OAM Administrators

G Configuring OAM 11g for IPv6 Clients

H Troubleshooting

List of Figures

List of Tables

List of Examples

What's New

This section describes new features of the Oracle Access Manager 11g Release 1 (11.1.1).

Product and Component Name Changes

The original product name, Oblix NetPoint, was changed to Oracle Access Manager and OAM 7.x releases were available from Oracle as part of Oracle Application Server 10g Release 2 (10.1.2).

Many component names remain the same. However, there are several important changes that you should know about, as shown in the following table:

	OAM 10g	**OAM 11g**
Deployment	Stand alone server	Deployed in a container
Component Names	Access Server Policy Manager WebGate AccessGate Identity Server WebPass	OAM Server OAM Administration Console OAM Agent OAM Agent N/A N/A
Agents	WebGate AccessGate	OAM Agent OAM Agent
Console Names	Policy Manager Identity System Console Access System Console	OAM Administration Console N/A N/A
Directory Profiles	Directory Profiles	User-Identity Stores
Identity Administration	Identity Server	Identity agnostic (Oracle Identity Manager 11g is used by default)
Administrators	Master Administrator Master Identity Administrator Master Access Administrator Delegated Administrators	OAM Administrator N/A N/A N/A
Agent and partner application registration	N/A	OAM Administration Console Remote registration tool provides automated Agent registration and application domain creation with default security policies

	OAM 10g	OAM 11g
Automated creation of OAM 10g form-based authentication scheme, policy domain, access policies, and WebGate profile for the Identity Asserter for single sign-on	OAMCfgTool Platform-agnostic tool and scripts	N/A
Configuration Store	LDAP	XML file
Policy Store	LDAP	XML file or RDBMS
Policy Model	Open (default allow)	Closed (default deny)
Policy Domain	Policy Domain	Application Domain
Session management	Stateless, stored in a cookie	Stateful, stored on the server
Authentication to LDAP	LDAP defined system wide	LDAP defined in an authentication scheme
Resource Types	Resource Type	Resource Type
Resources	Resource	Resource
Host Identifiers	Host Identifiers	Host Identifiers
Authentication	Authentication Authentication Scheme Authentication Plug-ins Authentication Rule	Authentication Authentication Scheme Authentication Modules Authentication Policy
Authorization	Authorization Authorization Rule Authorization Expression	Authorization Constraint Authorization Policy
Actions	Actions	Responses
Software Developer Kit	Access Manager SDK	Access Manager SDK
Access Protocol	NetPoint Access Protocol (NAP)	Oracle Access Protocol (OAP)
Access Protocol port number	6021	5575 (assigned by the Internet Assigned Numbers Authority (IANA))

New Features for Release 11g Release 1 (11.1.1)

See Chapter 1, "Introduction to Oracle Access Manager 11g and Administration"

Preface

This guide provides information on daily administration and policy configuration tasks using Oracle Access Manager.

Audience

This document is intended for administrators who are familiar with the following concepts:

- Oracle WebLogic Server concepts and administration
- LDAP server concepts and administration
- Database concepts and administration (for policy and session management data)
- Web server concepts and administration
- WebGate and mod_osso agents
- Auditing, logging, and monitoring concepts
- Integration of the Policy store, Identity store, and familiarity with Oracle Identity Management and OIS might be required

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Deaf/Hard of Hearing Access to Oracle Support Services

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

Related Documents

For more information, see the following documents in the Oracle Fusion Middleware 11g Release 1 (11.1.1) documentation set:

- *Oracle Access Manager 11g Release 1 (11.1.1) Release Notes*
- Oracle Fusion Middleware Installation Guide for Oracle Identity Management—Explains how to use the Oracle Universal Installer and the WebLogic Configuration Wizard for initial Oracle Access Manager 11g deployment. Installing Oracle Access Manager 11g WebGates is also covered.
- *Oracle Fusion Middleware Integration Guide for Oracle Access Manager*—Explains how to set up Oracle Access Manager to run with other Oracle and third-party products
- *Oracle Fusion Middleware Upgrade Planning Guide*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management*
- *Oracle Fusion Middleware Upgrade Guide for Java EE*—For information about the types of Java EE environments available in 10g and instructions for upgrading those environments to Oracle Fusion Middleware 11g.
- *Oracle Fusion Middleware Administrator's Guide*—Describes how to manage Oracle Fusion Middleware, including how to change ports, deploy applications, and how to back up and recover Oracle Fusion Middleware. This guide also explains how to move data from a test to a production environment.
- *Oracle Fusion Middleware Application Security Guide*—Explains deploying Oracle Access Manager 10g SSO solutions, which have been replaced by OAM 11g SSO.
- *Oracle Application Server Single Sign-On Administrator's Guide*—For details about using OracleAS Single Sign-On with mod_osso to protect access to Web applications.
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management*—For a step-by-step guide to deployment.
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*—Provides a section on customized Oracle Access Manager commands in the chapter "Infrastructure Security Custom WLST Commands".

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.

Part I

Introduction and Getting Started

This part of the book provides an introduction to Oracle Access Manager 11g Release 1 (11.1.1).

Part I contains the following chapters

- Chapter 1, "Introduction to Oracle Access Manager 11g and Administration"
- Chapter 2, "Getting Started with OAM Administration and Navigation"

1

Introduction to Oracle Access Manager 11g and Administration

This chapter provides a high-level overview of Oracle Access Manager 11g, administration tasks, and links to chapters in this book where you can find more information. This chapter contains the following sections:

- Introduction to This Book
- Introduction to Getting Started with OAM 11g and Administration
- OAM 11g System Management
- Single Sign-on and Policies
- Session Management
- Logging and Auditing
- Monitoring OAM Performance
- Using OAM 10g WebGates with OAM 11g
- Appendixes

1.1 Introduction to This Book

This book provides information to help administrators manage OAM 11g components and policies within one or more WebLogic administration domains.

Each WebLogic Server domain is a logically related group of Oracle WebLogic Server resources. WebLogic administration domains include a special Oracle WebLogic Server instance called the Administration Server. Usually, the domain includes additional Oracle WebLogic Server instances called Managed Servers, where Web applications and Web Services are deployed.

Information in this book is grouped into the following main parts to help administrators quickly locate information:

- Part I, Introduction and Getting Started
- Part II, OAM 11g System Management
- Part III, Single Sign-on, Policies, and Testing
- Part IV, Session Management and Life Cycle Management
- Part V, Logging and Auditing
- Part VI, Monitoring OAM Performance

- Part VIII, Appendixes

1.2 Introduction to Getting Started with OAM 11g and Administration

This section introduces the information in Part I of this guide and includes the following topics:

- Introduction to Oracle Access Manager and OAM 11g Administration
- Getting Started with OAM 11g Administration and Navigation

1.2.1 Introduction to Oracle Access Manager and OAM 11g Administration

OAM administration tasks can be organized around daily and periodic system administration, policy creation and management, session management, diagnostics, and troubleshooting. Initially, the LDAP group used to define administrators is the same for OAM and WebLogic. Initially, the same credentials are used for log in to both the OAM Administration Console and the WebLogic Server Administration Console. The LDAP group for OAM administrators can be changed.

Oracle Access Manager and Oracle Identity Management are components of Oracle Fusion Middleware 11g. Oracle Fusion Middleware is a collection of standards-based software products that spans a range of tools and services from Java EE and developer tools, to integration services, business intelligence, and collaboration. Oracle Fusion Middleware offers complete support for development, deployment, and management.

For more information about Oracle Access Manager, see the following topics:

- About Oracle Access Manager 11g and Single Sign-On
- Enhancements in Oracle Access Manager 11g
- Oracle Access Manager 10g Functionality Not Available with 11g
- About Installation versus Upgrading

1.2.1.1 About Oracle Access Manager 11g and Single Sign-On

Single sign-on (SSO) enables users, and groups of users, to access multiple applications after authentication. SSO eliminates multiple sign-on requests. Oracle Access Manager 11g is the Oracle Fusion Middleware 11g single sign-on solution. Oracle Access Manager 11g operates independently as described in this book and also operates with the Authentication Provider as described in the *Oracle Fusion Middleware Application Security Guide*

Oracle Access Manager 11g is a Java Platform, Enterprise Edition (Java EE)-based enterprise-level security application that provides restricted access to confidential information and centralized authentication and authorization services. All existing access technologies in the Oracle Identity Management stack converge in Oracle Access Manager 11g.

A Web server, Application Server, or any third-party application must be protected by a WebGate or mod_osso instance that is registered with Oracle Access Manager as an agent. to enforce policies The agent acts as a filter for HTTP requests. Oracle Access Manager enables administrators to define authentication and authorization policies.

> **Note:** WebGates are agents provided for various Web servers by Oracle as part of the product. AccessGates are custom access clients created using the Access Manager SDK for use with non-Web applications. Unless explicitly stated, information in this book applies equally to both.

Oracle Access Manager 11g provides single sign-on (SSO), authentication, authorization, and other services to registered agents (in any combination) protecting resources. Agents include:

- OAM 11g WebGates
- OAM 10g WebGates
- IDM Domain Agent
- OSSO Agents (10g mod_osso)

You can also integrate with OAM 11g, any Web applications currently using Oracle ADF Security and the OPSS SSO Framework, as described in Appendix C.

There are several important differences between Oracle Access Manager 11g and Oracle Access Manager 10g, as described in "Enhancements in Oracle Access Manager 11g".

> **See Also:**
>
> - Product and Component Name Changesin the What's New chapter
> - Introduction to Oracle Access Manager 11g Architecture in Chapter 2

1.2.1.2 Enhancements in Oracle Access Manager 11g

Oracle Access Manager 11g includes several important enhancements that were not available with Oracle Access Manager 10g. These enhancements are listed in Table 1–1.

Table 1–1 Enhancements in* Oracle Access Manager *11g

New Functionality for Oracle Access Manager 11g
■ Platform Support: Oracle WebLogic Server Application Server platform and server portability is available for any platform that runs the supported Oracle WebLogic Server. ■ Installation: Simplified Oracle Access Manager installation using the Oracle Universal Installer and initial deployment using the WebLogic Configuration Wizard is described in the Oracle Fusion Middleware Installation Guide for Oracle Identity Management. ■ Backward Compatibility: Support for mixed-release agents: Register and use Oracle Access Manager 10g agents (WebGates and AccessGates) and OracleAS 10g SSO agents (mod_osso) for SSO s provided. See Chapter 5, Chapter 6, and Part VII. ■ Upgrading and Co-existence: Utilities to upgrade an existing OSSO deployments are provided is described in *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management*. Co-existence after upgrading OSSO is introduced in Appendix B.
Built-in support for OracleAS 10g SSO partner applications, and for single sign-on across OSSO 10g-protected applications and OAM 10g WebGate protected applications. See Part III. Per-agent-based shared secret key increases security and performance by moving cookie encryption and decryption to the agent. See Chapter 5
Embedded LDAP for user and group information is described in Chapter 3. Integration with Oracle Entitlement Server MicroSM to enable database storage of policies. See Chapter 3.

Table 1–1 (Cont.) Enhancements in Oracle Access Manager *11g*

New Functionality for Oracle Access Manager 11g
▪ Usability and lifecycle improvements as described through out this guide ▪ Rich and intuitive graphical user interface is shown throughout this guide
A new OAM 11g Access Tester replaces the OAM 10g Access Tester for on-the-fly evaluation of Oracle Access Manager policies. See Chapter 10
Session Management functions are provided, as described in Chapter 12: ▪ WebGate maximum user session timeout is now supported by WebGate through the host cookie See Table 2–1, " Comparison: OAM 11g versus OAM 10g versus OSSO 10g" ▪ WebGate idle session timeout is now supported using in-memory states through the Oracle Coherence-based Session Management Engine.
Events can be audited using the underlying Oracle Fusion Middleware Common Audit Framework, as described in Chapter 14
Windows Native Authentication is supported with applications protected with either an OSSO Agent or OAM Agent. For more information, see *Oracle Fusion Middleware Integration Guide for Oracle Access Manager*.

See Also: "Oracle Access Manager 10g Functionality Not Available with 11g"

1.2.1.3 Oracle Access Manager 10g Functionality Not Available with 11g

Oracle Access Manager 10g provides several functions that are not included with Oracle Access Manager 11g. Table 1–2 provides an overview.

Table 1–2 Functionality Not Available with Oracle Access Manager 11g

Unavailable or Unsupported Functions
Extensibility framework required for building customizations
Application-domain-level delegated administration
Complex policy constructs (AND, OR semantics for multiple rules)
Impersonation support
LDAP filter-based authorization and response calculations
Authorization for mod_osso-protected resources
Replaced by Oracle Fusion Middleware Identity Manager: Identity Server, WebPass, Identity System Console, User Manager, Group Manager, Organization Manager

1.2.1.4 About Installation versus Upgrading

The *Oracle Fusion Middleware Supported System Configurations* document provides certification information on supported installation types, platforms, operating systems, databases, JDKs, and third-party products related to Oracle Identity Management 11g. You can access the *Oracle Fusion Middleware Supported System Configurations* document by searching the Oracle Technology Network (OTN) Web site:

```
http://www.oracle.com/technology/software/products/ias/files/fusion_
certification.html
```

Following installation, you can configure Oracle Access Manager in a new WebLogic Server domain or in an existing WebLogic Server domain. Using the Oracle Fusion Middleware Configuration Wizard, the following components are deployed for a new domain:

- WebLogic Administration Server
- Oracle Access Manager Console deployed on the WebLogic Administration Server (sometimes referred to as the OAM Administration Server, or simply AdminServer)
- A Managed Server for Oracle Access Manager
- An application deployed on the Managed Server

> **See Also:** Oracle Fusion Middleware Installation Guide for Oracle Identity Management

OracleAS 10g SSO deployments can be upgraded to use Oracle Access Manager 11g SSO. After upgrading and provisioning OSSO Agents with OAM 11g, authentication is based on OAM 11g Authentication Policies. However, only OAM Agents (WebGates/AccessGates) use OAM 11g Authorization Policies. Over time, all mod_osso agents in the upgraded environment should be replaced with WebGates to enable use of OAM 11g Authorization policies.

For details about co-existence after the upgrade, see:

- *Oracle Fusion Middleware Upgrade Planning Guide*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management* (E10129-02)
- Appendix B, "Co-existence Overview: OAM 11g and OSSO 10g"

1.2.2 Getting Started with OAM 11g Administration and Navigation

Administrators use the:

- OAM Administration Console to register and manage OAM system configurations and security elements and policies.

  For a quick tour of OAM 11g Administration Console and the most common functions and tasks, see Chapter 2, "Getting Started with OAM Administration and Navigation".

  > **Note:** Custom Administrative command-line tools (WebLogic Scripting Tool, also known as WLST) provide an alternative to the OAM Administration Console for a specific set of functions, as noted when appropriate in this guide

- WebLogic Server Administration Console to view the Summary of Server Configuration (Cluster, Machine, State, Health, and Listening Port) of deployed OAM Servers within the WebLogic Server domain, and also to Start, Resume, Suspend, Shutdown, or Restart SSL on these servers.

  For details about the WebLogic Server Administration Console, see *Oracle Fusion Middleware Administrator's Guide*.
- Custom OAM WebLogic Scripting Tool for command-line input
- Remote registration tool for registering agents and applicatin domains

1.3 OAM 11g System Management

This section introduces the information in Part II of this guide and includes the following topics:

- Data Sources
- OAM Servers and the Administration Console
- Policy Enforcement Agents

1.3.1 Data Sources

The term "data source" is a Java Database Connectivity (JDBC) term that is used within Oracle Access Manager to refer to a collection of user identity stores or a database for policies.

Oracle Access Manager 11g supports several types of data sources that are typically installed for the enterprise. Each data source is a storage container for various types of information.

> **Note:** Oracle Access Manager configuration data is stored in an XML file: oam_config.xml. Oracle recommends that you use only the OAM Administration Console or WebLogic Scripting Tool (WLST) commands for changes; do not edit this file.

A data source must be registered with Oracle Access Manager 11g to enable authentication when a user attempts to access a protected resource (and during authorization, to ensure that only authorized users can access a resource).

The data source must be installed and registered for OAM 11g during the initial deployment process described in the Oracle Fusion Middleware Installation Guide for Oracle Identity Management.

- User Identity Store: Central LDAP storage in which an aggregation of user-oriented data is kept and maintained in an organized way.

  > **Note:** Oracle Access Manager 11g does not include identity services; there is no native user, group, or role store.

  By default, OAM 11g uses the embedded LDAP in the WebLogic Server domain as the user identity store. However, a number of other external LDAP repositories can also be registered as user identity stores.

- Database: A collection of information that is organized and stored so that its content can be easily accessed, managed, and updated.

  Policy Store: OAM 11g policy data must be stored in a database that is extended with the OAM-specific schema and registered with Oracle Access Manager 11g.

  Session Store: By default, OAM session data is stored within in-memory caches that is migrated to the policy store (database). You can also have an independent database for session data, as described in Chapter 3. For information about sessions, see Chapter 12.

  Audit Store: Audit data can be stored either in a file or in a separate database (not the policy store database). For information on auditing, see Chapter 14.

- A Java keystore is associated with OAM 11g and used to store security keys that are generated to encrypt agent traffic and session tokens. Every OAM and OSSO Agent has a secret key that other agents cannot read. There is also a key to encrypt Oracle Coherence-based session management traffic. However, the keystore is not visible and cannot be managed or modified.

Note: Passwords for keys are stored in a credential store.

Within Oracle Access Manager, User Identity Store details can be managed (registered, viewed, modified, or deleted) from the Oracle Access Manager Administration Console. For more information, see Chapter 3, "Managing Data Sources".

See Also: Appendix F, "Introduction to Custom WLST Commands for OAM Administrators" introduces custom WLST commands to create, edit, or display user identity store configuration.

1.3.2 OAM Servers and the Administration Console

OAM Servers were known as Access Servers in OAM release 10g. OAM Servers provide the Oracle Access Manager 11g runtime instance deployed on Oracle WebLogic Managed Servers. Registered agents communicate with the OAM Server.

Note: Administrators can extend the WebLogic Server domain and add more OAM Servers whenever needed, as described in theOracle Fusion Middleware Installation Guide for Oracle Identity Management.

The OAM Administration Console was known as Policy Manager in OAM release 10g. The OAM 11g Administration Console is a Java EE application that must be installed and run on the same computer as the WebLogic Administration Server. Other key applications that run on the WebLogic Administration Server include the WebLogic Server Administration Console and Enterprise Manager for Fusion Middleware Control.

Note: The OAM Administration Console might also be referred to as the OAM Administration Server. However, it is not a peer of the OAM Server deployed on a WebLogic Managed Server.

Several global settings are shared among all OAM Servers, which can be managed using the OAM Administration Console:

- SSO Engine, as introduced in "Single Sign-On" on page 1-9
- Session Management, as introduced in "Session Management" on page 1-12
- Auditing, as introduced in "Component Event Message Logging" on page 1-13
- Oracle Coherence settings shared by all OAM Servers, as described in
- OAM Proxy details for Simple or Cert mode communication are described in "Managing Common OAM Proxy Simple and Cert Mode Security" on page 4-13

You can use the OAM Administration Console to manage server registrations, as described in Chapter 4, "Managing OAM Server Registration".

Note: You can add a new managed server instance with the OAM Server runtime using either:

- The WebLogic Server Administration Console, which requires that you manually register the OAM Server instance as described in Chapter 4
- The WebLogic Configuration Wizard
- Customized Oracle WebLogic Scripting Tool (WLST) commands for OAM

The last two methods automatically register the OAM Server instance, which appears in the OAM Administration Console; no additional steps are required.

See Also: Appendix F, "Introduction to Custom WLST Commands for OAM Administrators" introduces custom WLST commands to manage server configuration.

Oracle Access Manager 11g Servers are compatible with various policy enforcement Agents. For more information, see "Policy Enforcement Agents".

1.3.3 Policy Enforcement Agents

A policy-enforcement agent is any front-ending entity that acts as an access client to enable single sign-on across enterprise applications.

To secure access to protected resources, a Web server, Application Server, or third-party application must be associated with a registered policy enforcement agent. The agent acts as a filter for HTTP requests, and must be installed on the computer hosting the Web server where the application resides.

Individual agents must be registered with Oracle Access Manager 11g to set up the required trust mechanism between the agent and OAM Server. Registered agents delegate authentication tasks to the OAM Server.

Oracle Access Manager 11g supports the following types of agents in any combination:

- **OAM Agents**: A WebGate is one type of agent. It is a Web server plug-in that acts as an access client. WebGate intercepts HTTP requests for Web resources and forwards them to the OAM Server for authentication and authorization).
  - **WebGate 11g**: Must be installed independently. After registration with OAM 11g, WebGates directly communicates with Oracle Access Manager 11g services. No proxy is used.
  - **WebGate 10g**: Must be installed independently. After registration with Oracle Access Manager 11g, OAM 10g WebGates communicate with OAM 11g services through a Java EE-based OAM proxy that acts as a bridge.

    IDM Domain Agent: This Java agent is installed and registered out of the box to provide SSO protection for resources in the Identity Management domain. The agent's oamsso_logout application is also configured and deployed in the WebLogic (and OAM) AdminServer and all managed servers. The IDMDomainAgent performs as an OAM 10g Agent to enforce OAM 11g policies.

  - **AccessGate 10g**: An AccessGate is a custom access client that was created using the Access Manager software developer kit (SDK). AccessGates can protect Web and non-web resources.

- **OSSO Agent (mod_osso 10g)**: After registration with Oracle Access Manager, OSSO 10g Agents communicate directly with Oracle Access Manager 11g services through an OSSO proxy.

  The OSSO proxy supports existing OSSO agents when upgrading to OAM 11g. The OSSO proxy handles requests from OSSO Agents and translates the OSSO protocol into a protocol for Oracle Access Manager 11g authentication services.

You can use the following methods and tools to register agents with Oracle Access Manager 11g:

- **OAM Administration Console**: Register and manage OAM and OSSO agent registration as described in Chapter 5

- **Remote Registration**: Use the Oracle-provided command-line tool as described in Chapter 6.

From an existing 10g OAM or OSSO deployment you can:

- Provision OAM 10g WebGates with OAM 11g, as described in Chapter 17.

- Upgrade OracleAS 10g SSO (OSSO) as described in the *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management*. Read about co-existence with OAM 11g Servers in Appendix B.

> **See Also:** "Co-existence: OAM 11g SSO versus OAM 10g SSO with OracleAS SSO 10g" on page 1-15

1.4 Single Sign-on and Policies

This section introduces the information in Part III of this guide and includes the following topics:

- Single Sign-On
- OAM Policy Model and Shared Policy Components
- OAM Policy Model, Application Domains, and Policies
- Centralized Logout for OAM 11g
- Connectivity and Policy Testing

1.4.1 Single Sign-On

Single sign-on (SSO) is a process that gives users the ability to access multiple protected resources (Web pages and applications) with a single authentication.

Oracle Access Manager 11g converges SSO architectures such as Identity Federation for Partner Networks, and Service Oriented Architecture (SOA), to name a few. Oracle Access Manager 11g provides single sign-on (SSO) through a common SSO Engine that provides consistent service across multiple protocols.

To delegate authentication tasks to Oracle Access Manager 11g, agents must reside with the relying parties and must be registered with Oracle Access Manager 11g. Registering an agent sets up the required trust mechanism between the agent and Oracle Access Manager 11g SSO.

Note: Single Sign-on for the Oracle Access Manager 11g Administration Console, and other Oracle Identity Management consoles deployed in a WebLogic container, is enabled using the pre-registered IDM Domain Agent and companion application domain. No further configuration is needed for the consoles.

Single sign-on can be implemented in a variety of ways:

- **Single Network Domain SSO:** You can set up OAM 11g single sign-on for resources within a single network domain (*mycompany.com*, for example). This includes protecting resources belonging to multiple WebLogic administration domains within a single network domain.
- **Multiple Network Domain SSO**: With OAM 11g, this is a standard feature. When 11g WebGates are used exclusively all cookies in the system are host-based. However, you must have control over all the domains. If some domains are controlled by external entities (not part of the OAM deployment), Oracle recommends that you use Oracle Identity Federation. For details, see *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*.
- **Multiple WebLogic Server Domain SSO**: The basic administration unit for WebLogic Server instances is known as a domain. You can define multiple WebLogic administration domains based on different system administrators' responsibilities, application boundaries, or the geographical locations of WebLogic servers. However, all Managed Servers in a cluster must reside in the same WebLogic Server domain.
- **SSO with Mixed Release Agents**: Oracle Access Manager 11g seamlessly supports registered OAM 11g and OAM 10g Agents, and OSSO Agents (mod_osso 10g), which can be used in any combination.

See Also:

- "OAM Servers and the Administration Console" on page 1-7
- "Policy Enforcement Agents" on page 1-8
- Chapter 7, "Introduction to the OAM Policy Model, Single Sign-On"

1.4.2 OAM Policy Model and Shared Policy Components

The Oracle Access Manager 11g policy model provides both authentication and authorization services within the context of an application domain.

Note: Oracle Access Manager 10g provides authentication and authorization services within the context of a policy domain. OracleAS SSO 10g provides only authentication.

In the Oracle Access Manager 11g policy model, the following components are shared and can be configured for use within any application domain:

- Resource Types: Defines the type of resource to be protected and the associated operations. The default resource type is HTTP. However, administrators can define non-http resource types that can be applied to specific resources in an application domain. The Access Tester can be used to evaluate policy enforcement for HTTP resources only.

- Host Identifiers: Simplifies the identification of a Web server host by enabling administrators to include all possible hostname variations within a named definition. When adding resources to an application domain, administrators can choose one of the named definitions and then specify the resource URL.

  Virtual Web Hosting: Enables support of multiple domain names and IP addresses that each resolve to their unique subdirectories on a single server. The same host can have multiple sites being served either based on multiple NIC cards (IP based) or multiple names (for example, abc.com and def.com) resolving to same IP.

- Authentication Schemes: Identifies the authentication level, challenge method and redirect URL, and the underlying authentication module to perform user authentication. When adding authentication policies to an application domain, administrators can choose one of the named authentication schemes to use with specified resources, as well as the success and failure URLs.

For more information about the policy model and shared components, see Chapter 8, "Managing Policy Components".

1.4.3 OAM Policy Model, Application Domains, and Policies

Application domains are the top-level constructs of the Oracle Access Manager 11g policy model. Each application domain provides a logical container for resources or sets of resources, and the associated policies that dictate who can access specific resources. Certain shared components are used within each application domain, as described in "OAM Policy Model and Shared Policy Components".

> **Note:** To enhance security, OAM 11g default behavior is to deny access when a resource is not protected by a policy that explicitly allows access. In contrast, OAM 10g default behavior allowed access when a resource was not protected by a rule or policy that explicitly denied access.
>
> OAM 10g provided authentication and authorization within the context of a policy domain. In contrast, OracleAS SSO 10g provides only authentication.

Each Oracle Access Manager 11g application domain includes the following elements:

- Resources

  Each resource definition in an application domain requires a Resource Type, Host Identifier (only for HTTP resources), and a URL to the specific resource. You can have as many resource definitions as you need in an application domain.

- Authentication Policies and Responses for Specific Resources

  Each authentication policy includes a unique name, one authentication scheme, success and failure URLs, one or more resources to which this policy applies, and administrator-defined responses to be applied after successful authentication.

  > **Note:** Depending on the OAM 11g policy responses specified for authentication or authorization success and failure, the end user might be redirected to a specific URL, or user information might be passed to other applications through a header variable or a cookie value.

- Authorization Policies, Constraints, and Responses for Specific Resources

Each authorization policy includes a unique name, success and failure URLs, and one or more resources to which this policy applies. In addition, administrators can define specific constraints (conditions) that must be fulfilled for a successful authorization and define responses to be applied after successful authorization.

> **Note:** OAM 10g enables authorization actions to be taken depending on the evaluation of the administrator-defined authorization expression contained one or more authorization rules.

For more information about the policy model and application domains, see Chapter 9, "Managing Policies to Protect Resources and Enable SSO".

1.4.4 Centralized Logout for OAM 11g

Oracle Access Manager 11g provides single logout (also known as global log out) for user sessions. With OAM, single logout refers to the process of terminating an active user session.

For details, see Chapter 11, "Configuring Centralized Logout for OAM 11g".

1.4.5 Connectivity and Policy Testing

Oracle provides a portable, stand-alone Java application that replaces the OAM 10g Access Tester function. The OAM 11g Access Tester simulates registered Agents connecting to OAM Servers. The scripted execution allows for command-line processing. You can record and playback scripts and capture output for different functions. Encrypted and multiple-server connections are supported.

You can use the Access Tester to troubleshoot agent to server connections in addition to on-the-fly testing of request and response semantics and access policy designs.

For details, see Chapter 10, "Validating Connectivity and Policies Using the Access Tester".

1.5 Session Management

Part IV of this book describes session management.

With OAM 11g, session management refers to the process of managing user session information with support for user- or administrator-initiated events, and time-out based events.

Administrators can configure Oracle Access Manager 11g session lifecycle settings. The database for session storage is initially configured with Oracle Access Manager configuration.

- In-memory Session Store: Uses embedded technology from Oracle Coherence to provide a distributed cache with low-data access latencies and to transparently move data between distributed caches (and the database policy store)
- Database Session Store: Provides fault-tolerance and scaleability for very large deployments (hundreds of thousands of simultaneous logins). In this case, you must be using a database policy and session-data store that is extended with the OAM-specific schema.

For more information, see Chapter 12, "Managing Sessions".

1.6 Logging and Auditing

This section introduces the information in Part V of this guide and includes the following topics:

- Component Event Message Logging
- Common Audit Framework
- Performance Metrics in the OAM Administration Console

1.6.1 Component Event Message Logging

Logging is the mechanism by which components write messages to a file to capture critical component events. Each Oracle Access Manager component instance writes process and state information to a log file.

You can configure logging to provide information at various levels of granularity. For instance, you can record errors, errors plus state information, or errors and states and other information to the level of a debug trace. You can also eliminate sensitive information from the logs. For more information, see Chapter 13, "Logging Component Event Messages".

You can also use a custom Oracle WebLogic Scripting Tool (WLST) command to change OAM logging levels.

> **See Also:** Appendix F, "Introduction to Custom WLST Commands for OAM Administrators" introduces custom WLST commands to change OAM logging levels

1.6.2 Common Audit Framework

With Oracle Access Manager 11g, auditing refers to the process of collecting for review specific information related to administrative, authentication, and run-time events. Auditing can help you evaluate adherance to polices, user access controls, and risk management procedures.

> **Note:** Auditing is not available for every Oracle Access Manager 11g component. However, logging is available for every OAM component.

Events are audited using the underlying Oracle Fusion Middleware Common Audit Framework. This framework uses a database audit store to provide scalability and high-availability for the audit framework. The database must include the audit schema.

> **Note:** The Oracle Fusion Middleware Common Audit Framework database audit store does not include OAM policy or session-data and is not configured through the OAM Administration Console.

Administrators can control and specify certain auditing parameters using the Oracle Access Manager Administration Console. Oracle Access Manager auditing configuration is recorded in the file `oam-config.xml`. Event configuration (mapping events to levels) occurs in the `component_events.xml`. An audit record contains a sequence of items that can be configured to meet particular requirements.

> **Note:** Oracle recommends that you use only the OAM Administration Console or WebLogic Scripting Tool (WLST) commands for changes; do not edit oam_config.xml.

Out-of-the-box, there are several sample audit reports available with Oracle Access Manager and accesible with Oracle Business Intelligence Publisher. You can also use Oracle Business Intelligence Publisher to create your own custom audit reports.

For more information, see Chapter 14, "Auditing OAM Administrative and Run-time Events".

1.7 Monitoring OAM Performance

Part VI of this book describes:

- Performance Metrics in the OAM Administration Console
- Performance Metrics in Fusion Middleware Control

1.7.1 Performance Metrics in the OAM Administration Console

Performance metrics can be collected in memory for components during the completion of particular events. You can monitor the time spent in a particular area or track particular occurrences or state changes.

OAM administrators monitor performance for Oracle Access Manager 11g using the Monitoring command in the OAM Administration Console.

For more information, see Chapter 15, "Monitoring OAM Metrics by Using Oracle Access Manager".

1.7.2 Performance Metrics in Fusion Middleware Control

Live, dynamic OAM performance metrics can be viewed in Fusion Middleware Control.

For more information, see Chapter 16, "Monitoring OAM Performance by Using Fusion Middleware Control".

1.8 Using OAM 10g WebGates with OAM 11g

This section introduces the information in Part VII of this guide and includes the following topics:

- Provisioning OAM 10g WebGates for OAM 11g
- Configuring 10g WebGates for Apache v2-based Web Servers (OHS and IHS)
- Configuring 10g WebGates for the IIS Web Server
- Configuring 10g WebGates for the ISA Server
- Configuring Lotus Domino for OAM 10g WebGates

1.8.1 Provisioning OAM 10g WebGates for OAM 11g

Everything you need to know about installing and using OAM 10g WebGates with OAM 11g is provided in Chapter 17, "Managing OAM 10g WebGates with OAM 11g".

1.8.2 Configuring 10g WebGates for Apache v2-based Web Servers (OHS and IHS)

Details about installing and configuringApache v2-based Web Servers (OHS and IHS) for OAM 10g WebGates with OAM 11g is provided in Chapter 18, "Configuring Apache, OHS, IHS for 10g WebGates".

1.8.3 Configuring 10g WebGates for the IIS Web Server

Details about installing and configuring IIS Web servers for OAM 10g WebGates with OAM 11g is provided in Chapter 19, "Configuring the IIS Web Server for 10g WebGates".

1.8.4 Configuring 10g WebGates for the ISA Server

Everything you need to know about configuring the ISA Server for OAM 10g WebGates with OAM 11g is provided in Chapter 20, "Configuring the ISA Server for 10g WebGates".

1.8.5 Configuring Lotus Domino for OAM 10g WebGates

Everything you need to know about installing and configuring Lotus Domino for use with OAM 10g WebGates and OAM 11g is provided in Chapter 21, "Configuring Lotus Domino Web Servers for 10g WebGates".

1.9 Appendixes

This section introduces the information in Part VIII of this guide and includes the following topics:

- Co-existence: OAM 11g SSO versus OAM 10g SSO with OracleAS SSO 10g
- Moving OAM 11g From Test (Source) to Production (Target)
- Integration with Oracle ADF Applications
- Internationalization and Multibyte Data Support for OAM 10g WebGates
- Secure Communication and Certificate Management
- Custom WebLogic Scripting Tool Commands for OAM
- OAM 11g for IPv6 Clients
- Troubleshooting

1.9.1 Co-existence: OAM 11g SSO versus OAM 10g SSO with OracleAS SSO 10g

Table 1–3 outlines several ways to use OAM 11g when you have various starting points.

Table 1–3 OAM 11g Co-existence Summary

If you have ...	To use OAM 11g SSO ...
OAM 10g integrated with OSSO 10g	You can upgrade the OSSO deployment to OAM 11g as introduced in Appendix B.

Table 1–3 (Cont.) OAM 11g Co-existence Summary

If you have ...	To use OAM 11g SSO ...
Web Servers other than Oracle HTTP Server	See Chapter 17 for details on: - Locating and Installing the Latest OAM 10g WebGate for OAM 11g: - Provisioning a 10g WebGate with OAM 11g - Configuring Centralized Logout for 10g WebGate with OAM 11g
OracleAS 10g SSO (OSSO) .	Use the Oracle-provided Upgrade Assistant, which scans the existing OracleAS 10g SSO server configuration, accepts as input the 10g OSSO policy properties file and schema information, and carries configured partner applications into the destination Oracle Access Manager 11g SSO. After running the upgrade assistant and performing post-upgrade tasks, existing partner apps (including Portal, Forms, Reports, and Discoverer) would be using OAM instead of OSSO as their SSO provider. Note: Existing mod_osso modules and OracleAS 10g SSO server partners can work seamlessly with OAM Servers and OAM 11g SSO. However, eventually all mod_osso modules should be replaced with OAM Agents to enable use of OAM 11g Authorization Policies. See Appendix B for an introduction to post-upgrade co-existence between OAM 11g and OSSO 10g Servers.

1.9.2 Moving OAM 11g From Test (Source) to Production (Target)

OAM 11g streamlines the transfer of configuration data from one deployment to another. For instance, from a small test environment to a larger production deployment (and vice versa).

For more information, see Appendix A, "Transitioning OAM 11g from a Test to a Production Environment".

1.9.3 Integration with Oracle ADF Applications

The Oracle Application Developer Framework (ADF) and applications that are coded to Oracle ADF standards interface with the OPSS SSO Framework. The Oracle Platform Security Services (OPSS) single sign-on framework provides a way to integrate applications in a domain with a single sign-on (SSO) solution.

You can integrate a Web application that uses Oracle ADF security and the OPSS SSO Framework with an Oracle Access Manager 11g SSO security provider for user authentication. For more information, see Appendix C, "Integrating Oracle ADF Applications with Oracle Access Manager 11g SSO".

1.9.4 Internationalization and Multibyte Data Support for OAM 10g WebGates

Appendix D, "Internationalization and Multibyte Data Support for OAM 10g WebGates" provides information on internationalization and multibyte data support.

1.9.5 Secure Communication and Certificate Management

With Oracle Access Manager 11g, credential collection occurs using the HTTP(S) channel; authorization occurs over the NetPoint Access Protocol (NAP) channel (also referred to as the Oracle Access Manager Protocol channel).

HTTP(S) Channel: Oracle recommends enabling the secure sockets layer (SSL) for communication across the HTTP(S) channel to transport credentials and to exchange security tokens. Both functions require signing or encryption with certificates.

Oracle Access Manager 11g provides a central component to manage certificates used across all Oracle Access Manager components, including WebGates.

NAP Channel: Oracle recommends using either Simple (Oracle-signed certificates) or Cert mode (outside certificate authority) to secure communication between WebGates and OAM Servers during authorization. Oracle provides a certificate import utility that you can use when you have signed certificates. For information, see Appendix E, "Securing Communication with OAM 11g".

Note: Oracle Access Manager 11g does provide support for customers who use self-signed certificates.

1.9.6 Custom WebLogic Scripting Tool Commands for OAM

OAM administrators can use custom WebLogic Scripting Tool (WLST) commands to perform certain configuration tasks.

For more information, see Appendix F, "Introduction to Custom WLST Commands for OAM Administrators".

1.9.7 OAM 11g for IPv6 Clients

Oracle Access Manager supports Internet Protocol Version 4 (IPv4). Oracle Fusion Middleware supports Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6). IPv6 is enabled with Oracle HTTP Server with the mod_wl_ohs plug-in.

For more information, see Appendix G, "Configuring OAM 11g for IPv6 Clients".

1.9.8 Troubleshooting

For tips and troubleshooting information, see Appendix H, "Troubleshooting".

2 Getting Started with OAM Administration and Navigation

This chapter describes the initial steps needed to log in and navigate around the Oracle Access Manager 11g Administration Console. This chapter includes the following topics:

- Prerequisites
- Introduction to Oracle Access Manager 11g Architecture
- Introduction to OAM Installation and Configuration
- Introduction to OAM Administrators
- Logging In to and Signing Out of Oracle Access Manager 11g
- Introduction to the OAM Administration Console and Controls
- Introduction to Policy Configuration and System Configuration Tabs
- Viewing Configuration Details in the Console
- Conducting Searches
- Using Online Help
- Command-Line Tools
- Logging Component Events

2.1 Prerequisites

All tasks in this book presume that you have Oracle Access Manager 11g deployed as described in the Oracle Fusion Middleware Installation Guide for Oracle Identity Management.

> **Note:** You can access the Oracle Access Manager Administration Console when the WebLogic Administration Server is running. If the OAM Administration Console is protected by a WebGate, the OAM Server must be running.

Before you begin tasks in this chapter:

- Learn about the Administration Console as described in "Introduction to the OAM Administration Console and Controls" on page 2-11.
- Verify the administrative LDAP group defined in the primary user identity store.

Note: The default LDAP group for both WebLogic and OAM administrators, "Administrator", is set during initial deployment using the Oracle Fusion Middleware Configuration Wizard, as described in "Introduction to OAM Administrators" on page 2-9.

2.2 Introduction to Oracle Access Manager 11g Architecture

Oracle Access Manager 11g provides a full range of Web perimeter security functions that include Web single sign-on; authentication and authorization; policy administration; auditing, and more.

- About Oracle Access Manager 11g Architecture
- Comparing Oracle Access Manager 11g with OAM 10g and OSSO 10g

2.2.1 About Oracle Access Manager 11g Architecture

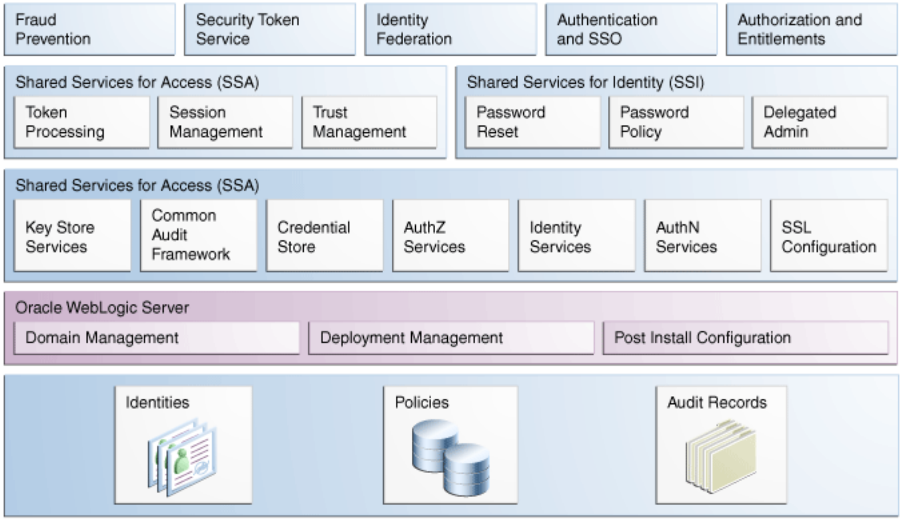
This topic provides an overview of Oracle Access Manager 11g, which sits on Oracle WebLogic Servers and is part of the Oracle Fusion Middleware Access Management architecture.

While providing backward compatibility and co-existence with existing solutions, Oracle Access Manager 11g replaces and converges:

- Oracle Access Manager 10g
- Oracle Application Server SSO (OSSO) 10g
- Oracle Sun OpenSSO

As illustrated in Figure 2–1, all user identities, policies, and audit records reside in centrally managed data stores. Oracle WebLogic Server provides domain management, deployment management, and post-installation configuration. Oracle Access Manager manages all agents and policies centrally.

Figure 2–1 Oracle Fusion Middleware Access Management Architecture



Both Oracle WebLogic Server and Oracle Access Manager 11g rely on the Oracle Platform Security Services for authentication, authorization, secure communication

(SSL), the common Audit Framework, as well as the credential store, and identity services.

Shared services for Access (SSA) include token processing, session management. Shared services for identity (SSI) include password reset, password policy, and delegated administration through Oracle Identity Manager.

Fraud prevention, security token service, identity federation, authentication and SSO, and authorization and Entitlements are integrated.

Figure 2–2 illustrates the primary Oracle Access Manager 11g components and services. The Protocol Compatibility Framework interfaces with OAM WebGates, mod_osso agents, and custom AccessGates created using the Access Manager Software Developer Kit (SDK).

Figure 2–2 Oracle Access Manager 11g Components and Services

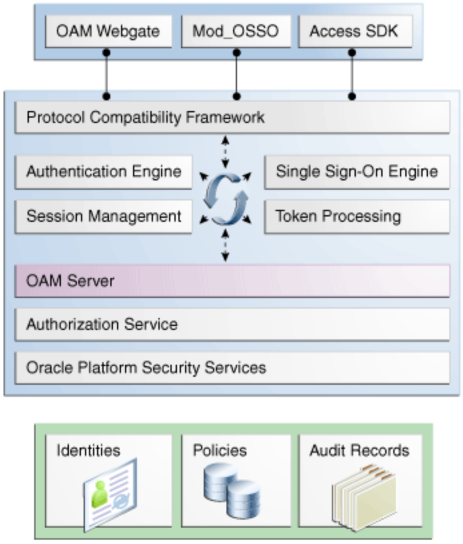
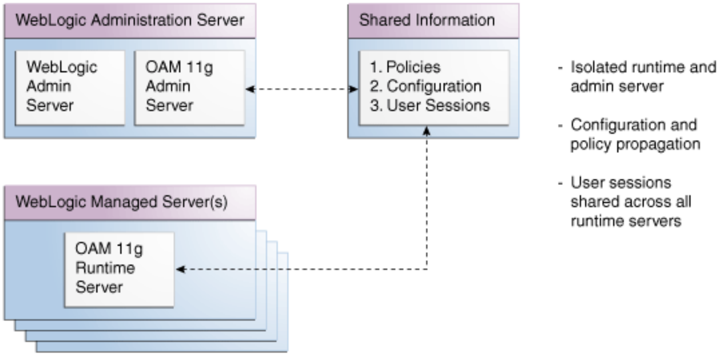


Figure 2–3 illustrates the distribution of Oracle Access Manager components.

Figure 2–3 Component Distribution


WebLogic Administration Server
WebLogic Admin Server
OAM 11g Admin Server
Shared Information
1. Policies
2. Configuration
3. User Sessions
- Isolated runtime and admin server
- Configuration and policy propagation
- User sessions shared across all runtime servers
WebLogic Managed Server(s)
OAM 11g Runtime Server

The Oracle Access Manager Administration Console (sometimes referred to as the Oracle Access Manager Admin Server) resides on the Oracle WebLogic Administration

Server (known as AdminServer). WebLogic Managed Servers hosting OAM runtime instances are known as OAM Servers.

Shared information consists of:

- Agent and server configuration data
- Oracle Access Manager policies
- User session data is shared among all OAM Servers

For more information, see "Comparing Oracle Access Manager 11g with OAM 10g and OSSO 10g".

2.2.2 Comparing Oracle Access Manager 11g with OAM 10g and OSSO 10g

This topic introduces Oracle Access Manager 11g architecture and provides a comparison against the 10g architecture for Oracle Access Manager and OSSO. Included are the following topics:

Oracle Access Manager 11g differs from Oracle Access Manager 10g in that the identity administration features have been transferred to Oracle Identity Manager 11g (including user self-service and self registration, workflow functionality, dynamic group management, and delegated identity administration).

Oracle Access Manager 10g supported Single Sign-on using a single session cookie (the ObSSOCookie) that contained the user identity and user session information required to access target resources that had the same or lower authentication level. The ObSSOCookie was encrypted and decrypted using a global shared secret key, the value of which was stored in the directory server. The ObSSOCookie was consumed by Access System components to verify the user identity and allow or disallow access to protected resources.

To close any possible security gaps, Oracle Access Manager 11g provides new server-side components that maintain backward compatibility with existing Oracle Access Manager 10g policy-enforcement agents (WebGates) and OSSO 10g agents (mod_osso). New Oracle Access Manager 11g WebGates are enhanced versions of 10g WebGates, that support a per-agent secret key for the Single Sign-on (SSO) solution. Thus, cookie-replay type of attack are prevented. The 11g WebGates are all trusted at the same level; a cookie specific for the WebGate is set and cannot be used to access any other WebGate-protected applications on a user's behalf.

Unless explicitly stated, the term "WebGate" refers to both an out of the box WebGate or a custom AccessGate.

Oracle Access Manager 11g uses technology from Oracle Coherence to provide centralized, distributed, and reliable session management.

For a list of names that have changed with Oracle Access Manager 11g, see "Product and Component Name Changes" on page xxxi. Table 2–1 provides a comparison of Oracle Access Manager 11g, OAM 10g, and OracleAS SSO 10g.

Table 2–1 Comparison: OAM 11g versus OAM 10g versus OSSO 10g

	OAM 11g	**OAM 10g**	**OSSO 10g**
Architecture Components	- Agents: WebGate, AccessGate, mod_osso, and IDM Domain Agent - OAM Server - OAM Administration Console (installed on WebLogic Administration Server) Note: Eight Administrator languages are supported.	- Resource WebGate (RWG) - Authentication WebGate (AWG) - Access Server - Policy Manager Note: Eight Administrator languages are supported.	- mod_osso (partner) - OracleAS SSO server (OSSO server)
Cookies	Host-based authentication cookie: - **11g WebGate, One per agent**: OAMAuthnCookie_ <host:port>_<random number> set by WebGate using the authentication token received from the OAM Server after successful authentication. **Note**: A valid OAMAuthnCookie is required for a session. - **10g WebGate**, One ObSSOCookie for all 10g WebGates. - **One for the OAM Server**: OAM_ID	- Domain-based ObSSOCookie for WebGates (including the AWG), for both authentication and session management	- Host-based authentication cookie: **one per partner**: OHS-*host-port* **one for OSSO server**: (but not with OAM 11g) - Domain-level session cookie for global inactivity timeout (GITO) if enabled (for interoperability with OAM 11g)
Cryptographic keys The protocols used to secure information exchange on the Internet.	- One per agent secret key shared between WebGate and OAM Server - One OAM Server key	One global shared secret key for all WebGates	- One key per partner shared between mod_osso and OSSO server - OSSO server's own key - One global key per OSSO setup for the GITO domain cookie
Key storage	- **Agent side**: A per agent key is stored locally in the Oracle Secret Store - **OAM 11g server side:** A per agent key, and server key, are stored in the credential store on the server side	Global shared secret stored in the directory server only (not accessible to WebGate)	- **mod_osso side**: partner keys and GITO global key stored locally in obfuscated configuration file - **OSSO server side**: partner keys, GITO global key, and server key are all stored in the directory server

Table 2–1 (Cont.) Comparison: OAM 11g versus OAM 10g versus OSSO 10g

	OAM 11g	OAM 10g	OSSO 10g
Encryption / Decryption (The process of converting encrypted data back into its original form)	Introduces client-side cryptography and ensures that cryptography is performed at both the agent and server ends: 1. WebGate encrypts obrareq.cgi using the agent key. **Note**: obrareq.cgi is the authentication request in the form of a query string redirected from WebGate to OAM Server. 2. OAM Server decrypts the request, authenticates, creates the session, and sets the server cookie. 3. OAM Server also generates the authentication token for the agent (encrypted using the agent key), packs it in obrar.cgi with a session token (if using cookie-based session management), authentication token and other parameters, then encrypts obrar.cgi using the agent key. **Note**: obrar.cgi is the authentication response string redirected from the OAM 11g server to WebGate. 4. WebGate decrypts obrar.cgi, extracts the authentication token, and sets a host-based cookie.	■ Token generation/ encryption, and validation/ decryption are delegated to the Access Server. ■ Both obrareq.cgi and obrar.cgi are sent unencrypted, relying on the underlying HTTP(S) transport for security.	Cryptography is performed at both mod_osso and OSSO server: 1. site2pstore token (request from mod_osso to server) is encrypted using the partner key locally at mod_osso. 2. OSSO server decrypts site2pstore token, authenticates, and generates its own cookie. 3. urlc token (the response from OSSO server to mod_osso) is encrypted using the partner key at the server. 4. mod_osso decrypts the urlc token locally and re-encrypts using its own format to set in a host-based cookie.
Session Management	■ OAM 10g session idle timeout behavior is supported through the Session Management Engine (SME). Session states are retained in memory	■ Single domain supported. **Multi-domain**: If a user idles out on one domain, but not on the authentication WebGate, the AWG cookie is still valid, re-authentication is not needed.A new cookie is generated with the refreshed timeout.	■ Single domain supported through a domain-level cookie for global inactivity timeout (GITO). **Multi-domain SSO**: After a user logs in to one domain, and then goes to a different domain, he is considered idle from the first domain, When the idle times out on the original domain, the user must re-authenticate on the original domain.

Table 2–1 (Cont.) Comparison: OAM 11g versus OAM 10g versus OSSO 10g

	OAM 11g	OAM 10g	OSSO 10g
Client IP	■ Maintain this ClientIP, and include it in the host- based OAMAuthnCookie.	■ Include the original clientIP inside the ObSSOCookie. If IP validation is configured, when cookie presented in later authentication or authorization requests this original clientIP is compared with the presenter's IP. Rejection occurs if there is no match	■ Include the original clientIP inside the host cookie. In later authentication requests, when the cookie is presented, the original clientIP is compared with the presenter's IP. Rejection occurs if there is no match
Response token replay prevention	■ Include RequestTime (the timestamp just before redirect) in obrareq.cgi and copy it to obrar.cgi to prevent response token replay.	N/A	■ Include RequestTime (timestamp just before redirect) in the site2pstore token and copy it to the urlc token to prevent token replay.

Centralized log-out	■ The `logOutUrls` (OAM 10g WebGate configuration parameter) is preserved.
	■ New 11g WebGate parameters are provided:
	`logoutRedirectUrl`
	`logoutCallbackUrl`
	`Logout Target URL`
	For more information, see Chapter 11.
		■ Single domain is supported.
		Once a user logs off from one WebGate, the domain cookie is cleared and the user is considered to be logged off the entire domain.
		■ Multi-domain SSO can be supported through chained customized logout pages.
			The OSSO server cookie includes a list of partner IDs.
			When a user logs off from one partner application:
			1. OSSO server pulls a list of the logout URLs.
			2. OSSO server clears its own cookie.
			3. OSSO server redirects to a customized JSP page (hosted on the OSSO server), and passes the list of logout URLs in the request.
			4. The JSP page loads those logout URLs that contains some image tags of check marks, and as a result of the loading, the cookies for those mod_osso instances are cleared

2.3 Introduction to OAM Installation and Configuration

This section provides a brief overview of OAM deployments and installation:

- About Deployment Types and OAM
- About Post-Installation Tasks

2.3.1 About Deployment Types and OAM

Table 2–2 describes the types of deployments you might have within your enterprise, even though these might be named differently in your enterprise.

Table 2–2 Deployment Types

Deployment Type	Description
Development Deployment	Ideally a *sandbox*-type setting where the dependency on the overall deployment is minimal
QA Deployment	Typically a smaller shared deployment used for testing
Pre-production Deployment	Typically a shared deployment used for testing with a wider audience
Production Deployment	Fully shared and available within the enterprise on a daily basis

During initial installation and configuration you can create a new WebLogic Server domain (or extend an existing domain) and define information for OAM Servers, Database Schemas, optional WebLogic Managed Servers and clusters, and the embedded LDAP Server.

> **See Also:** The "Understanding Oracle WebLogic Server Domains" chapter in the *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server* guide provides information about Oracle WebLogic Server administration domains.

Regardless of the deployment size or type, in a new WebLogic Server domain the following OAM-related components are deployed using the Oracle Fusion Middleware Configuration Wizard:

- WebLogic Administration Server
- Oracle Access Manager Console deployed on the WebLogic Administration Server (sometimes referred to as the OAM Administration Server)
- A WebLogic Managed Server for Oracle Access Manager
- Application deployed on the Managed Server

> **Note:** In an existing WebLogic Server domain, the WebLogic Administration Server is already installed and operational.

While using the Oracle Fusion Middleware Configuration Wizard, the **with-DB config template** was chosen to set up the database for application domain metadata. The database must be extended with the OAM-specific schema using the Repository Creation Utility (RCU). The policy store bootstrap occurs on the initial AdminServer startup after running the Configuration Wizard. For more information, see the Oracle Fusion Middleware Installation Guide for Oracle Identity Management.

The default Embedded LDAP is set as the primary user identity store for OAM 11g.

A Java key store is set up to be used for certificates for Simple or Certificate-based communication between OAM Servers and WebGates during authorization. The key store bootstrap also occurs on the initial AdminServer startup after running the Configuration Wizard.

2.3.2 About Post-Installation Tasks

During initial deployment, the WebLogic Administrator userID and password are set for use when signing in to both the OAM Administration Console and WebLogic Server Administration Console. A different administrator can be assigned for OAM, as described in "Introduction to OAM Administrators" on page 2-9.

OAM administrators can log in and use the OAM Administration Console to manage:

- User identity stores
- OAM Server registration
- Partner (agent and partner application) registration
- Application domains and policies to protect resources
- User sessions
- Common Server Properties

2.4 Introduction to OAM Administrators

Only users with sufficient privileges can log in to the Oracle Access Manager Administration Console or use OAM administrative command-line tools such as the remote registration tool or WLST. The WebLogic Scripting Tool (WLST) is a command-line scripting environment that can be used to manage, and monitor WebLogic Server domains. Administrators can also use customized OAM WLST commands to perform a number of tasks.

> **See Also:** Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"

During initial deployment, the administrator userID and password are set. By default, access to the OAM Administration Console is provided using the WebLogic Server "Administrators" group. These credentials provide access to both the WebLogic Server Administration Console and the Oracle Access Manager Administration Console.

> **Note:** Initially, administrative users must log in to the OAM Administration Console using the WebLogic Administrator credentials set during initial OAM configuration.

Table 2–3 describes the administrator Role that is recognized by Oracle Access Manager and WebLogic, and the default LDAP group to which the Role is mapped in the primary user identity store.

Table 2–3 Role Mapping from an LDAP Group to OAM Administrator

OAM Administrator Role	Description and LDAP Group
OAM Administrator's Role	The LDAP group defined within the primary user identity store that grants users full OAM system and policy configuration privileges. Default Group = Administrators **Note**: Specifying a different LDAP group prohibits WebLogic administrators from logging in to OAM or from using OAM administrative command-line tools.

Your enterprise might require independent sets of administrators: one set of users responsible for OAM administration and a different set for WebLogic administration. For more information, see "Defining a New OAM Administrator Role" on page 3-9.

2.5 Logging In to and Signing Out of Oracle Access Manager 11g

This section describes how to log in to and sign out of the Oracle Access Manager Administration Console directly.

This section provides the following topics:

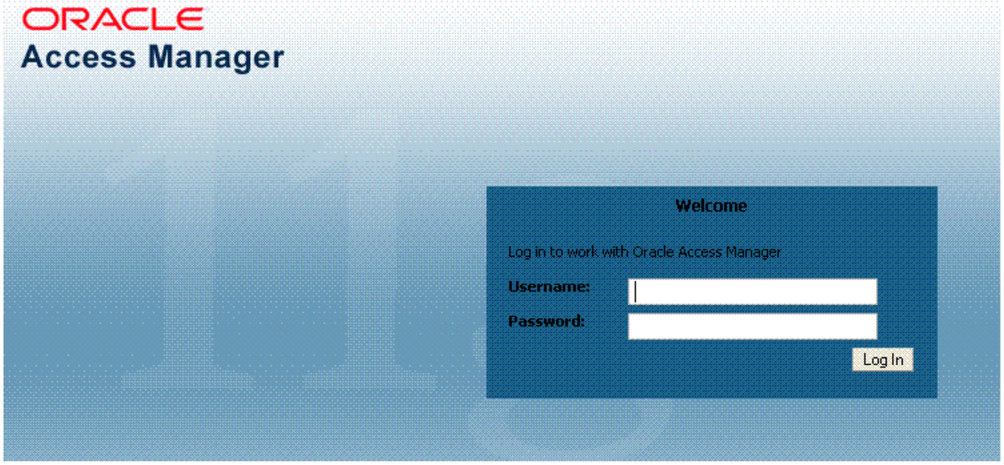
- Logging In to the Oracle Access Manager 11g Administration Console
- Signing Out of Oracle Access Manager 11g Administration Console

> **Note:** If you have Oracle Identity Navigator installed to access multiple consoles from one URL, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Navigator*.

2.5.1 Logging In to the Oracle Access Manager 11g Administration Console

The OAM log in page is shown in Figure 2–4.

Figure 2–4 Oracle Access Manager 11g Log In Page



Note: Ensure that you use the correct administrative credential for log in. Initially, the LDAP group for the OAM Administrator is the same as the LDAP group defined for the WebLogic Server Administration Console ("Administrators") and the primary user identity store is the WebLogic Embedded LDAP.

To log in to Oracle Access Manager 11g

1. In a browser window, enter the URL to the Oracle Access Manager 11g using the appropriate protocol (HTTP or HTTPS). For example:

   ```
   https://hostname:port/oamconsole/
   ```

   In the sample URL shown here:

   - HTTPS represents the Hypertext Transfer Protocol (HTTP) with the Secure Socket Layer (SSL) enabled to encrypt and decrypt user page requests and the pages returned by the Web server
   - *hostname* refers to fully-qualified domain name of the computer hosting the Oracle Access Manager 11g Administration Console
   - *port* refers to the designated bind port for the OAM Administration Console (this is the same as the bind port for the WebLogic Server Administration Console)
   - /oamconsole/refers to the OAM Administration Console Log In page

2. On the Log In page, enter the OAM Administrator credentials. For example:

   Username: *Admin_login_id*

   Password: *Admin_password*

3. Click the Log In button or press the Enter key.
4. Proceed as follows:

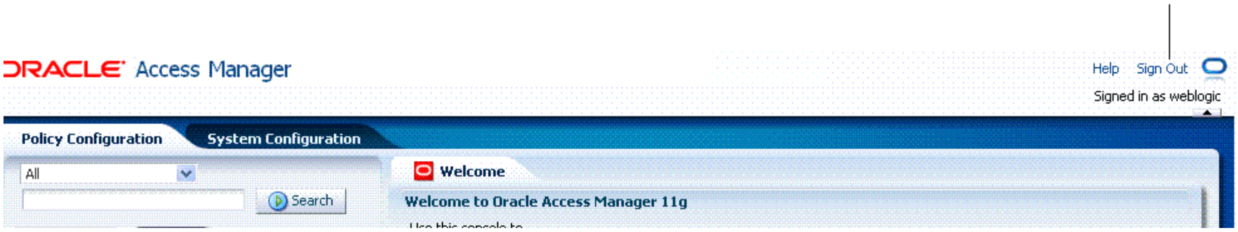
- **Successful**: Policy Configuration, and System Configuration tabs appear on the left; Welcome page is on the right. Tour the console, as described in "Introduction to Policy Configuration and System Configuration Tabs" on page 2-20 or start performing tasks on your own.
- **Not Successful**: Log in again and ensure that you enter information exactly as specified for the OAM Administrator in the primary user identity store.

See Also: "Introduction to OAM Administrators" on page 2-9

2.5.2 Signing Out of Oracle Access Manager 11g Administration Console

The Sign Out link appears in the upper-right corner of the Administration Console, as shown in Figure 2–5. You select the Sign Out link to conclude your session. Oracle recommends that you also close the browser window after signing out.

Figure 2–5 Sign Out Link, Upper-right Corner



To sign out of Oracle Access Manager 11g Administration Console

1. Click the Sign Out link in the upper-right corner of the console.
2. Close your browser window.

2.6 Introduction to the OAM Administration Console and Controls

The Oracle Access Manager Administration Console is a Web-based program that provides function-level tabs and controls, as well as page-level tabs and controls. This section introduces the Oracle Access Manager 11g Administration Console.

The Oracle Access Manager 11g Administration Console provides the system and policy configuration management functions required by administrators. You can enter the URL to the Oracle Access Manager 11g console in a browser window:

```
https://hostname:port/oamconsole
```

In the sample URL, *hostname* refers to computer that hosts the Oracle Access Manager 11g Administration Console; *port* refers to the HTTP port number on which the console host listens; /oamconsole refers to the Log In page.

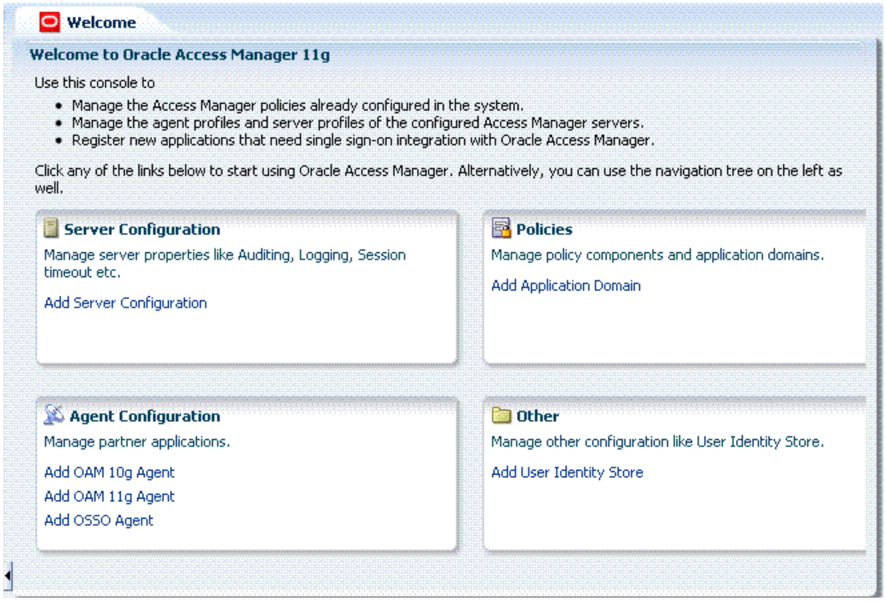
This section provides a quick introduction to orient you to the Oracle Access Manager Administration Console.

- Console Layout and Controls
- Elements on a Page
- Selecting Controls in the Administration Console

2.6.1 Console Layout and Controls

Figure 2–6 provides a look at the Administration Console as it appears immediately after log in.

Figure 2–6 OAM Administration Console Welcome Page and Policy Configuration Tab



Welcome

Welcome to Oracle Access Manager 11g

Use this console to

- Manage the Access Manager policies already configured in the system.
- Manage the agent profiles and server profiles of the configured Access Manager servers.
- Register new applications that need single sign-on integration with Oracle Access Manager.

Click any of the links below to start using Oracle Access Manager. Alternatively, you can use the navigation tree on the left as well.

Server Configuration

Manage server properties like Auditing, Logging, Session timeout etc.

Add Server Configuration

Policies

Manage policy components and application domains.

Add Application Domain

Agent Configuration

Manage partner applications.

Add OAM 10g Agent

Add OAM 11g Agent

Add OSSO Agent

Other

Manage other configuration like User Identity Store.

Add User Identity Store

The OAM 11g Administration Console provides named function tabs on the left above the search controls and a menu and tool bar above the navigation tree. Open pages appear on the right. Currently the Welcome page is open.

Following topics provide more information:

- Welcome Page
- Function-Level Tabs and Controls
- Content Pages and Page Controls

> **See Also:** "Selecting Controls in the Administration Console" on page 2-19

2.6.1.1 Welcome Page

Initially, the Welcome page is open and active on the right side of the console. Sections on the Welcome page include a brief description of a specific function and one or more "shortcuts" (links that you can select) to initiate certain tasks immediately as explained in Table 2–4.

Table 2–4 Welcome Page Quick Pick Sections

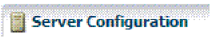
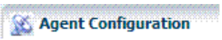
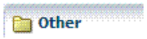
Quick Pick Section	Description
Server Configuration  Server Configuration	Click **Add Server Configuration** to launch a fresh Create: OAM Server page. See "About the System Configuration Tab" on page 2-20 for more information.

Table 2–4 (Cont.) Welcome Page Quick Pick Sections

Quick Pick Section	Description
Policies 	Click **Add Application Domain** to launch a fresh Application Domains page. See "About the Policy Configuration Tab" on page 2-22 for more information.
Agent Configuration  	■ Click **Add OAM 10g Agent** to launch a fresh Create: OAM Agent page. ■ Click **Add OAM 11g Agent** to launch a fresh Create: OAM Agent page. ■ Click **Add OSSO Agent** to launch a fresh Create: OSSO Agent page.
Other  	Click **Add Identity Store** to launch a fresh Create: User Identity Store page.

2.6.1.2 Function-Level Tabs and Controls

Table 2–5 introduces the function-level tabs in the OAM 11g Administration Console.

Table 2–5 Function Tabs and Descriptions

Function Tab Name	Description
Policy Configuration 	Provides access to definitions for Shared Components and Application Domains. This tab is active and the related navigation tree is visible for browsing on the left side of the screen when you enter the console. See "About the Policy Configuration Tab" on page 2-22 for more information.
System Configuration 	Provides access to system-level definitions for Agents, Servers, and Data Sources. This is not the active tab when you enter the console, which is why it appears a different color. See "About the System Configuration Tab" on page 2-20 for more information.
Browse 	Provides the navigation tree from which you can access nodes and instances related to the active configuration tab (Policy or System). This tab is active when you enter the Administration Console.
Search Results 	Provides access to the results of your latest search. Search controls appear above the Browse and Search Results tabs. For more information, see "Conducting Searches" on page 2-24.

The following topics provide more information about specific controls:

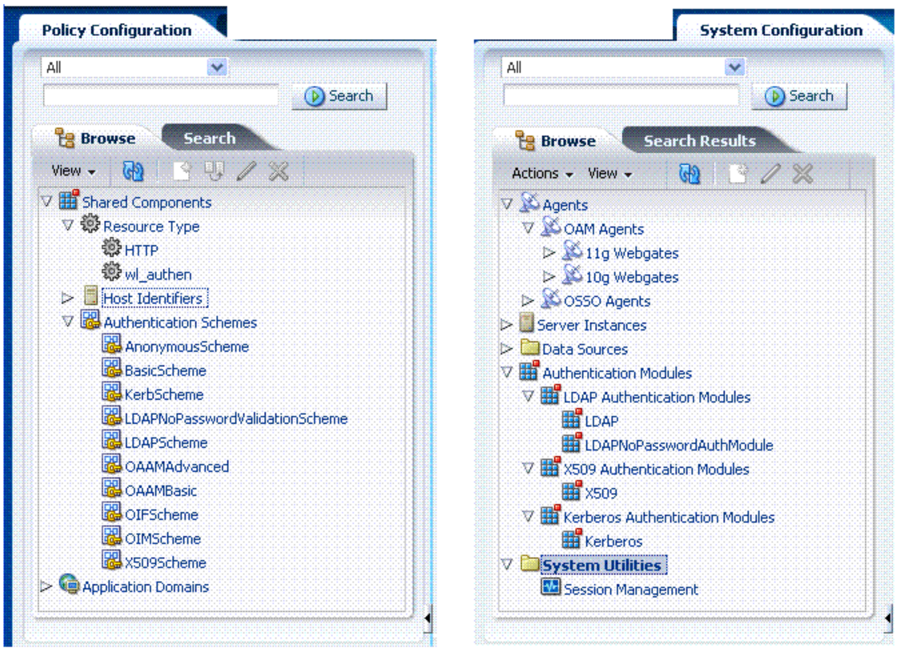
- Browse Tab and Navigation Tree
- Menu and Tool Bar
- View Menu
- Actions Menu

> **See Also:** "Selecting Controls in the Administration Console" on page 2-19

2.6.1.2.1 Browse Tab and Navigation Tree When the Browse tab is selected, the navigation tree for the active configuration tab (System Configuration or Policy Configuration tab) is visible. Named nodes identify groups under which you can choose individual instances on which to take action.

The nodes in the navigation tree for the Policy Configuration and System Configuration tabs are shown in Figure 2–7.

Figure 2–7 Policy Configuration and System Configuration Navigation Trees



For more information, see the following topics:

- "About the System Configuration Tab" on page 2-20
- "About the Policy Configuration Tab" on page 2-22

2.6.1.2.2 Menu and Tool Bar A menu and tool bar appears above the navigation tree, as shown in Figure 2–8. Menus provide commands that you can use to take action on the selected item in the navigation tree. Many menu commands are also provided as command buttons in the tool bar for quick access.

Figure 2–8 Menu and Tool Bar Above the Navigation Tree

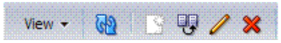


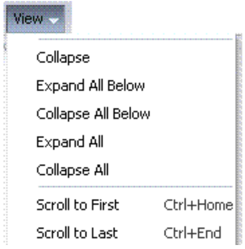
Table 2–6 provides a description of each command button in the tool bar. Buttons appear in color when they are available. When a command cannot be used, the command button (or menu item) appears in grey.

Table 2–6 Command Buttons in the Tool Bar

Button	Definition	Description
↻	Refresh	Revives the navigation tree, in the same way a Web browser refreshes a Web page.
🗋	Create	Opens a fresh page under the selected node in the navigation tree, which you can fill in to add a new configuration of the selected type. The new page opens as the active page on the right side of the navigation tree. This is available when you can add a new configuration, for instance, under Server Instances, or a specific Agent type, or a user identity store, or a non-HTTP Resource Type or Host Identifier or Application Domain. **Alternatively**, use the Create command on the Actions menu as described in Table 2–8.
⧉	Duplicate	Creates a copy of the selected configuration in the navigation tree, named "copy of *original*." The copy opens as the active page for immediate editing. Many fields are filled in. **Exception**: Fields that make up the unique identifier of the object (for example, Name of the policy or the URL pattern of a resource) are not automatically filled in. **Note**: You edit and save the duplicate as usual.
✎	Edit	Opens the instance you have selected in the navigation tree, to view or modify. The configuration page opens as the active page on the right side of the navigation tree. **Alternatively**, double click the instance name to display a page for editing.
✖	Delete	Removes the selected configuration. A deleted configuration is removed from the navigation tree and is no longer accessible to the system. For instance, if you delete an Agent configuration, the Agent is no longer registered and cannot be used. **Alternatively**, use the Delete command on the Actions menu as described in Table 2–8.
Detach	Detach	Separates the selected item (a results table on a configuration page, for instance) and displays it alone as a full page. Note: If you are viewing a detached table, you can click this button to re-attach it to the corresponding page and restore the standard page view.

2.6.1.2.3 View Menu Figure 2–9 illustrates the View menu, which is available for use with both the Policy Configuration tab and the System Configuration tab.

Figure 2–9 View Menu



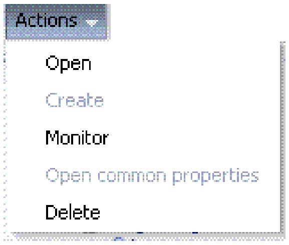
Unavailable items (those that cannot be used on the selection in the navigation tree) appear in grey. View menu command descriptions are provided in Table 2–7.

Table 2–7 View Menu Command Descriptions

Command	Description
Expand	Immediately reveal items within the selected node in the navigation tree. This does not open or activate a configuration page. **Alternatively**, click the icon beside the node in the navigation tree.
Collapse	Immediately conceal everything within the selected node in the navigation tree. This does not close an open page. **Alternatively**, click the icon beside the node in the navigation tree.
Expand All Below	Immediately reveal everything within the selected node. For example, click Application Domains and then click Expand All Below to see all application domains.
Collapse All Below	Immediately close the selected node and conceal its content. This does not close an open page.
Expand All	Immediately reveal all nodes and instances in the navigation tree. This has no impact on open pages.
Collapse All	Immediately conceal all nodes and instances in the navigation tree. This has no impact on open pages.
Scroll to First Ctrl+Home	Locates and displays the first item in the navigation tree or results table.
Scroll to Last Ctrl+End	Locates and displays the last item in the navigation tree or results table.

2.6.1.2.4 Actions Menu This menu is available only when the System Configuration tab is active. Figure 2–9 illustrates the Actions menu, which provides appropriate commands for the selection in the navigation tree. For instance, if you have Server Instances selected in the navigation tree one of the commands on the Actions menu enables you to open the Server Common Properties page for viewing or editing.

Figure 2–10 Actions Menu



Actions menu command descriptions are provided in Table 2–8. Certain commands on this menu mirror functions that are available by using command buttons in the tool bar. Unavailable items (those that cannot be used on the selection in the navigation tree) appear in grey.

Table 2–8 System Configuration, Action Menu, Command Descriptions

Command	Description
Open	Opens the configuration page for the selected instance in the navigation tree. This is not available when you have a node selected in the navigation tree. **Alternatively**, double-click the instance name in the navigation tree to open a page.
Create	Activates a fresh page that you can fill in to define a new configuration. **Alternatively**, click the Create button in the tool bar as described in Table 2–6.

Table 2–8 (Cont.) System Configuration, Action Menu, Command Descriptions

Command	Description
Monitor	Displays the monitoring page for the Agent selected in the navigation tree. For more information, see Chapter 15.
Open common properties	Opens the OAM Server Common Properties page, which provides various functional configurations shared among all OAM servers. This is available only when the Server Instances node is selected in the navigation tree.
Delete	Removes the selected instance registration. The deleted registration is removed from the navigation tree and is no longer accessible to the system. For instance, if you delete an agent registration, the Agent is no longer registered and cannot be used. **Alternatively**, click the Delete button in the tool bar as described in Table 2–6.

2.6.1.3 Content Pages and Page Controls

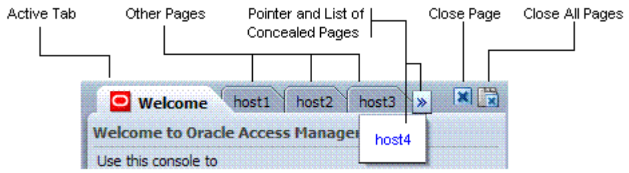
Like the Welcome page, any open content pages appear on the right side of the console.

The active content page is visible and generally provides a work space where you can add, view, or modify related settings. A named tab identifies each open page, like the tabs on manila folders. The tab of the active page is white.

Up to ten pages can be open simultaneously per configuration tab: Policy Configuration tab or System Configuration tab. Only the named tabs of opened pages for the currently active configuration tab are shown.

Only the active page is visible, with as many named tabs of other open pages that can fit on one line. You can click a named tab to activate the corresponding page. When named tabs of open pages do not fit on one line, a pointer is provided that enables you to open and choose from a list of concealed pages. Figure 2–11 illustrates multiple pages open at the same time. You can see named tabs for each page and controls to access pages that are concealed (or to close the active page or close multiple pages).

Figure 2–11 Tabs of Open Pages, and Page Controls



Each page appears only once. No warning is issued if you attempt to open the same page multiple times. However, the page is only one time.

> **Note:** There is no warning if you open the page for the same item more than once.

The controls that you can use with open pages are described in Table 2–9.

> **See Also:** "Selecting Controls in the Administration Console" on page 2-19

Table 2–9 Controls for Open Pages

Page Control	Definition	Description
	View a list of concealed pages	Click the pointer to view the list of concealed pages when you cannot view all tabs simultaneously.
☒	Close Active Page	Click this button to close the active page. **Note**: Closing a page before clicking Apply discards any changes or additions without warning. The changes are lost. You can use this to cancel changes you do not want to retain.
☒	Close Multiple Pages	▪ Click this button to initiate closing multiple open pages. ▪ In the dialog box that appears, click the box beside the name of each page you want to close. ▪ Click OK to complete the action. **Note**: Closing a page before clicking Apply discards any changes or additions without warning. The changes are lost. You can use this to cancel changes you do not want to retain.

2.6.2 Elements on a Page

Pages in the Administration Console contain one or more graphical user interface elements as described in Table 2–10. For an example of each element n the Administration Console, see Figure 2–13 or log into the console and have a look.

Table 2–10 Page Elements and Descriptions

Page Element	Description
Named tab	Identifies each open page on the right side of the console. Also, displays a page of related, lower-level settings. See Figure 2–14 for an example.
Page controls	Enables you to close one or more pages. See Table 2–9.
Apply button	Submits changes or additions made to the page.
Named text box	Enables you to enter relevant details in the named field using the keyboard.
Option button	Enables you to choose one of several options. For example, you can click an option button to define a state (Enabled vs. Disabled) or a security mode (Open vs. Simple vs. Cert).
Tables	Displays current specifications or space for new specifications. Tables have independent command buttons independent from page-level and option buttons.
Command buttons for tables ✚ ✖	Enables you to: Add a fresh row. Remove the selected row.
Drop down lists	Provides a menu of choices on certain pages (and as part of the Search controls). You can choose one item from those listed.

2.6.3 Selecting Controls in the Administration Console

This section describes how to select the desired node or instance in the navigation tree, and selecting commands and page controls in the Administration Console. The usual selection guidelines apply.

Table 2–11 describes selections and controls.

Table 2–11 Selection Tasks and Controls

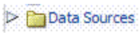
Task	Control	Description
Expand a node	 ▷ Data Sources	Click the Expand button beside the desired node in the navigation tree to reveal nodes or instances within it.
Collapse a node	▽ Agents	Click the Collapse button beside the desired node in the navigation tree to conceal nodes or instances beneath it.
Display View menu	Right-click mouse button	Right-click the desired node in the navigation tree to display a pop-up View menu.
Select	Click mouse button	Click the desired item on which to operate. For example, click the desired: - Icon, node, or instance name in the navigation tree (Shared Components is one example) - Search Button: Initiates a search based on specified criteria - Menu name and command to take action on the selected item in the navigation tree - Command button to take immediate action: Menu and tool bar buttons (Table 2–6) Close page buttons (Table 2–9) - Command Button on a Page or Table: Apply: Submits additions and changes on the active page. Table or section buttons (Table 2–10) ✚ Add a new row. ✖ Remove the selected row. - Links: Help, and Sign Out are examples
Activate	Click mouse button	Click to activate the desired: - Function tab: System Configuration, Policy Configuration, Browse, Search - Named tab on a page to reveal related lower-level settings to view or modify: for instance, SSO Engine, Session, and so on - Named Page tab to reveal (activate) the page - Text field to enter information on a page - Page Control (close or close all as described in Table 2–9)

Table 2–11 (Cont.) Selection Tasks and Controls

Task	Control	Description
Open	Double click mouse button	Double-click an instance name to open the configuration page. For example, double-click a specific: ■ Resource Type name ■ Host Identifier definition name ■ Authentication scheme name ■ Resource name in an application domain ■ Authentication policy name in an application domain ■ Authorization policy name in an application domain ■ Agent instance name ■ Server instance name ■ User identity store instance name ■ Database instance name ■ Authentication module name ■ System utility name
Highlight	Drag cursor	Drag the cursor across text in a box to highlight its content.

2.7 Introduction to Policy Configuration and System Configuration Tabs

This section provides a quick tour to orient you to major Oracle Access Manager functions:

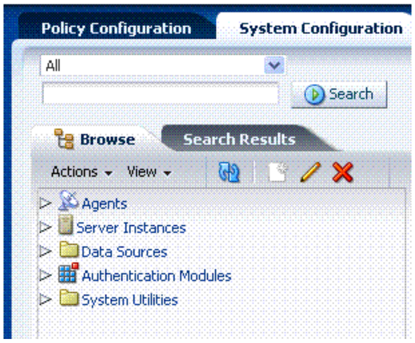
- About the System Configuration Tab
- About the Policy Configuration Tab

2.7.1 About the System Configuration Tab

Figure 2–12 shows the console. The Policy Configuration and System Configuration tabs appear on the left. Search controls appear directly beneath the Policy Configuration and System Configuration tabs. The navigation tree for the active (White) tab is identified by a Browse tab, which appears directly beneath the Search controls.

A tool bar separates the Browse and Search Results tabs from the navigation tree. The Actions menu is available only with the System Configuration tab; the View menu is always available. The active page appears on the right. The Welcome page is currently the active and open.

Figure 2–12 System Configuration Tab and navigation tree



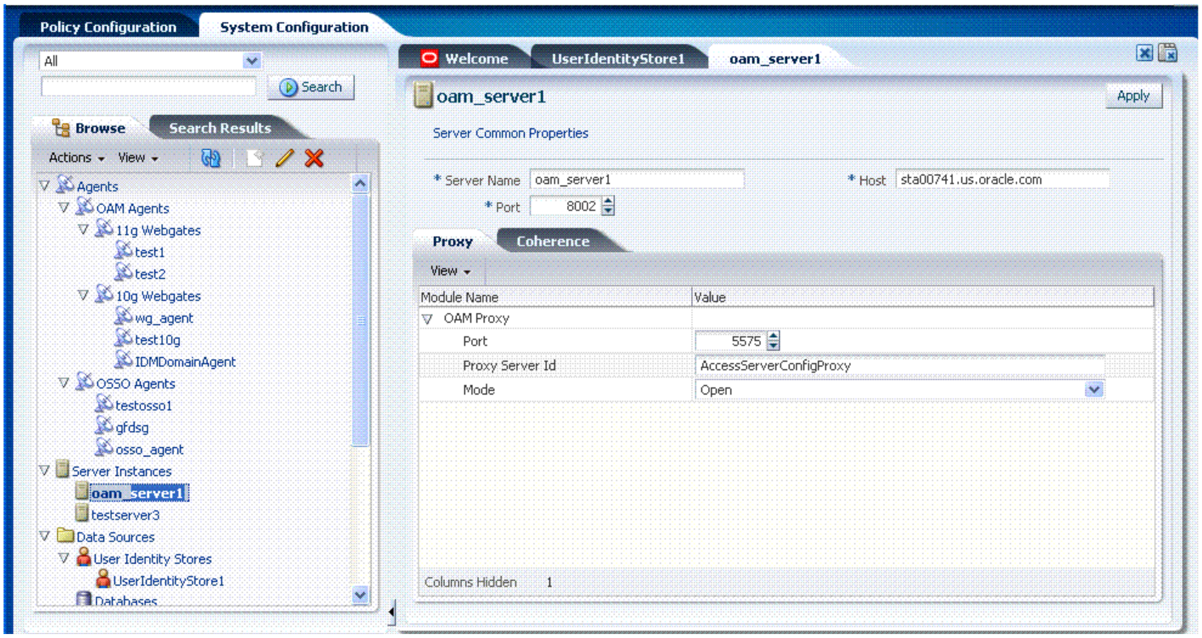
The System Configuration tab is currently active. It gives administrators access to Agent, Server, Data Source, and Authentication Module configuration details, and System Utilities. The navigation tree beneath the search controls and tool bar is related to the active tab.

See "Console Layout and Controls" on page 2-12 for details on navigating and selecting command buttons, page controls, and menu items in the console.

You can also use commands on the View menu to expand the selected node in the navigation tree or to expand all nodes simultaneously. For instance, click Expand All from the View menu to see all nodes and related instances at one time.

Figure 2–13 provides an expanded view of nodes and instances on the System Configuration tab, navigation tree. A server instance is selected in the tree and the related configuration page appears on the right.

Figure 2–13 System Configuration: Expanded Tree (Left), Active Page (Right)



oam_server1

Server Common Properties

* Server Name: oam_server1

* Host: sta00741.us.oracle.com

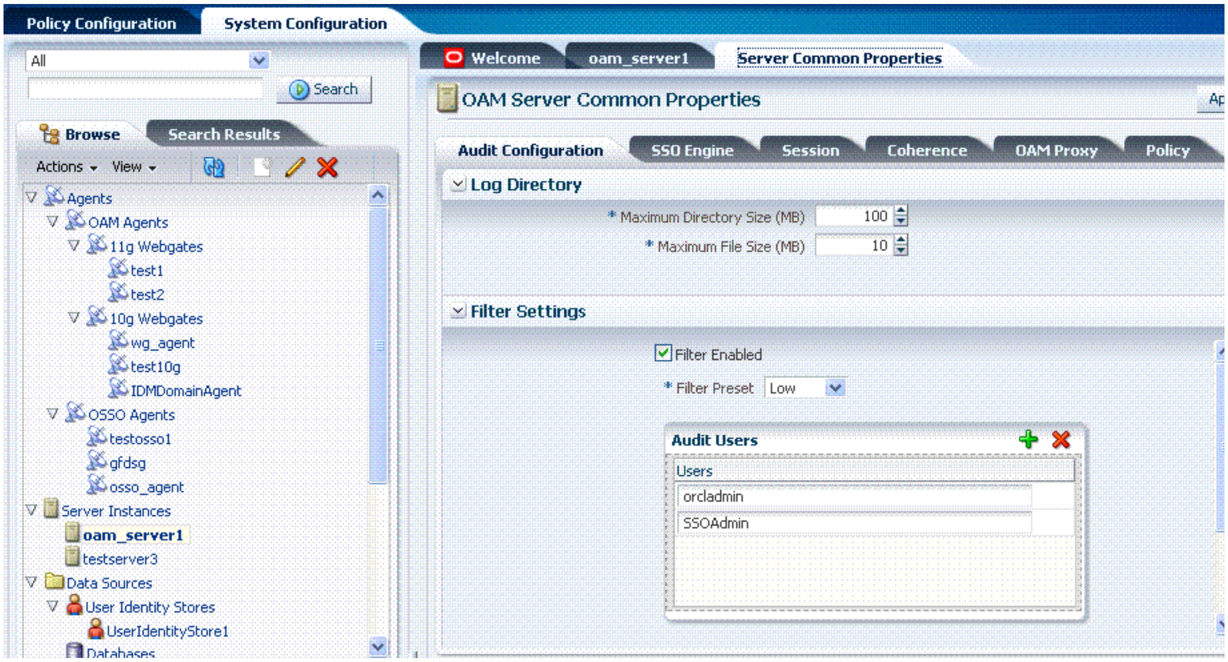
* Port: 8002

Module Name	Value
OAM Proxy	
Port	5575
Proxy Server Id	AccessServerConfigProxy
Mode	Open

Columns Hidden 1

Figure 2–14 shows the OAM Server Common Properties page, which provides tabs for Auditing, SSO Engine, Session, Coherence, OAM Proxy, and Policy configuration details. This group of definitions is common to all Oracle Access Manager Servers. The Auditing Configuration tab is active and the relevant page is open on the right.

Figure 2–14 OAM Server Common Properties



* Maximum Directory Size (MB) 100

* Maximum File Size (MB) 10

☑ Filter Enabled

* Filter Preset Low

Users
orcladmin
SSOAdmin

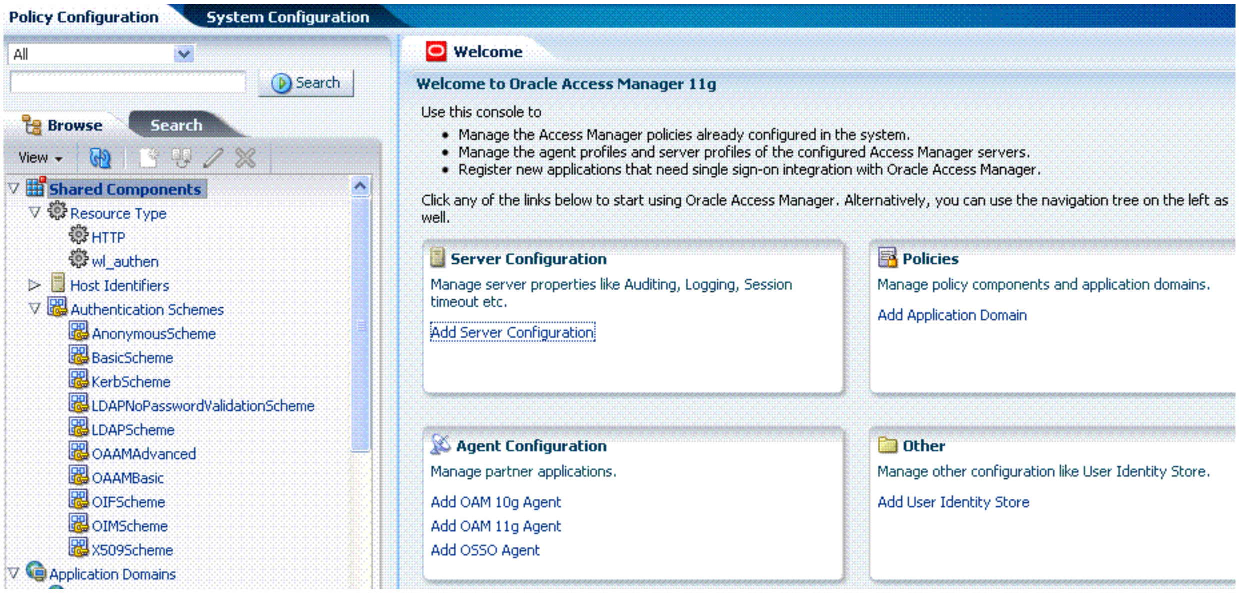
For more information about system configuration, see:

- Part II, "OAM 11g System Management"
- Part III, "Single Sign-on, Policies, and Testing"

2.7.2 About the Policy Configuration Tab

The Policy Configuration tab in the Oracle Access Manager Administration Console gives administrators access to application domain and shared component configurations. The view in Figure 2–15 lists first-level items beneath Shared Components and Authentication Schemes.

Figure 2–15 Policy Configuration Tab, Navigation Tree, and Active Page



Welcome to Oracle Access Manager 11g

Use this console to

- Manage the Access Manager policies already configured in the system.
- Manage the agent profiles and server profiles of the configured Access Manager servers.
- Register new applications that need single sign-on integration with Oracle Access Manager.

Click any of the links below to start using Oracle Access Manager. Alternatively, you can use the navigation tree on the left as well.

Manage server properties like Auditing, Logging, Session timeout etc.

Manage policy components and application domains.

Manage partner applications.

Manage other configuration like User Identity Store.

See "Console Layout and Controls" on page 2-12 for details on navigating and selecting command buttons, page controls, and menu items in the console.

You can also use commands on the View menu to expand the selected node in the navigation tree or to expand all nodes simultaneously. For instance, click Expand All from the View menu to see all nodes and related instances at one time.

See Also: Part V, "Logging and Auditing"

2.8 Viewing Configuration Details in the Console

Administrators can view configuration details of individual agents, servers instances, data sources, shared components, and application domains from the OAM Administration Console.

In this example, you will view configuration details for an OAM Agent (WebGate). However, you can use similar steps to view configuration details for server instances, data sources, application domains, or shared components.

Alternatively, you can use custom WLST commands for OAM to view agent and server details.

See Also: Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"

To view configuration details using the Administration Console

1. Go to the Oracle Access Manager Administration console and log in as usual. For example:

   ```
   https://hostname:port/oamconsole
   ```

   In the sample URL, *hostname* refers to computer that hosts the Oracle Access Manager 11g Administration Console; *port* refers to the HTTP port number on which the console host listens; /oamconsole identifies the Administration Console.

2. Click the named tab that provides the configuration details you want to view. For example:

   System Configuration

3. Either select Expand All from the View menu, or expand a node to view its content. For example:

   Expand the Server Instances node, as described in Table 2–11, " Selection Tasks and Controls" on page 2-19.

4. Double-click the instance name in the navigation tree to display the configuration page on the right.

5. View the page and note any specific details of interest.

6. Close the page by clicking the control in the upper-right corner.

2.9 Conducting Searches

This topic describes what you can search for and how to perform a search in the Administration Console.

- About Search Controls
- Searching for an Instance

2.9.1 About Search Controls

Search controls are shown and described in Table 2–12.

Table 2–12 Search Control Definitions

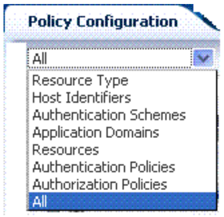
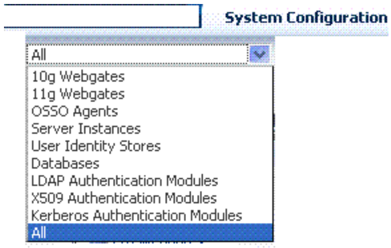
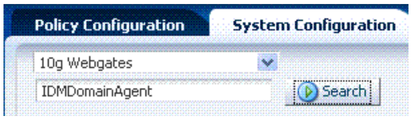
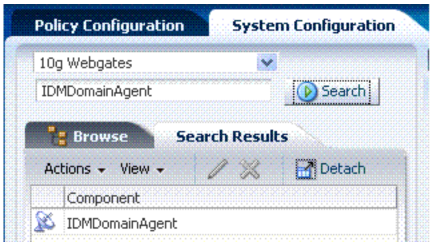
Search Control	Description
 Policy Configuration All Resource Type Host Identifiers Authentication Schemes Application Domains Resources Authentication Policies Authorization Policies All	From the Policy Configuration Search menu, choose an item to define your search.
 System Configuration All 10g Webgates 11g Webgates OSSO Agents Server Instances User Identity Stores Databases LDAP Authentication Modules X509 Authentication Modules Kerberos Authentication Modules All	From the System Configuration Search menu, choose an item to define your search.

Table 2–12 (Cont.) Search Control Definitions

Search Control	Description
 Policy Configuration System Configuration 10g Webgates IDMDomainAgent Search	In the text field, enter the exact name of the instance you want to find. Click the Search button to initiate the search. Note: The name you enter in the field must be an exact match, including capitalization. No wild cards are allowed.
 Policy Configuration System Configuration 10g Webgates IDMDomainAgent Search Browse Search Results Actions View Detach Component IDMDomainAgent	Click the Search Results tab to reveal the results of your search. Click a command button in the tool bar to remove the instance. The configuration page appears on the right side of the console. Click Detach in the tool bar to expand the table to a full page. Select a View menu item to alter the appearance of the results table.

2.9.2 Searching for an Instance

This topic describes how to perform a search using the capabilities in the Administration Console.

In the example in the following procedure, a search is conducted for an application domain. The procedure is generally the same, regardless of the type or instance you might choose.

To perform a search

1. Activate the Policy or System Configuration tab.
2. From the search type list, choose a type to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   my_host_identifier

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit command button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal when the Confirmation window appears.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.
6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

2.10 Using Online Help

At any time while using the Oracle Access Manager Administration Console, you can click the Help link at the top of the page to get more information. Online Help topics link to information in an online version of this book.

Online Help topics link to information in an online version of this book. Online Help procedures provide a brief introduction, followed by the procedure itself.

Generally speaking, topics that are displayed by selecting Help in the Administration Console appear in only English and Japanese languages. Online Help is not translated into the nine ADMIN languages.

You can click the Welcome tab to display a list of topics that describe actions you can take. For specific help topics, use the following procedure.

To locate a specific help topic

1. From the Administration Console, click a tab or named node in the navigation tree.
2. Click Help in the upper-right corner of the Administration Console.
3. Review the page that appears in a new window and select one of the following links to:
   - **More**—Click this link to view more information.
   - **How?**—Click this link to see steps to perform a task related to your help search.
   - **Contents**—In the left Help pane, expand Contents to see all help topics as well as all topics in the online manual.
   - **Search**—Displays a search window where you can enter your help search criteria.
4. Click the following buttons, as needed:
   - **View**—Displays a set of viewing options.
   - **Arrows**—Return to the previous page or go forward to the next page.
   - **Printer Icon**—Prints the page.
   - **Envelope Icon**—Emails the page.

2.11 Command-Line Tools

Several command-line tools are available to perform various tasks using the keyboard rather than the Administration Console. After using these commands, configurations will be available in the Administration Console:

- Remote registration tool, oamreg, enables remote registration of OAM Agents and OSSO Agents (mod_osso), and creation of default application domains.

  See Also: Chapter 6, "Registering Partners (Agents and Applications) Remotely"

- Upgrade Assistant (UA) enables you to transfer OSSO 10g configuration to Oracle Access Manager 11g.

See Also:

- *Oracle Fusion Middleware Upgrade Planning Guide*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management*

- Oracle WebLogic Scripting Tool (WLST) provides a number of custom OAM command-line alternatives for tasks you can perform in the OAM Administration Console.

  See Also: Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"

2.12 Logging Component Events

You can use the logging mechanism to capture critical Oracle Access Manager 11g component events. Logging is the mechanism by which Oracle Access Manager 11g components write messages to a file. These messages can be logged at different levels of granularity.

For more information, see Chapter 13, "Logging Component Event Messages".

Part II

OAM 11g System Management

Part II provides information about managing low-level system configuration details for Oracle Access Manager 11g Release 1 (11.1.1).

Part II contains the following chapters:

- Chapter 3, "Managing Data Sources"
- Chapter 4, "Managing OAM Server Registration"
- Chapter 5, "Registering Partners (Agents and Applications) by Using the Console"
- Chapter 6, "Registering Partners (Agents and Applications) Remotely"

3 Managing Data Sources

This chapter describes the steps to register and administer data sources in the Oracle Access Manager Administration Console. This chapter includes the following topics:

- Prerequisites
- Introduction to Managing Data Sources
- Managing User Identity Store and OAM Administrator Registrations
- Managing the Database by Using the OAM Administration Console

3.1 Prerequisites

This section identifies requirements for tasks in this chapter. Before you begin tasks in this chapter, be sure to review the following topics:

- Introduction to Managing Data Sources
- Managing User Identity Store and OAM Administrator Registrations

3.2 Introduction to Managing Data Sources

Various types of data must be managed for Oracle Access Manager 11g, as described in following topics:

- About User Identity Stores
- About the OAM Policy and Session Data Store
- About the OAM Configuration Data File
- About Security Keys and the Embedded Java Key Store

See Also:

- Chapter 12 for details about session data stored in-memory using Oracle Coherence and propagated to Oracle Database
- Chapter 14 for details about Audit data is stored within audit files or a separate Oracle Database (not the policy store)

3.2.1 About User Identity Stores

OAM administrator and user identities are stored within an LDAP user identity store for use during authentication and authorization.

A User Identity Store is a centralized LDAP store in which an aggregation of administrator and user-oriented data is kept and maintained in an organized way.

Only user and group identity data are stored in the centralized LDAP store. Only the Primary user identity store can be used to authenticate:

- Administrators signing in to use the OAM Administration Console, remote registration, and custom administrative commands for OAM 11g in WLST
- Users attempting to access an OAM-protected resource

In the OAM 11g Administration Console, User Identity Store registrations are organized under the Data Sources node of the System Configuration tab. Administrators can register, view, modify, and delete User Identity Store registrations using either the OAM Administration Console or custom WLST commands for OAM 11g.

During initial WebLogic Server domain configuration using the Oracle Fusion Middleware Configuration Wizard, the embedded LDAP is configured as the one and only user identity store.

Note: The embedded LDAP performs best with fewer than 10,000 users. With more users, consider a separate enterprise LDAP server.

Within the embedded LDAP, the Administrators group is created with "`weblogic`" is seeded as the default administrator.

See Also: *Oracle Fusion Middleware Securing Oracle WebLogic Server* Part Number E13707-01

Administrators can define multiple user identity stores for OAM 11g. Each identity store can rely on a different LDAP provider. External LDAP repositories can provide user, role, and group membership information to be used when evaluating policies during authentication and authorization. After installing and configuring an enterprise LDAP directory server, the administrator must register it with OAM 11g to ensure connectivity with Oracle Access Manager servers.

After registering the identity store, administrators can reference it in one or more authentication modules that form the basis for authentication schemes.

Note: Only the Primary user identity store is used for administrator and user authentication.

For more information, see "About the User Identity Store Registration Page" on page 3-4.

See Also: Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"

3.2.2 About the OAM Policy and Session Data Store

OAM 11g requires a database to store OAM policy data and (optionally) OAM user session data.

The policy database must be installed according to vendor instructions, and extended with the OAM-specific schema using RCU, as described in Oracle Fusion Middleware Installation Guide for Oracle Identity Management. Running the Oracle Access Manager with Database Policy Store configuration template automatically prepares the database to store OAM 11g policy and session data.

The database is specified for Oracle Access Manager 11g during initial configuration in a Oracle WebLogic Server domain using the Oracle Fusion Middleware Configuration Wizard.

Note: Your OAM 11g deployment can have one policy and one session store, at most. By default, a single JDBC data source is used for both.

The following data is maintained:

- Policy data, including authentication modules and schemes, application domains, authentication and authorization policies.
- Session data, as a persistent backup to distributed in-memory storage

  An administrator must extend the database with the OAM-specific schema.

For more information, see "Managing the Database by Using the OAM Administration Console" on page 3-10.

Note: The preferred mode for audit data storage in production environments is writing audit records to a stand-alone RDBMS database for audit data only. This is done using a separately configured audit store. The policy store is not used for audit data.

3.2.3 About the OAM Configuration Data File

Oracle Access Manager provides an XML file (oam-config.xml) containing all OAM-related system configuration data. Each OAM Server has a local copy of the latest configuration XML file.

Any changes that are made to the OAM deployment configuration, including server and agent registration, are stored in the oam-config.xml file and are automatically propagated to each OAM Server.

Note: The oam-config.xml file should not be edited. Changes could result in lost data or overwriting of the file during data synch operations.

Whether you have fail over configured in a high-availability environment, or not, all OAM Servers always have the latest oam-config.xml file.

3.2.4 About Security Keys and the Embedded Java Key Store

The preferred keystore format is JKS (Java KeyStore). A Java keystore is associated with OAM 11g behind the scenes and is used to store cryptographic security keys that are generated to encrypt agent traffic and session tokens.

- Every OAM and OSSO Agent has a secret key that other agents cannot read.
- There is also a key to encrypt Oracle Coherence-based session management traffic.
- During agent and partner (application) registration, a key is generated that is used for encrypting and decrypting SSO Cookies (ObSSOCookie for WebGates and mod_osso cookie).

Table 3–1 compares the cryptographic keys generated by OAM 11g, 10g, and OSSO 10g, as well as a brief description of there each is stored.

Table 3–1 Key Comparison

	OAM 11g	OAM 10g	OSSO 10g
Cryptographic keys	▪ One per agent secret key shared between WebGate and OAM Server ▪ One OAM Server key ▪ 11g WebGate: OAMAuthnCookie ▪ 10g WebGate: ObSSOCookie	One global shared secret key for all OAM WebGates	▪ One key per partner shared between mod_osso and OSSO server ▪ OSSO server's own key ▪ One global key per OSSO setup for the GITO domain cookie
Keys storage	▪ **Agent side**: A per agent key is stored locally in the Oracle Secret Store in a wallet file ▪ **OAM 11g server side:** A per agent key, and server key, are stored in the credential store on the server side	Global shared secret stored in the directory server only (not accessible to WebGate)	▪ **mod_osso side**: partner keys and GITO global key stored locally in obfuscated configuration file ▪ **OSSO server side**: partner keys, GITO global key, and server key are all stored in the directory server

Note: The key store is not available through the console and cannot be viewed, managed, or modified.

3.3 Managing User Identity Store and OAM Administrator Registrations

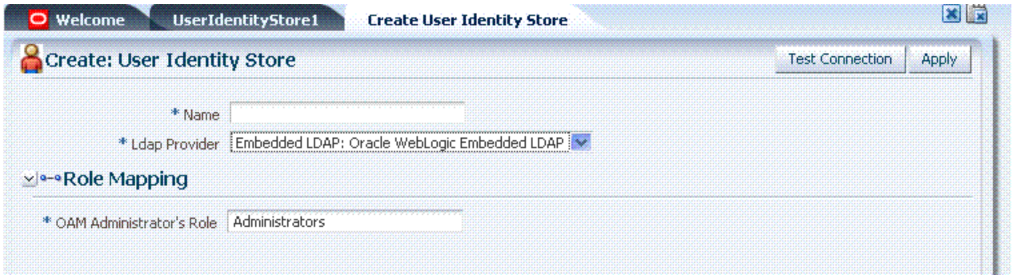
This section provides the steps you need to manage user identity store registrations using the OAM 11g Administration Console.

- About the User Identity Store Registration Page
- Searching for a User Identity Store Registration
- Registering a New User Identity Store
- Viewing or Editing a User Identity Store Registration
- Deleting a User Identity Store Registration
- Defining a New OAM Administrator Role

3.3.1 About the User Identity Store Registration Page

This topic describes the various user identity store settings under the System Configuration tab.

Figure 3–1 Create User Identity Store Page for Embedded LDAP



Create: User Identity Store

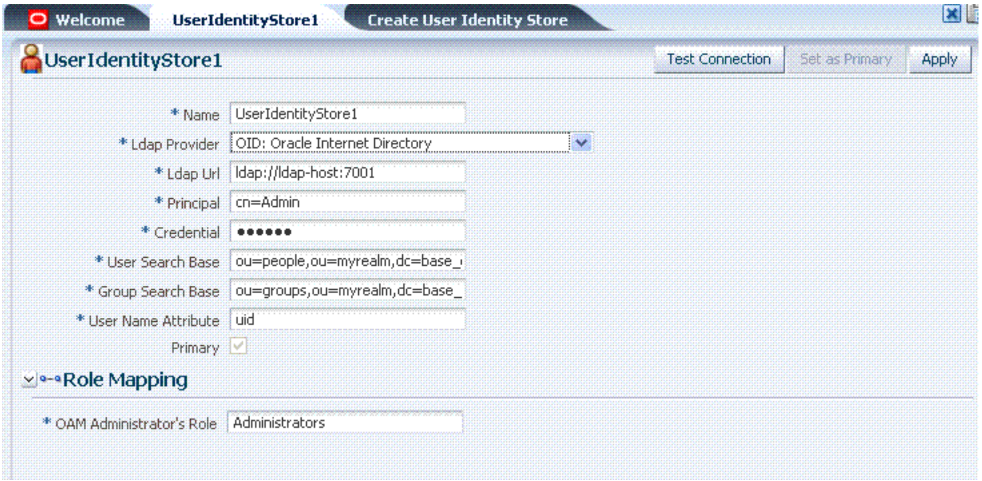
* Name
* Ldap Provider: Embedded LDAP: Oracle WebLogic Embedded LDAP

Role Mapping

* OAM Administrator's Role: Administrators

Figure 3–2 illustrates a completed user identity store registration page in the Administration Console. With the exception of the Embedded LDAP store, all registrations require the same type of information regardless of the LDAP directory server you are registering. A Test Connection button appears at the top of this page. By default, the embedded LDAP store is set as the Primary user identity store. If a data source is not set as primary, the Set As Primary button is available.

Figure 3–2 Completed User Identity Store Registration for Oracle Internet Directory



UserIdentityStore1

* Name: UserIdentityStore1
* Ldap Provider: OID: Oracle Internet Directory
* Ldap Url: ldap://ldap-host:7001
* Principal: cn=Admin
* Credential: ●●●●●●
* User Search Base: ou=people,ou=myrealm,dc=base_
* Group Search Base: ou=groups,ou=myrealm,dc=base_
* User Name Attribute: uid

Primary ☑

Role Mapping

* OAM Administrator's Role: Administrators

Required settings are identified by the asterisk (*). Table 3–2 describes each element and is organized by element types.

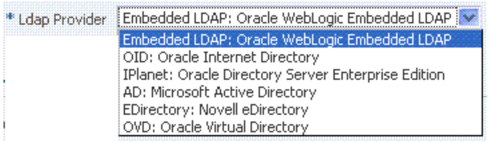
Table 3–2 Required User Identity Store Elements

Elements	Description
Name	A unique name for this registration. Use up to 30 characters for the name.
	Location and Credentials
LDAP URL	The URL for the LDAP host, including the port number. For example, the default embedded LDAP host might be: `ldap://localhost:7001` You can also specify ldaps://, which supports SSL_NO_AUTH.

Table 3–2 (Cont.) Required User Identity Store Elements

Elements	Description
Principal	The user DN for the connection pool over which all other BINDs occur. Oracle recommends a non administrative user with appropriate Read and Search privileges for the user and group base DNs. For example: `uid=amldapuser,ou=people,o=org`
credential	The password of the Principal, which is encrypted for security.
	Connection Details
Connection Pool Size	The initial size set for the connection pool.
Connection Wait Timeout	The number (in seconds) that connection requests can wait before timing out in the event of a fully utilized pool.
Connection Pool Retry Count	The number of time that the connection is retried when there is a connection failure.
Search Time Limit	The time limit for LDAP searches and bind operations on the connection pool. Default: 0
Referral Policy	One of these values: ▪ follow: Follows referrals during an LDAP search ▪ ignore: Ignores referral entries during an LDAP search ▪ throw: Results in a ReferralException, which can be caught by the component user.
	Users
User Name Attribute	The attribute that identifies the username. For example: `uid`
User Search Base	The node in the directory information tree (DIT) under which user data is stored, and the highest possible base for all user data searches. For example: `ou=`*`people`*`,ou=`*`myrealm`*`,dc=`*`base_domain`*
User Filter Object Class	The object classes to be included in search results for users, in a comma-separated list of user object class names. For example: user,person.
	Groups
Group Name Attribute	The attribute that identifies the group name. Default: cn
Group Search Base	Currently only static groups are supported, with the `uniquemember` attribute. The node in the directory information tree (DIT) under which group data is stored, and the highest possible base for all group data searches. For example: `ou=`*`groups`*`,ou=`*`myrealm`*`,dc=`*`base_domain`*
Group Filter Object Class	The object classes to be included in the search results for groups, in a comma-separated list of group object classes. For example: groups,groupOfNames.
	Group Caching
Group Cache Enabled	Boolean value for group cache: true or false. Default: true
Group Cache Size	Integer for the group cache size. Default: 10000
Group Cache TTL	Integer (in seconds) for Time to Live for group cache elements. Default: 0

Table 3–2 (Cont.) Required User Identity Store Elements

Elements	Description
LDAP Provider	A list of all supported LDAP providers from which you can choose.  * Ldap Provider: Embedded LDAP: Oracle WebLogic Embedded LDAP Embedded LDAP: Oracle WebLogic Embedded LDAP OID: Oracle Internet Directory IPlanet: Oracle Directory Server Enterprise Edition AD: Microsoft Active Directory EDirectory: Novell eDirectory OVD: Oracle Virtual Directory
Primary	Primary is checked only when this LDAP store is set as the primary User Identity Store (by clicking the Set as Primary button at the top of the registration page). Checking this option has no effect.
OAM Administrator's Role	The group defined within the primary user identity store that grants users full OAM system and policy configuration privileges. Default Group = Administrators **Note**: Specifying a different LDAP group prohibits WebLogic administrators from logging in to OAM. See Also: "Introduction to OAM Administrators" on page 2-9.

See Also: Details about classifying users in Chapter 9, "Managing Policies to Protect Resources and Enable SSO"

3.3.2 Searching for a User Identity Store Registration

Users with valid OAM Administrator credentials can use this procedure to search for a user identity store using the Administration Console.

Prerequisites

The user identity store that you intend to register must be installed and running with appropriate users and groups and roles defined.

To search for a user identity store registration

1. Activate the System Configuration tab.
2. From the search type list, choose the User Identity Stores type to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   my_user_identity_store

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit command button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.

6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

3.3.3 Registering a New User Identity Store

Users with valid OAM Administrator credentials can use this procedure to register a new user identity store using the Administration Console.

After you register the identity store, you can reference it in one or more authentication modules that form the basis for authentication schemes.

> **See Also:** "About the User Identity Store Registration Page" on page 3-4

Prerequisites

The user identity store that you intend to register must be installed and running.

To register a new user identity store definition

1. From the System Configuration tab, navigation tree, expand the Data Source nodes.
2. Click User Identity Stores.
3. Click the Create command button in the tool bar.
4. Add appropriate values for each element, as described in Table 3–2.
5. Role Mapping: Enter the name of the LDAP group defined for OAM administrators within the user identity store (Table 3–2).
6. Click Apply to submit the registration.
7. Click Test Connection button to confirm connectivity, then close the Confirmation window.
8. Close the registration page, then re-open it.
9. Set as Primary: Click this button to set this LDAP store as the primary user identity store for authentication and authorization.
10. Click Apply to submit the registration and close the page.
11. Configure one or more authentication modules to use this store, as described in "Managing Authentication Modules" on page 8-12.

3.3.4 Viewing or Editing a User Identity Store Registration

Users with valid OAM Administrator credentials can modify the registration of a user identity store.

To view or modify a user identity store registration

1. From the System Configuration tab, navigation tree, expand the Data Sources node.
2. Expand the User Identity Stores node.
3. Double-click the name of the desired User Identity Store registration.
4. Modify values as needed (see Table 3–2).
5. Click Apply to update the registration (or close the page without applying changes).

6. Click Test Connection button to confirm connectivity, then close the Confirmation window.
7. Close the registration page, then re-open it.
8. Set as Primary: Click this button to set this LDAP store as the primary user identity store for authentication and authorization.
9. Close the page when you finish.

3.3.5 Deleting a User Identity Store Registration

Users with valid OAM Administrator credentials can use this procedure to delete a non-primary user identity store registration using the Administration Console.

> **Note:** You cannot delete the primary user identity store registration.

To delete a secondary user identity store registration

1. Edit LDAP Authentication Modules that reference the store to be deleted (to ensure a valid identity store is referenced).
2. From the System Configuration tab, navigation tree, expand the Data Sources node.
3. Expand the User Identity Stores node.
4. Optional: Double-click the desired instance name to confirm it is the one to delete and then close the page.
5. Click the desired instance name and then click the Delete button in the tool bar.
6. Click the Delete button in the Confirmation window (or click Cancel to dismiss the window and retain the instance).
7. Confirm that the definition is no longer listed in the navigation tree.

3.3.6 Defining a New OAM Administrator Role

By default, the OAM Administrators role is the same as the WebLogic Administrators role (Administrators). However, OAM administrators can use the following procedure to define a new OAM administrator role which must be stored in the primary user identity store for OAM 11g.

To change the OAM Administrator role

1. In the Primary user identity store for OAM:
    a. Define a different LDAP group to use for OAM Administrators.
    b. Ensure that your OAM Administrators group is available in the group search base.
2. Log in to the OAM Administration Console and:
    a. Open the desired User Identity Store registration, as described in "Viewing or Editing a User Identity Store Registration".
    b. Modify the OAM Administrator role as needed (see Table 3–2).
    c. Apply the changes to this registration, test the connection, and close the page.
    d. Set as Primary: Re-open the registration page, click the Set as Primary button, and click Apply.

3. Test the New Role: Close the browser window, then re-open it.

   a. Sign out of the OAM Administration Console and close the browser window.

   b. Start up the OAM Administration Console and attempt to log in using the previous OAM Administrator role to confirm that this attempt fails.

   c. Log in using the new OAM Administrator role to confirm that this attempt is successful.

3.4 Managing the Database by Using the OAM Administration Console

This section includes the following topics:

- About Database Deployment for OAM 11g
- Configuring a Separate Database for Session Data

3.4.1 About Database Deployment for OAM 11g

Oracle requires a single database as the policy, which can also be used to store session data. Using the database as the session store provides greater scalability and fault-tolerance (against a power event taking all servers down). You can have up to one policy database and one session database.

During initial deployment using the WebLogic Configuration Wizard, the following database details are requested:

- Database login ID and password
- Database Service name and location

Using the WebLogic Configuration Wizard you can test the connection to the database. Also, the database is registered with OAM.

Basic schema creation occurs when the RCU is invoked. The RCU prepares the database to accept data for OAM 11g. Running the Oracle Access Manager with Database Policy Store configuration template automatically prepares the database to store OAM 11g policy and session data. Actual OAM policy elements are created the first time the WebLogic AdminServer is started with the OAM Administration Console deployed.

See Also: Oracle Fusion Middleware Installation Guide for Oracle Identity Management

Note: Only one database can be registered with OAM for use as a policy store. OAM includes a read-only oam-policy.xml file in the domain home. This file should not be edited directly. Changes can result in lost data or overwriting of the file during data synch operations.

3.4.2 Configuring a Separate Database for Session Data

Oracle Access Manager 11g includes a data source named "oamDS" which is configured against the database instance extended with the OAM Schema. The following pre-defined Java Naming and Directory Interface (JNDI) names are used by the OAM Server to refer the data source.

```
jdbc/oamds (used by both the policy layer and session layer to access database)
```

You can use the following procedure to create a separate database instance for session data using the WebLogic Administration Console. There is no support for this action in the OAM Administration Console.

To create and use an independent database for session data

1. Install and configure the database for session data and then use RCU with the OAM-specific schema to set up the database as a session data store.
2. Create a new Data Source instance for session data:
   a. From the WebLogic Administration Console, Domain Structure panel, expand the domain name, Services node.
   b. Expand JDBC, Data Source.
   c. Create a new Data Source with the JNDI name `jdbc/oamsession`.
   d. Save the changes.
   e. Stop the OAM Servers and the AdminServer to avoid potential loss of data during the next step.
   f. In oam-config.xml, edit the value of the DataSourceName attribute to the one configured in step 1. For example:

      ```
      domain-home/config/fmwconfig/oam-config.xml
      ```

      From:

      ```
      <Setting Name="SmeDb" Type="htf:map">
        <Setting Name="URL" Type="xsd:string">jdbc:oracle:thin://amdb.example.
          com:2001/AM</Setting>
        <Setting Name="Principal" Type="xsd:string">amuser</Setting>
        <Setting Name="Password" Type="xsd:string">password</Setting>
        <Setting Name="DataSourceName" Type="xsd:string">jdbc/oamds</Setting>
      </Setting>
      ```

      To:

      ```
      <Setting Name="SmeDb" Type="htf:map">
        <Setting Name="URL" Type="xsd:string">jdbc:oracle:thin://amdb.example.
          com:2001/AM</Setting>
        <Setting Name="Principal" Type="xsd:string">amuser</Setting>
        <Setting Name="Password" Type="xsd:string">password</Setting>
        <Setting Name="DataSourceName"
      Type="xsd:string">jdbc/oamsession</Setting>
      </Setting>
      ```

3. Restart AdminServer and OAM Servers.

4 Managing OAM Server Registration

This chapter describes how to provision and manage OAM Server instance registrations using the Oracle Access Manager 11g Administration Console. The following topics are included:

- Prerequisites
- Introduction to OAM Server Registration and Management
- Managing Individual OAM Server Registrations
- Introduction to Managing OAM Server Common Properties
- Managing Common OAM Proxy Simple and Cert Mode Security
- Managing Run Time Policy Evaluation Caches

4.1 Prerequisites

Ensure that the following environmental considerations are met:

- A new Managed Server has been added to the domain using either the Oracle WebLogic Server Administration Console or WLST commands.
- The Oracle JRF Template was applied to the Managed Server (or cluster) if needed. For details, see *Oracle Fusion Middleware Administrator's Guide*.

Oracle recommends that you review the "Introduction to OAM Server Registration and Management".

4.2 Introduction to OAM Server Registration and Management

This section introduces Oracle Access Manager server instance registration and management in the following topics:

- About Server Side Differences Between OAM 11g and OAM 10g
- About Individual OAM Server Registrations
- About the Embedded Proxy Server and Backward Compatibility
- About OAM 11g SSO and Legacy OAM 10g SSO in Combination with OSSO
- About Communication Between OAM Servers and WebGates
- About Server Common Properties

4.2.1 About Server Side Differences Between OAM 11g and OAM 10g

Table 4–1 summarizes server-side differences between Oracle Access Manager 11g, OAM 10g, and OracleAS SSO 10g (extracted from the overall comparison in Table 2–1).

Table 4–1 Summary: Server-side Differences with OAM 11g versus OAM 1g versus OSSO 10g

	OAM 11g	OAM 10g	OSSO 10g
Server-side components	■ OAM Server (installed on a WebLogic Managed Sever) ■ OAM Administration Console (installed on WebLogic Administration Server)	■ Access Server ■ Policy Manager	■ OracleAS SSO server (OSSO server)
Cryptographic keys The protocols used to secure information exchange on the Internet.	■ One per agent secret key shared between WebGate and OAM Server, generated during Agent registration ■ One OAM Server key, generated during Server registration	One global shared secret key per WebGate	■ One key per partner shared between mod_osso and OSSO server ■ OSSO server's own key ■ One global key per OSSO setup for the GITO domain cookie
Keys storage	■ **Agent side**: A per agent key is stored locally in the Oracle Secret Store in a wallet file ■ **OAM 11g server side:** A per agent key, and server key, are stored in the credential store on the server side	Global shared secret stored in the directory server only (not accessible to WebGate)	■ **mod_osso side**: partner keys and GITO global key stored locally in obfuscated configuration file ■ **OSSO server side**: partner keys, GITO global key, and server key are all stored in the directory server

4.2.2 About Individual OAM Server Registrations

Administrators can add one or more Managed Servers to the WebLogic Server domain for use with Oracle Access Manager 11g. When using the WebLogic Configuration Wizard, the OAM Server is automatically registered with OAM 11g. However, if the configuration wizard was not used, the OAM Server must be registered with the OAM 11g to open a communication channel.

Alternatively. You can use custom WLST commands for OAM to display, edit, or delete a server registration Any changes are automatically propagated to the OAM Administration Console and to every OAM Server in the cluster.

> **See Also:** Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"

Only OAM Servers are registered with OAM 11g. The OAM Administration Console on the WebLogic Administration Server is not registered with itself.

Regardless of the method used to register an OAM Server, the details (also known as a registration) are organized under the System Configuration tab in the OAM Administration Console. OAM Server registration details within the OAM Administration Console include:

- Server name, Host, Port
- Proxy: Performs as the legacy Access Server and defines the communication security mode. For more information, see:

  - About the Embedded Proxy Server and Backward Compatibility
  - About OAM 11g SSO and Legacy OAM 10g SSO in Combination with OSSO
  - About Communication Between OAM Servers and WebGates
- Oracle Coherence: Provides a distributed cache for various OAM services, including session data.

Administrators can search for a specific instance registration, register a newly installed OAM Server, view, modify, or delete server registrations using the Oracle Access Manager Administration Console. For more information, see "About the OAM Server Registration Page" on page 4-5.

4.2.3 About the Embedded Proxy Server and Backward Compatibility

Oracle Access Manager 11g server-side components maintain backward compatibility with existing Oracle Access Manager 10g policy-enforcement agents (OAM 10G WebGates and AccessGates) and OracleAS SSO 10g mod_osso (known as OSSO Agents in 11g).

Legacy OAM 10g SSO: The OAM Proxy can accept requests from multiple Access clients concurrently and enables all WebGates and AccessGates to interact with Oracle Access Manager 11g services. For more information, see "OAM Proxy Page" on page 4-6.

> **See Also:** "About OAM 11g SSO and Legacy OAM 10g SSO in Combination with OSSO"

Legacy OracleAS 10g (OSSO): The integrated OSSO proxy handles token generation and validation in response to token requests during authentication using OSSO Agents with OAM 11g. The OSSO proxy needs no configuration. Simply register the OSSO agent with OAM 11g as described in Chapter 5 and Chapter 6.

4.2.4 About OAM 11g SSO and Legacy OAM 10g SSO in Combination with OSSO

You can upgrade OracleAS SSO to use OAM 11g SSO when you have a legacy deployment where OAM 10g is integrated and used in combination with OracleAS (OSSO) 10g.

After upgrading OSSO to use OAM 11g, you can have OAM 10g WebGates operating with OAM 11g SSO the same deployment. In this situation, the OAM Proxy forwards requests to either the OAM 10g Access Server or to OAM 11g services as needed.

The OAM 10g ObSSOCookie is an encrypted session-based single sign-on cookie that is generated when a user authenticates successfully. The OAM 10g ObSSOCookie stores user identity information, which you can cache if needed.

The integrated OAM Proxy supports the AES encryption algorithm of the 10g ObSSOCookie to enable backward compatibility with release 10g WebGates. The 10g Access Server can decrypt the cookie created by the OAM 11g Proxy (and vice versa). This allows OAM 11g to perform authentication and OAM 10g to perform authorization (and vice versa).

> **Note:** An OAM 11g ObSSOCookie created by OAM Proxy is compatible with the ObSSOCookie created by an Oracle Access Manager 10g Access Server.

For more information, see "OAM Proxy Page" on page 4-6.

4.2.5 About Communication Between OAM Servers and WebGates

Communication modes for the OAP channel include:

- Open: Use this unencrypted mode if communication security is not an issue in your deployment.
- Simple: Use this Oracle-signed certificate mode if you have some security concerns, such as not wanting to transmit passwords as plain text, but you do not manage your own Certificate Authority (CA).
- Cert: Use if you want different certificates on OAM Servers and WebGates and you have access to a trusted third-party CA.

On each individual OAM Server registration, the security mode is defined on the Proxy tab, as described in "About the OAM Server Registration Page" on page 4-5.

Simple and Cert modes also require:

- Security passwords that are common to all OAM Servers and WebGates, as described in "Managing Common OAM Proxy Simple and Cert Mode Security" on page 4-13.
- Appropriately signed X.509 digital certificates, as described in Appendix E, "Securing Communication with OAM 11g".

At least one OAM Server instance must be running in the same mode as the agent during agent registration. Otherwise, agent registration fails. After agent registration, however, you can change the communication mode of the OAM Server. Communication between the agent and server would continue to work as long as the WebGate mode is at least at the same level as the OAM Server mode or higher. The agent mode can be higher but cannot be lower. For example, of OAM Server mode is Open, agents can communicate in any of the three modes. If OAM Server mode is Simple, agents can use Simple or Cert mode. If OAM Server mode is Cert, agents must use Cert mode.

See Also: Appendix E, "Securing Communication with OAM 11g"

4.2.6 About Server Common Properties

Using the OAM 11g Administration Console, you can view and modify global settings that all OAM Servers share in common, including those for the SSO Engine, Policy Store, Session Management, Auditing, and Oracle Coherence.

For more information, see "Introduction to Managing OAM Server Common Properties" on page 4-11.

4.3 Managing Individual OAM Server Registrations

This section describes how to register and manage OAM Server instances using the Oracle Access Manager Administration Console. Topics here include:

- About the OAM Server Registration Page
- Registering a Fresh OAM Server Instance
- Viewing or Editing Individual OAM Server and Proxy Settings
- Viewing or Editing Individual OAM Server and Proxy Settings

- Deleting an Individual Server Registration

> **See Also:** "Introduction to Managing OAM Server Common Properties" on page 4-11

4.3.1 About the OAM Server Registration Page

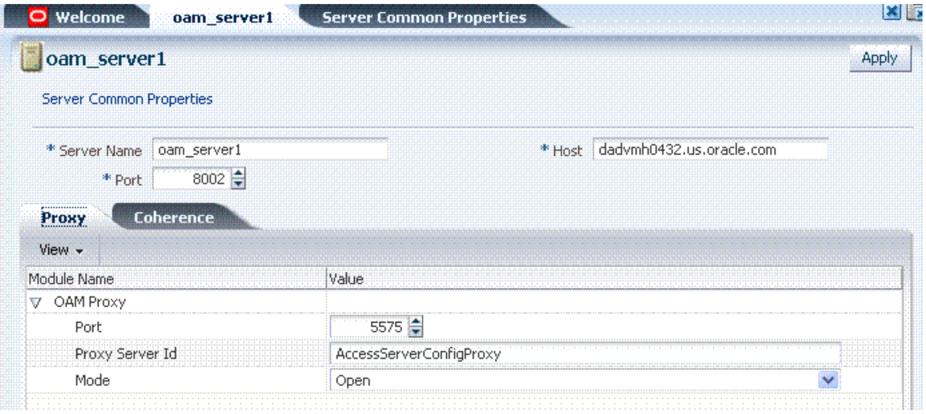
Users with valid OAM Administrator credentials can register a freshly installed Managed Server (OAM Server instance) or modify an existing OAM Server registration using the Oracle Access Manager Administration Console.

Alternatively: You can use custom WLST commands for OAM to register and manage OAM Server instances. Changes are reflected in the OAM Administration Console and are automatically propagated to every OAM Server in the cluster.

> **See Also:** Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"

Figure 4–1 illustrates a typical OAM Server registration page when viewed within the Oracle Access Manager Administration Console.

Figure 4–1 OAM Server Registration Page with Proxy Tab



oam_server1

Server Common Properties

* Server Name: oam_server1
* Host: dadvmh0432.us.oracle.com
* Port: 8002

Module Name	Value
OAM Proxy	
Port	5575
Proxy Server Id	AccessServerConfigProxy
Mode	Open

Individual server registration settings are described in Table 4–2.

Table 4–2 OAM Server Instance Settings

Element	Definition
Server Common Properties	Links to the OAM Server Common Properties page. See Also: "Introduction to Managing OAM Server Common Properties" on page 4-11.
Server name	The identifying name for this server instance, which was defined during initial deployment in the WebLogic Server domain.
Host	The full DNS name (or IP address) of the computer hosting the server instance. For example: *host2.company.com*.
Port	The port on which this server communicates (listens and responds). Default: 5575 Note: If both the SSL and Open ports of the Managed Server are enabled, then the Managed Server is set to the SSL port by default.If you must use the non-SSL port, the credential collector URL the authentication scheme must be set to the absolute URL which points to 'http' as the protocol and non-SSL port. See Also: Appendix E, "Securing Communication with OAM 11g"
Proxy	See "OAM Proxy Page" on page 4-6

Table 4–2 (Cont.) OAM Server Instance Settings

Element	Definition
Coherence	See "Coherence Page for Individual Servers" on page 4-7

See Also: "Managing Individual OAM Server Registrations" on page 4-4

4.3.1.1 OAM Proxy Page

An integrated proxy server (OAM Proxy) is installed with each Managed Server for Oracle Access Manager (OAM Server). The OAM Proxy is used as a legacy Access Server to provide backward compatibility for OAM 10g Agents that are registered with OAM 11g. The Agent can be freshly installed or currently operating within an OAM 10g SSO deployment.

Each OAM Proxy instance requires a different port. The proxy starts listening when the application starts. Registered access clients can immediately communicate with the proxy.

The OAM Proxy handles both configuration and run-time events. Each OAM Proxy can accept requests from multiple access clients concurrently. Each OAM Proxy enables OAM access clients to interact with Oracle Access Manager 11g services. This includes:

- 10*g* (10.1.4.3) WebGates
- 10*g* (10.1.4.2.0) WebGates
- 10*g* (10.1.4.0.1) WebGates
- 11g WebGates (needs no proxy)

Note: For AccessGates, OAM 11g provides authentication and authorization functionality only. Policy modification through AccessGates is not supported.

OAM Proxy settings consist of the details in Table 4–3.

Table 4–3 OAM Proxy Settings for an Individual OAM Server

OAM Proxy Setting	Type	Value
WebLogic Port	int (integer)	The port on which the Oracle WebLogic Server is listening, which is used by the proxy to redirect the user for credential collection.
Port	int (integer)	The unique port on which this OAM Proxy instance is listening.
Proxy Server ID		The identifier of the computer on which the OAM Proxy (and this OAM Server instance) resides. DNS hostname is preferred; however, you can use any valid and relevant string.

Table 4–3 (Cont.) OAM Proxy Settings for an Individual OAM Server

OAM Proxy Setting	Type	Value
Mode		OAM channel transport security for the OAM Proxy can be one of the following (the agent mode must match during registration and can be higher after registration): ■ Open: No encryption. ■ Simple: The data passed between the OAM Agent and OAM Server is encrypted using OAM self-signed certificates. Before specifying Simple mode, you must specify the global passphrase. ■ Cert: The data between the OAM Agent and OAM Server is encrypted using Certificate Authority (CA) signed X.509 certificates. **Note**: Before specifying Cert mode, you must acquire signed certificates from a trusted 3rd party Certificate Authority. **Note**: Simple and Cert transport security modes are governed by information defined on the OAM Server Common Properties OAM Proxy tab, as described in "Managing Common OAM Proxy Simple and Cert Mode Security" on page 4-13. **See Also**: Appendix E if you are configuring Simple or Cert transport security modes.

OAM Proxy Logging: Oracle Access Manager 11g components use the same logging infrastructure as any other Oracle Fusion Middleware 11g component, as described in Chapter 14. However, OAM Proxy uses Apache log4j for logging.

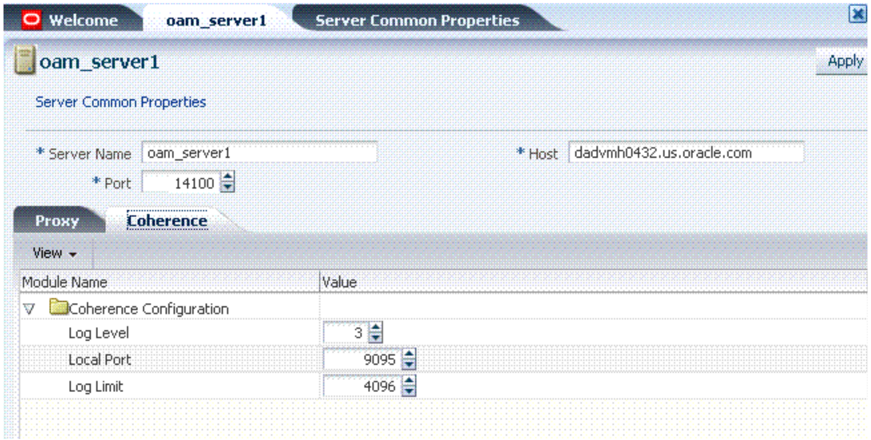
4.3.1.2 Coherence Page for Individual Servers

Coherence provides replicated and distributed (partitioned) data management and caching services on top of a reliable, highly scalable peer-to-peer clustering protocol. Coherence has no single points of failure; it automatically and transparently fails over and redistributes its clustered data management services when a server becomes inoperative or is disconnected from the network.

When a new server is added, or when a failed server is restarted, it automatically joins the cluster and Coherence fails back services to it, transparently redistributing the cluster load. Coherence includes network-level fault tolerance features and transparent soft re-start capability to enable servers to self-heal.

Coherence modules consist of the values, and types for the individual server instance, as shown in Figure 4–2.

Figure 4–2 Coherence Page and Values for an Individual OAM Server



* Server Name: oam_server1

* Host: dadvmh0432.us.oracle.com

* Port: 14100

Module Name	Value
Coherence Configuration	
Log Level	3
Local Port	9095
Log Limit	4096

WARNING: Oracle recommends that you do not modify Oracle Coherence settings for an individual server unless you are requested to do so by an Oracle Support Representative.

Table 4–4 Default Coherence Settings for Individual OAM Servers

Coherence Module	Type of Entry	Description and Default Values
LogLevel	String	The Coherence log level (from 0 to 9) for OAM Server events.
LogPort	int (integer)	The listening port for Coherence logging on the WebLogic Server.
LogLimit	String	The Coherence log limit

Coherence Logging: Appears only in the WebLogic Server log. There is no bridge from Oracle Coherence logging to Oracle Access Manager logging. For Oracle Fusion Middleware 11g logging infrastructure details, see Chapter 14.

4.3.2 Searching for an Individual OAM Server Registration

Users with valid OAM Administrator credentials can perform the following procedure to search for an OAM Server registration using the Administration Console.

Prerequisites

The OAM Server must be registered in the Oracle Access Manager Administration Console.

> **See Also:** "Registering a Fresh OAM Server Instance" on page 4-9

To locate an individual server instance registration

1. Activate the System Configuration tab.
2. From the search type list, choose the Server Instances type to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   my_OAM_Server

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit command button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.
6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

4.3.3 Registering a Fresh OAM Server Instance

Users with valid OAM Administrator credentials can perform the following task to register a new Managed Server (OAM Server) instance using the OAM Administration Console.

> **Note:** Each OAM Server must be registered to communicate with agents.

Prerequisites

The new Managed Server instance must be configured in the Oracle WebLogic Server domain, but not yet started.

See Also:

- Oracle Fusion Middleware Installation Guide for Oracle Identity Management
- "About the OAM Server Registration Page" on page 4-5

To register an OAM Server instance

1. Install the new Managed Server instance and configure it in the Oracle WebLogic Server domain, but do not start this instance.
2. Log in to the OAM Administration Console as usual.
3. From the Welcome page, Server Configuration panel, click the Add Server Configuration link to open a fresh page.

   Alternatively: From the System Configuration tab, click Server Instances then click the Create button in the tool bar to open a fresh page.
4. On the Create: OAM Server page, enter details for your instance, as described in Table 4–2:
   - Server name
   - Host
   - Port
5. Proxy: Enter or select details for this OAM Proxy instance, as described in Table 4–3:
   - WebLogic Port:

- Port
- Proxy Server ID
- Mode (Open, Simple, or Cert)

  See Also: Appendix E if you are using Simple or Cert mode

6. Coherence: Oracle recommends that you do not modify Oracle Coherence settings for an individual server instance unless you are requested to do so by an Oracle Support Representative.

   See Also: "Using Oracle Coherence for Troubleshooting" on page H-16

7. Click Apply to submit the configuration, which should appear in the navigation tree (or close the page without applying changes).
8. Start the newly registered server.

4.3.4 Viewing or Editing Individual OAM Server and Proxy Settings

Users with valid OAM Administrator credentials can perform the following task to view or modify settings for an individual server instance using the Administration Console. For instance, you might decide to change a listening port or the Proxy communication transport security mode.

Changes are immediately visible in the OAM Administration Console and propagated to all OAM Servers in the cluster.

See Also:

- "About the OAM Server Registration Page" on page 4-5
- Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"

To view or modify a server instance registration

1. From the System Configuration tab, navigation tree, click to expand the Server Instances node.
2. Double-click the desired instance name to display its configuration, and then proceed as follows:
   - View Only: Close the page when you finish viewing details.
   - Modify: Perform remaining steps to edit the configuration.
3. On the Create: OAM Server page, enter details for your instance, as described in Table 4–2.
4. **Proxy**: Enter or select details for this OAM Proxy instance, as described in Table 4–3.

   See Also: Appendix E if you are using Simple or Cert mode

5. **Coherence**: Oracle recommends that you do not modify Oracle Coherence settings for an individual server instance unless you are requested to do so by an Oracle Support Representative.

> **See Also:** "Using Oracle Coherence for Troubleshooting" on page H-16

6. Click Apply to submit the changes (or close the page without applying change).

4.3.5 Deleting an Individual Server Registration

Users with valid OAM Administrator credentials can perform the following task to delete a server registration, which disables the OAM server.

To delete a server registration

1. From the System Configuration tab, navigation tree, click to expand the Server Instances node.
2. Double-click the desired instance name to confirm details, then close the page.
3. Click the desired instance name, click the Delete button in the tool bar, and confirm removal in the Confirmation window.
4. Confirm that the instance is removed from the navigation tree.
5. Finalize server instance removal by removing the instance from the WebLogic Server Administration Console.

   The Node Manager on Managed Server host handles the rest automatically.

4.4 Introduction to Managing OAM Server Common Properties

OAM Server Common Properties apply to all OAM Server instances. This section provides the following topics about common server and common single sign-on (SSO) Engine settings:

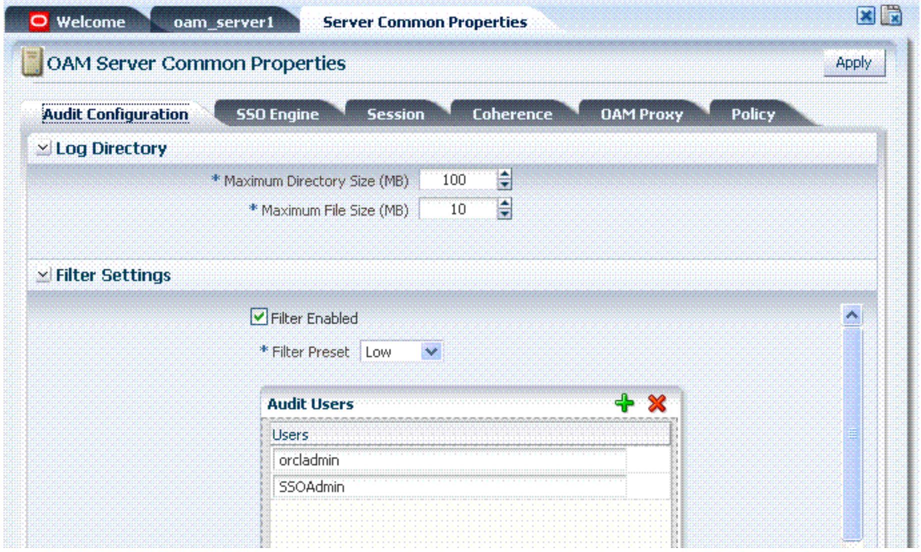
- About OAM Server Common Properties Pages
- Displaying OAM Server Common Properties Pages

4.4.1 About OAM Server Common Properties Pages

OAM Server Common Properties apply to all OAM Server instances. A number of common properties tabs appear on the OAM Server Common Properties page:

Figure 4–3 shows the OAM Server Common Properties page and named tabs.

Figure 4–3 OAM Server Common Properties Page



* Maximum Directory Size (MB) 100

* Maximum File Size (MB) 10

☑ Filter Enabled

* Filter Preset Low

Users
orcladmin
SSOAdmin

The Audit Configuration tab is open when you display OAM Server Common Properties. Common tabs and functionality are described in Table 4–5. Administrators can control and specify certain auditing parameters from this tab. Oracle Access Manager auditing configuration is recorded in the file `oam-config.xml`.

Table 4–5 OAM Server Common Properties Tabs

Tab Name	Description
Audit Configuration	Oracle Access Manager supports auditing for a large number of administrative and run-time events, uniform logging and exception handling, and the diagnostics of all audit events. For details about configuring auditing, see Chapter 14, "Auditing OAM Administrative and Run-time Events".
SSO Engine	Single sign-on enables users, and groups of users, to access multiple applications after a single sign-on and successful authentication. SSO eliminates multiple log ins. For more information, see "Managing the Common SSO Engine" on page 9-45.
Session	Session management refers to the process of managing the lifecycle requirements of a user session, and notification of session events to enable global logout. Global logout is required for OSSO Agents (mod_osso) to ensure that logging out of a session on any entity propagates the logout to all entities. For more information, see Chapter 12, "Managing Sessions".
Coherence	Common Oracle Coherence settings shared by all OAM Servers differ from those for individual OAM Servers. However, in both cases Oracle recommends that you make no adjustments to these settings unless instructed to do so by an Oracle Support Representative. See Also: "Using Oracle Coherence for Troubleshooting" on page H-16.
OAM Proxy	During initial deployment, the Simple mode global passphrase or the Cert (certificate) mode keystore alias and password are defined. For more information, see "Managing Common OAM Proxy Simple and Cert Mode Security" on page 4-13.
Policy	During runtime policy evaluation, the Resource Matching Cache maps the requested URL to the policy. See Also: "Managing Run Time Policy Evaluation Caches" on page 4-16

See Also: Details for other operations common to all OAM components:

- Chapter 13, "Logging Component Event Messages"
- Chapter 15, "Monitoring OAM Metrics by Using Oracle Access Manager"

4.4.2 Displaying OAM Server Common Properties Pages

Users with valid OAM Administrator credentials can perform the following task to display OAM Server Common Properties pages.

To display OAM Server Common Properties pages

1. From the System Configuration tab, navigation tree, double-click Server Instances to display the OAM Server Common Properties page and tabs.

   Alternatively: Click the Server Common Properties link at the top of an individual server registration page.

2. Click the named tab that identifies the configuration details you want and refer to additional information as described in Table 4–5.

4.5 Managing Common OAM Proxy Simple and Cert Mode Security

This section provides the following details:

- About Simple and Cert Mode Transport Security
- About the Common OAM Proxy Page for Secure Server Communications
- Viewing or Editing Simple or Cert Settings for OAM Proxy

4.5.1 About Simple and Cert Mode Transport Security

Table 4–6 outlines the similarities between Simple and Cert modes.

See Also: Appendix E, "Securing Communication with OAM 11g"

Table 4–6 Summary: Simple and Cert Mode

Artifact or Process	Simple Mode	Cert Mode	Open Mode
X.509 digital certificates only.	X	X	N/A
Communication between OAM Agents and OAM Servers is encrypted using Transport Layer Security, RFC 2246 (TLS v1).	X	X	N/A
For each public key there is a corresponding private key that Oracle Access Manager stores in a file:	aaa_key.pem generated by openSSL	aaa_key.pem generated by your CA	N/A
Signed certificates in Privacy Enhanced Mail (PEM) format	aaa_cert.pem generated by openSSL	aaa_cert.pem generated by your CA	N/A
During OAM Server configuration, secure the private key with a Global passphrase or PEM format details, depending on which mode you are using. Before an OAM Server or WebGate can use a private key, it must have the correct passphrase.	Global passphrase stored in a nominally encrypted file: ▪ password.xml	PEM format: ▪ KeyStore Alias ▪ KeyStore Alias Password	N/A

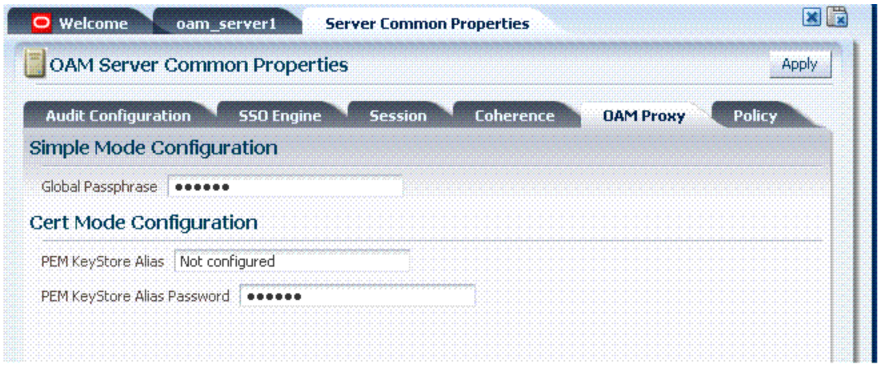
Table 4–6 (Cont.) Summary: Simple and Cert Mode

Artifact or Process	Simple Mode	Cert Mode	Open Mode
During OAM Agent or OAM Server registration, the communication mode is propagated to the OAM 11g Administration Console.	Same passphrase for each WebGate and OAM Server instance.	Different passphrase for each WebGate and OAM Server instance.	N/A
The certificate request for the WebGate generates the certificate request file, which you must send to a root CA that is trusted by the OAM Sever. The root CA returns the WebGate certificates, which can then be installed either during or after WebGate installation.	cacert.pem The certificate request, signed by the Oracle-provided openSSL Certificate Authority	aaa_req.pem The certificate request, signed by the your Certificate Authority	N/A
Encrypt the private key using the DES Algorithm. For example: `openssl rsa -in aaa_key.pem -passin pass: -out aaa_key.pem -passout pass: passphrase -des`	N/A	X	N/A
Agent Key Password	N/A	Enter a password during agent registration (see Table 5–5).	N/A
During Agent registration, ObAccessClient.xml is generated in: $DOMAIN_HOME/output/$*Agent_Name*/	ObAccessClient.xml **Copy to**: **11g WebGate**: *11gWebGate_instance_dir*/ **10g WebGate**: $*WebGate_install_dir*/	ObAccessClient.xml **Copy to**: **11g WebGate**: *11gWebGate_instance_dir*/ **10g WebGate**: $*WebGate_install_dir*/	ObAccessClient.xml **Copy to**: **11g WebGate**: *11gWebGate_instance_dir*/ **10g WebGate**: $*WebGate_install_dir*/
During Agent registration, password.xml is generated in: $DOMAIN_HOME/output/$*Agent_Name*/ **See Also**: Appendix E	password.xml **Copy to**: **11g WebGate**: *11gWebGate_instance_dir*/ **10g WebGate**: $*WebGate_install_dir*/	password.xml **Copy to**: **11g WebGate**: *11gWebGate_instance_dir*/ **10g WebGate**: $*WebGate_install_dir*/	N/A
During Agent registration, aaa_key.pem is generated in: $DOMAIN_HOME/output/$*Agent_Name*/ **See Also**: Appendix E	aaa_key.pem **Copy to**: **11g WebGate**: *11gWebGate_instance_dir*/ **10g WebGate**: $*WebGate_install_dir*/	aaa_key.pem **Copy to**: **11g WebGate**: *11gWebGate_instance_dir*/ **10g WebGate**: $*WebGate_install_dir*/	N/A

4.5.2 About the Common OAM Proxy Page for Secure Server Communications

The OAM Proxy tab on the OAM Server Common Properties page is where administrators can configure Simple or Cert communication security for use by all server instances and WebGates. Figure 4–4 illustrates the settings on this tab.

Figure 4–4 Server Common OAM Proxy Page



Simple Mode Configuration

Global Passphrase ••••••

Cert Mode Configuration

PEM KeyStore Alias Not configured

PEM KeyStore Alias Password ••••••

Table 4–7 describes the settings required for Simple or Cert mode configurations.

Table 4–7 Server Common OAM Proxy Secure Communication Settings

Mode	Description
Simple Mode Configuration	The global passphrase for Simple mode communication using OAM-signed X.509 certificates. This is set during initial OAM Server installation. Administrators can edit this passphrase and then reconfigure all existing OAM Agents to use it, as described in"Viewing or Editing Simple or Cert Settings for OAM Proxy".
Cert Mode Configuration	Details required for the KeyStore where the Cert mode X.509 certificates signed by an outside Certificate Authority reside: ■ Keystore Alias ■ Keystore Alias Password **Note**: These are set during initial OAM Server installation. The certificates can be imported using the import certificate utility or the keytool shipped with JDK. Administrators can edit the alias and password and then reconfigure all existing OAM Agents to use it, as described in"Viewing or Editing Simple or Cert Settings for OAM Proxy".

4.5.3 Viewing or Editing Simple or Cert Settings for OAM Proxy

OAM Administrators can use this procedure to confirm or alter settings for the common OAM Proxy.

See Also:

- "Registering and Managing WebGate Agents Using the Administration Console" on page 5-9
- Appendix E, "Securing Communication with OAM 11g"

To view or edit Simple or Cert mode settings for the OAM Proxy

1. From the System Configuration tab, navigation tree, double-click Server Instances to display the Server Common Properties page.

   Alternatively: From the server instance registration page, click the Server Common Properties link.

2. Click the OAM Proxy tab.
3. **Simple Mode Configuration**: Specify the Global Passphrase.
4. **Cert Mode Configuration**: Specify the following details.

- PEM Keystore Alias
- PEM Keystore Alias Password

5. Click Apply to submit the changes and dismiss the Confirmation window (or close the page without applying changes).

4.6 Managing Run Time Policy Evaluation Caches

This section explains:

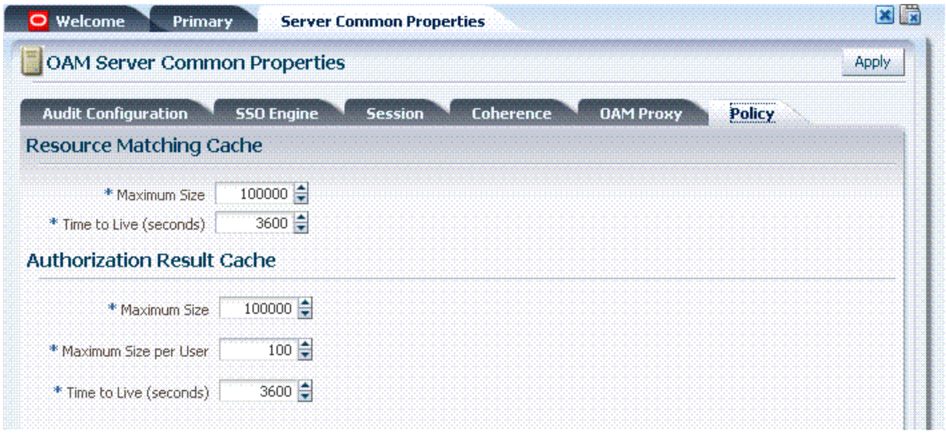
- About Common Run Time Policy Evaluation Caches
- Managing Common Runtime Policy Evaluation Caches

See Also: "About Run Time Resource Evaluation" on page 9-16

4.6.1 About Common Run Time Policy Evaluation Caches

Figure 4–5 illustrates the OAM Server Common Properties Policy tab. This tab provides settings for the Resource Matching Cache and the Authorization Result Cache, which come into play during policy evaluation at run time.

Figure 4–5 Common Policy Evaluation Caches



Resource Matching Cache

* Maximum Size 100000
* Time to Live (seconds) 3600

Authorization Result Cache

* Maximum Size 100000
* Maximum Size per User 100
* Time to Live (seconds) 3600

Table 4–8 outlines these global settings that apply to all servers and requests.

Table 4–8 Policy Evaluation Caches

Element	Description
Resource Matching Cache	Caches mappings between the requested URL and the policy holding the resource pattern that applies to the URL. Default Values: ■ Maximum Size 100000 Zero disables the cache ■ Time to Live (seconds) 3600 Zero disables Time to Live
Authorization Result Cache	Caches policy decisions for the requested URL and user. Default Values: ■ Maximum Size 100000 Zero disables the cache ■ Maximum Size per User 100 Zero disables the cache ■ Time to Live (seconds) 3600 Zero disables Time to Live

4.6.2 Managing Common Runtime Policy Evaluation Caches

OAM Administrators can use this procedure to manage the OAM Server common runtime policy evaluation caches.

> **See Also:** "Changing the Request Cache Type in a High Availability Environment" on page F-5

To manage common runtime policy evaluation cache settings

1. From the System Configuration tab, navigation tree, double-click Server Instances.

   Alternatively: From a server instance registration page, click the Server Common Properties link.

2. On the Server Common Properties page, click the Policy tab.
3. **Resource Matching Cache**: Specify details and click apply (Table 4–8).
4. **Authorization Result Cache**: Specify details and click apply (Table 4–8).
5. Click Apply to submit the changes and dismiss the Confirmation window (or close the page without applying changes).

5

Registering Partners (Agents and Applications) by Using the Console

Only a registered policy enforcement agent can communicate with OAM 11g authentication and authorization services. OAM administrators must register the agent that resides on the computer hosting the partner application to be protected. A partner application is one that delegates the authentication function to the SSO provider (Oracle Access Manager 11g) to spare users from re-authenticating when accessing multiple resources.

This chapter focuses on using the OAM Administration Console to perform agent registration and management. This chapter includes the following topics:

- Prerequisites
- Introduction to Policy Enforcement Agents
- Registering and Managing WebGate Agents Using the Administration Console
- Registering and Managing OSSO Agents Using the Administration Console

Note: To use the command-line to register a partner, see Chapter 6.

5.1 Prerequisites

Before you can perform tasks in this chapter ensure that the OAM Administration Console and a managed OAM Server are running.

Following are the knowledge-based requirements for tasks in this chapter.

- Review Introduction to Policy Enforcement Agents
- Review Chapter 17, "Managing OAM 10g WebGates with OAM 11g" if you are registering OAM 10g WebGates

5.2 Introduction to Policy Enforcement Agents

This section provides information about access clients, known as policy-enforcement agents, and the registration process that is required to set up the trust mechanism between the agent and Oracle Access Manager 11g SSO.

- About Policy-Enforcement Agents
- About the Pre-Registered IDM Domain Agent
- About Registering Partners (Agents and Applications)

- About File System Changes and Artifacts for Registered Agents

5.2.1 About Policy-Enforcement Agents

With Oracle Access Manager 11g, each policy enforcement Agent acts as a filter for HTTP requests. Your deployment can include the following agents in any combination:

- **OAM Agent**s:
  - OAM 11g WebGates
  - OAM 10g WebGates
- **OSSO Agent**s: mod_osso is part of the (still supported) OracleAS 10g single sign-on (OSSO) solution that authenticates users at a central OSSO Server.

  After registering 10g mod_osso as an agent, OAM 11g gives mod_osso the redirect URL for the user based on the authentication scheme associated with the OAM policy defined for the resource.

  > **Note:** The mod_osso module is an Oracle HTTP Server module that provides authentication to OracleAS applications.

Unless explicitly stated, details about OAM Agents apply equally to WebGates and AccessGates:

- WebGate is out of an box access client. This Web server access client intercepts HTTP requests for Web resources and forwards these to the OAM 11g Server. WebGates for various Web servers are shipped with Oracle Access Manager.
- AccessGate is a custom access client created for use with non-Web applications. This custom access client must be specifically developed using the Software Developer Kit (SDK) and Oracle Access Manager APIs, either by you or by Oracle. An AccessGate is a form of access client that processes requests for Web and non-Web resources (non-HTTP) from users or applications.

Table 5–1 provides information about all agents for OAM 11g.

Table 5–1 Agents for OAM 11g

Agents	Description
OSSO Agent (mod_osso 10g)	Following registration with OAM 11g, the mod_osso module: ■ Checks for an existing valid Oracle HTTP Server cookie ■ Redirects to the OAM Server if needed to contact the directory during authentication ■ Decrypts the encrypted user identity populated by the OSSO server ■ Sets the headers with user attributes
11g WebGates	After installation and registration with OAM 11g, 11g WebGates communicate with Oracle Access Manager 11g services using the OAM Proxy to "sanitize" the request and respond identically for all agents.

Table 5–1 (Cont.) Agents for OAM 11g

Agents	Description
10g WebGates	After installation and registration, OAM 10g WebGates directly communicate with Oracle Access Manager 11g services through a JAVA-based OAM proxy that acts as a bridge. OAM 10g WebGates include: ■ Freshly installed 10g WebGates for OAM 11g can support Web servers other than Oracle HTTP Server as described in Chapter 17. ■ Legacy 10g WebGates currently operating with OAM 10g and combined with OSSO as described in the *10g Oracle Access Manager Integration Guide*. ■ Legacy 10g WebGates configured as the Identity Assertion Provider (IAP) for SSO (for applications using WebLogic container-based security with OAM 10g, as described in the *Oracle Fusion Middleware Application Security Guide*). ■ Legacy 10g WebGates currently operating with Web Applications coded for Oracle ADF Security and the OPSS SSO Framework as described in Appendix C. See Also IDMDomainAgent in this table.
AccessGates	Only authentication and authorization is supported (not policy modification) for AccessGates.
IDMDomainAgent	The IDM Domain Agent provides single sign-on functionality for the IDM Administration Console. The IDM Domain Agent is installed and pre-configured as part of the Oracle Access Manager 11g Server installation and configuration. See Also: "About the Pre-Registered IDM Domain Agent" on page 5-4.

Table 5–2 provides a comparison of the agent types that are compatible with OAM 11g as well as the differences between OAM 11g and earlier agents (organized in columns).

Table 5–2 Comparing Agent Types and Differences

	OAM 11g	OAM 10g	OSSO 10g
Available agents	OAM Agents ■ 11g WebGate ■ 10g WebGate ■ IDM Domain Agent OSSO Agents ■ 10g mod_osso (partner)	WebGate and AccessGate ■ Resource WebGate (RWG) ■ Authentication WebGate (AWG) With OAM 10g, WebGate installation included Web server configuration.	■ mod_osso
Remote Registration Tool	Available to register agents to operate with OAM 11g.	Available to register agents operating with OAM 11g. **Note**: There was no remote registration equivalent for OAM 10g.	Available for agents operating with OAM 11g **Note**: There was no remote registration equivalent before OAM 11g.
Login Forms	/oam/pages/css/login_page.css	No login forms provided and used with a 10g WebGate are relevant for OAM 11g.	unchanged
logout.html	See Chapter 11 for details about configuring logout for 10g and 11g Agents	logout.html requires specific details when using a 10g WebGate with OAM 11g. See Chapter 11.	There is no change required for OAM 11g with mod_osso (OSSO Agents). Applications that use dynamic directives require no entry in mod_osso.conf. Instead, protection is written into the application as one or more dynamic directives.

Table 5–2 (Cont.) Comparing Agent Types and Differences

	OAM 11g	OAM 10g	OSSO 10g
Multiple network domain support	OAM 11g supports cross-network-domain single sign-on out of the box. Oracle recommends you use Oracle Identity Federation for this situation.	OAM provides a proprietary multiple network domain SSO capability that predates Oracle Identity Federation. If this is implemented in your OAM 10g deployment, you can register OAM 10g Agents with OAM 11g to continue this support.	
Cryptographic keys Notes: The protocols used to secure information exchange on the Internet.	■ One per-agent secret key shared between 11g WebGate and OAM Server ■ One OAM Server key **Note**: One key is generated and used per registered mod_osso or 11g WebGate. However, one single key is generated for all 10g WebGates.	There is just one global shared secret key per OAM deployment which is used by all the WebGates	■ One key per partner shared between mod_osso and OSSO server ■ OSSO server's own key ■ One global key per OSSO setup for the GITO domain cookie
Keys storage	■ **Agent side**: A per agent key is stored locally in the Oracle Secret Store in a wallet file ■ **OAM 11g server side:** A per agent key, and server key, are stored in the credential store on the server side	Global shared secret stored in the directory server only (not accessible to WebGate)	■ **mod_osso side**: partner keys and GITO global key stored locally in obfuscated configuration file ■ **OSSO server side**: partner keys, GITO global key, and server key are all stored in the directory server

Administrators can use either the OAM Administration Console or the remote registration tool to:

- Register a freshly installed OAM 11g Agent
- Provision a legacy (or freshly installed) OAM 10 WebGate for use with OAM 11g, as described in Chapter 17.
- Register an OSSO 10g Agent (mod_osso)

> **Note:** You can upgrade OracleAS 10g SSO, as described in the *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management*. During the upgrade, OSSO agents are registered with OAM 11g. See Appendix B, "Co-existence Overview: OAM 11g and OSSO 10g".

5.2.2 About the Pre-Registered IDM Domain Agent

The IDM Domain Agent provides single sign-on functionality for the IDM Administration Console. The IDM Domain Agent is installed and pre-configured as part of the Oracle Access Manager 11g Server installation and configuration.

The IDM Domain agent is a domain-wide agent:

- Once deployed, the IDM Domain agent is installed on every server in the domain
- Unless disabled, every request coming into the WebLogic Application Server is evaluated and processed by the IDM Domain Agent
- Configuration details are located under the 10g Webgates node (Policy Configuration tab) in the OAM Administration Console.

Certain IDM Domain Agent configuration elements are available in the WebLogic Administration Console (in the Security Provider section) and others in the OAM Administration Console.

WebLogic Administration Console, Security Provider Settings

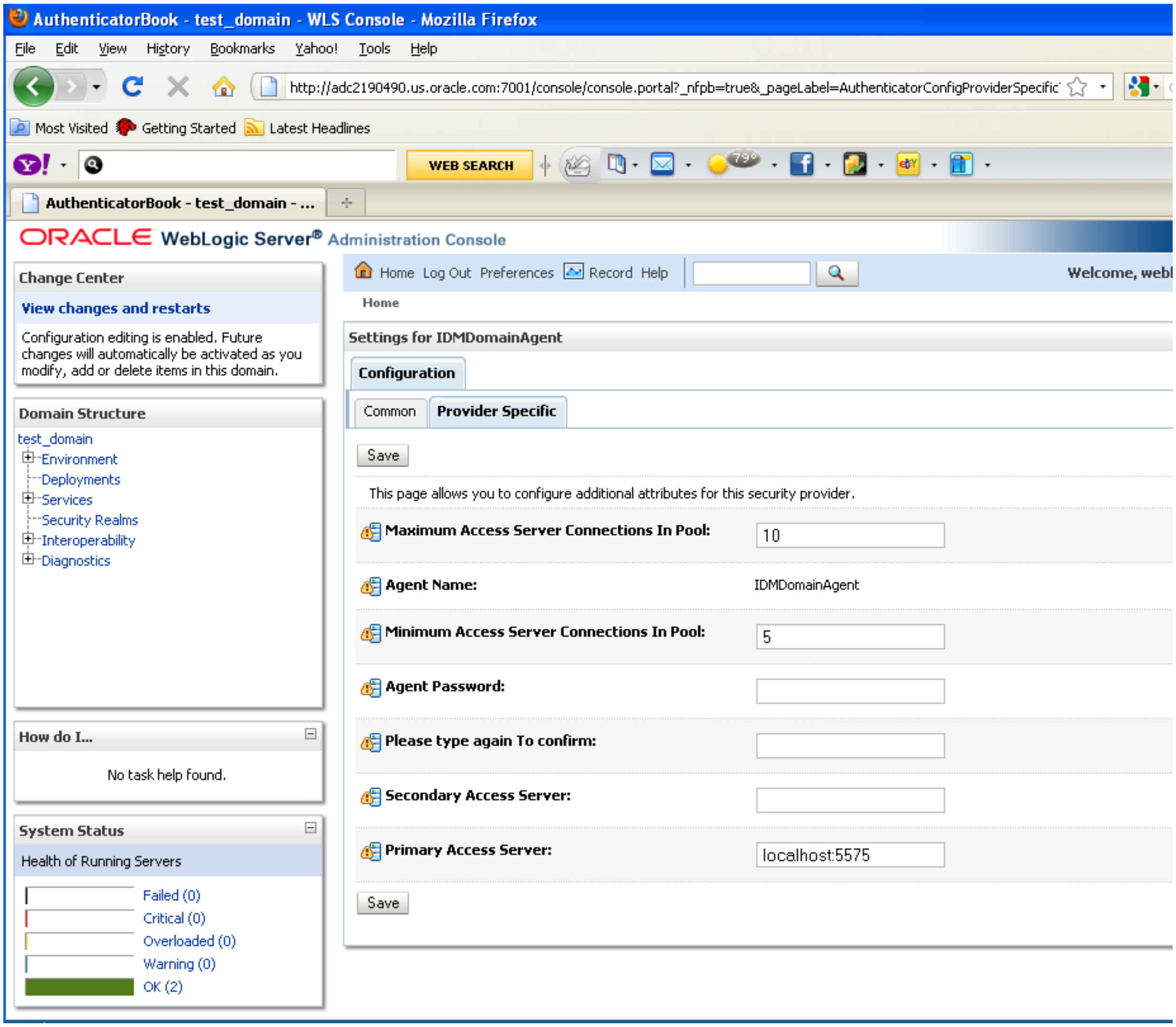
In the Security Provider section of the WebLogic Administration Console are five bootstrap configuration parameters.

While Oracle recommends that you retain these without making changes, there are circumstances where you might need to change one of the following parameters:

- Primary Access Server: You can replace this value with information for your actual OAM Server. The default value (localhost:5575) can be replaced with information for your actual OAM Server if more than one host is part of the IDM Domain.
- Agent Password: By default there is no password. However, you can add one here if you want to establish a password for the IDMDomainAgent connection to the OAM Server through the NetPoint (now Oracle) Access Protocol (NAP or OAP).

Figure 5–2 illustrates the default Security Provider settings for the IDM Domain Agent.

Figure 5–1 IDM Domain Agent Configuration in the WebLogic Administration Console


AuthenticatorBook - test_domain - WLS Console - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://adc2190490.us.oracle.com:7001/console/console.portal?_nfpb=true&_pageLabel=AuthenticatorConfigProviderSpecific
Most Visited Getting Started Latest Headlines
WEB SEARCH
AuthenticatorBook - test_domain - ...
ORACLE WebLogic Server® Administration Console
Change Center
View changes and restarts
Home Log Out Preferences Record Help
Welcome, webl
Home
Domain Structure
test_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
No task help found.
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (2)
Configuration
Common
Provider Specific
Save

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Settings for IDMDomainAgent

This page allows you to configure additional attributes for this security provider.

Maximum Access Server Connections In Pool:	10
Agent Name:	IDMDomainAgent
Minimum Access Server Connections In Pool:	5
Agent Password:	
Please type again To confirm:	
Secondary Access Server:	
Primary Access Server:	localhost:5575

OAM Administration Console, IDMDomainAgent Registration Page

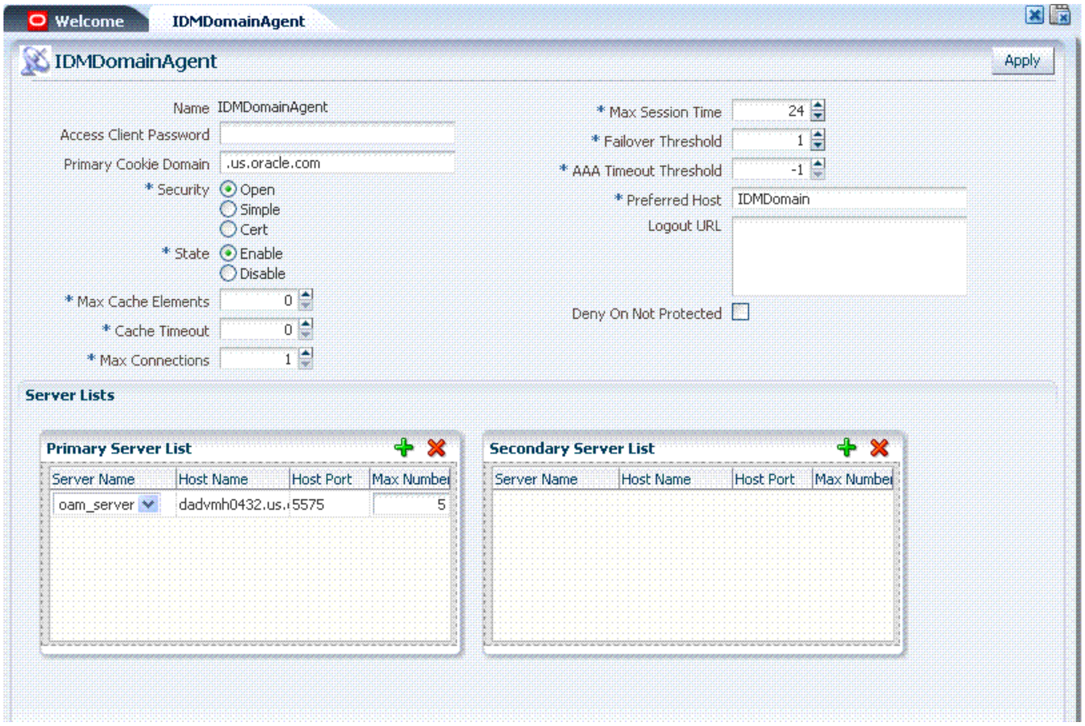
The IDMDomainAgent registration page is like all other OAM agent registration pages.

- Security Mode: Open is the only security mode available for the IDMDomainAgent. This cannot be changed.
- Preferred Host: IDMDomain is the pre-configured host required by this agent

> **Note:** The Access Client Password here must match the Agent Password in the WebLogic Administration Console. If you changed the Agent Password, you must also change the Access Client Password.

Figure 5–2 illustrates the default configuration characteristics for the IDM Domain Agent.

Figure 5–2 IDM Domain Agent Default Characteristics



IDMDomainAgent

Name	IDMDomainAgent
Access Client Password	
Primary Cookie Domain	.us.oracle.com
* Security	● Open ○ Simple ○ Cert
* State	● Enable ○ Disable
* Max Cache Elements	0
* Cache Timeout	0
* Max Connections	1
* Max Session Time	24
* Failover Threshold	1
* AAA Timeout Threshold	-1
* Preferred Host	IDMDomain
Logout URL	
Deny On Not Protected	□

Server Lists

Primary Server List

Server Name	Host Name	Host Port	Max Number
oam_server	dadvmh0432.us.	5575	5

Secondary Server List

Server Name	Host Name	Host Port	Max Number

With few exceptions, all agent registration elements are the same. Table 5–3 outlines the differences. All elements are described in Table 5–6, " Expanded OAM 11g and 10g WebGate Agent Elements and Defaults".

Table 5–3

Element	11g Webgate	10g WebGate	IDMDomainAgent
Primary Cookie Domain	N/A	x	x
Token Validity Period	x	N/A	N/A
Preferred Host	x	x	IDMDomain
Logout Callback URL	x	N/A	N/A

Table 5–3 (Cont.)

Element	11g Webgate	10g WebGate	IDMDomainAgent
Logout Redirect URL	x	N/A	N/A
Logout Target URL	x	N/A	N/A

Table 5–4 describes the resources protected by the IDM Domain Agent. Oracle recommends that you do not make any additions or changes. The WebLogic Administration Console (/console/.../*) and Fusion Middleware Enterprise Manager (/em/.../*) are not protected.

Table 5–4 Resources Protected by IDMDomainAgent

Resource	Description
OAM Console	/oamconsole/.../*
OINAV	/oinav/.../*
APM	/apm/.../*
OAAM Console	/oaam_admin/.../*
OIM Resources Using LDAPScheme	/admin/faces/pages/Admin.jspx /oim/faces/pages/Self.jspx /oim/faces/pages/Admin.jspx /xlWebApp/.../* /Nexaweb/.../*
OIM Resources Using OIMScheme	/admin/faces/pages/pwdmgmt.jspx
OIM Resources Using AnonymousScheme	/oim/faces/pages/USelf.jspx /admin/faces/pages/forgotpwd.jspx /admin/faces/pages/accountlocked.jspx /admin/.../*.js /admin/.../*.css /admin/.../*.png /admin/.../*.gif /oim/.../*.js /oim/.../*.css /oim/.../*.png /oim/.../*.gif

You can replace this agent with a 10g WebGate, as described in Chapter 17, "Managing OAM 10g WebGates with OAM 11g".

5.2.3 About Registering Partners (Agents and Applications)

Only registered policy enforcement agents can communicate with an OAM Server, and process information when a user attempts to access a protected resource.

OAM administrators must register the OAM Agent or OSSO Agent that resides on the computer hosting the application to be protected. Agent registration can include partner registration by automatically creating an application domain and default policies.

Following registration, agent details appear in the OAM Administration Console and are propagated to all Managed Servers in the cluster. If you choose to automatically create policies during agent registration, you can also view and manage the application domain and policies that were registered with the partner application.

Note: Registering an Agent is also known as "registering a partner application" or "registering a partner application with OAM".

During registration, the Agent is presumed to be on the same Web server as the application it is protecting. However, the Agent can be on a proxy Web server and the application can be on a different host.

During Agent registration:

- One key is generated per agent, accessible to the WebGate through a local wallet file on the client host, and to OAM Server through the Java Key Store on the server side.

  The Agent specific key must be accessible to WebGates through a secure local storage on the client machine. See Table 5–2.

- A key is generated for the partner (application) during registration. (except for 10g WebGate agents).
- An OAM application domain is created, named after the Agent, and populated with default authentication and authorization policies. The new application domain uses the same host identifier that was specified for the Agent during registration. For more information on application domains, see Chapter 9.

After registration, the agent can monitor attempts to access a Web site and use OAM Servers to provide authentication and authorization services before completing the request. Administrators can view, modify, or remove a registered agent using either the Administration Console or custom WLST commands for OAM 11g.

For more information, see:

- Registering and Managing WebGate Agents Using the Administration Console
- Registering and Managing OSSO Agents Using the Administration Console

See Also:

- Chapter 6, "Registering Partners (Agents and Applications) Remotely"
- Chapter 17, "Managing OAM 10g WebGates with OAM 11g"
- Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"

5.2.4 About File System Changes and Artifacts for Registered Agents

When you register an agent using the OAM Administration Console, a new file system directory is created for the Agent on the OAM Administration Console host:

<MW_HOME>/user_projects/domains/<*domain_name*>/output/<*agent_name*>

This new directory includes generated files for the registered agent that must be copied in to the agent's installation directory.

11g WebGate: Copy generated files to *WebGate_instance_dir*/webgate/config (for example, WebTier_Middleware_Home/Oracle_WT1/instances/instance1/config/OHS/ohs1/webgate/config)

10g WebGate: copy generated files to *WebGate_install_dir*/webgate/config

mod_osso: Copy generated files to *OHS_webserver_dir*/oracle/product /11.1.1/as_1/instances/*instance1*/config/OHS/*ohs1*/osso/

Generated files include the following:

- ObAccessClient.xml (for WebGates)

The pre-registered IDM Domain Agent does not use ObAccessClient.xml for bootstrap or configuration.

With OAM 10g, ObAccessClient.xml was generated on the agent side when the configureWebGate tool was run. With OAM 11g, you can use the Administration Console or the remote registration tool to create ObAccessClient.xml.

- cwallet.sso (for 11g WebGates, regardless of the transport security mode)
- certificate and password files for secure communication, if needed. For example, password.xml file or aaa_cert.pem and aaa_key.pem files.

Note: When editing an 11g WebGate registration, password.xml is updated only when the mode is changed from Open to Cert or Simple to Cert. In cert mode, once generated, password.xml cannot be updated. Editing the agent Key Password does not result in creation of a new password.xml.

- osso.conf file (for OSSO Agents)

Before WebGate startup, copy the ObAccessClient file from the generated location to the WebGate installation directory on the computer hosting the WebGate instance.

During WebGate run time, the ObAccessClient file is updated automatically when a change is discovered during periodic update checks.

Simple Mode Global passphrase stored in a nominally encrypted file: password.xml

Cert Mode:

- PEM KeyStore Alias
- PEM KeyStore Alias Password

5.3 Registering and Managing WebGate Agents Using the Administration Console

This section describes how to manage OAM Agents using the Administration Console. Topics include:

- About the Create OAM Agent Page
- Registering a WebGate Agent
- Viewing or Editing a WebGate Agent Registration
- Deleting a WebGate Agent Registration

See Also: The following chapters as needed

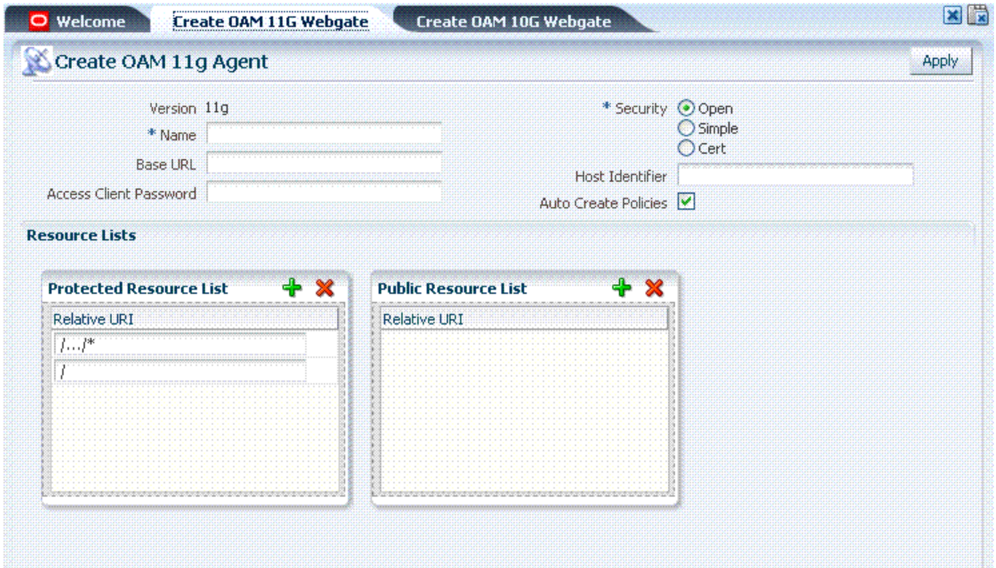
- Chapter 17, "Managing OAM 10g WebGates with OAM 11g"
- Chapter 6, "Registering Partners (Agents and Applications) Remotely"

5.3.1 About the Create OAM Agent Page

This topic describes OAM Agent registration using the OAM Administration Console.

Figure 5–3 illustrates the Create OAM 11g Agent page, under the System Configuration tab in the Administration Console. The Create OAM 10g Agent page includes the same elements. The page requests minimal information to streamline registration. Required information is identified by the asterisk (*).

Figure 5–3 Create OAM 11g Agent Page



Create OAM 11g Agent

Version 11g

* Name

Base URL

Access Client Password

* Security ● Open ○ Simple ○ Cert

Host Identifier

Auto Create Policies ☑

Resource Lists

Protected Resource List

Relative URI
/…/*
/

Public Resource List

Relative URI

The page includes named text fields where you enter requested information. Table 5–5 describes each of elements on the Create OAM Agent page. This is the same information that is requested when you use the remote registration tool with the simplified OAM request template as described in Chapter 6.

Table 5–5 Create OAM Agent Pages for OAM 10g and 11g Agents

OAM Agent Element	Description
Agent Name	The identifying name for this WebGate Agent. This is often the name of the computer that is hosting the Web server used by WebGate. **Note**: If the Agent Name exists, an error occurs and registration fails. If the host identifier exists, the unique Agent Base URL is added to the existing host identifier and registration proceeds.
Agent Base URL Optional for OAM 10g and 11g agents.	The host and port of the computer on which the Web server for the agent is installed. For example, http://*my_host:port* or https://*my_host:port*. The port number is optional. **Note**: A particular Agent Base URL can be registered once only. There is a one-to-one mapping from the Agent's Base URL to the Web server domain on which the WebGate is installed (as specified with the <hostidentifier> element). However, one domain can have multiple Agent's Base URLs.
Access Client Password	An optional, unique password for this WebGate, which was assigned during WebGate registration. When a registered WebGate connects to an OAM 11g Server, the password is used for authentication to prevent unauthorized WebGates from connecting to OAM 11g Servers and obtaining policy information.

Table 5–5 (Cont.) Create OAM Agent Pages for OAM 10g and 11g Agents

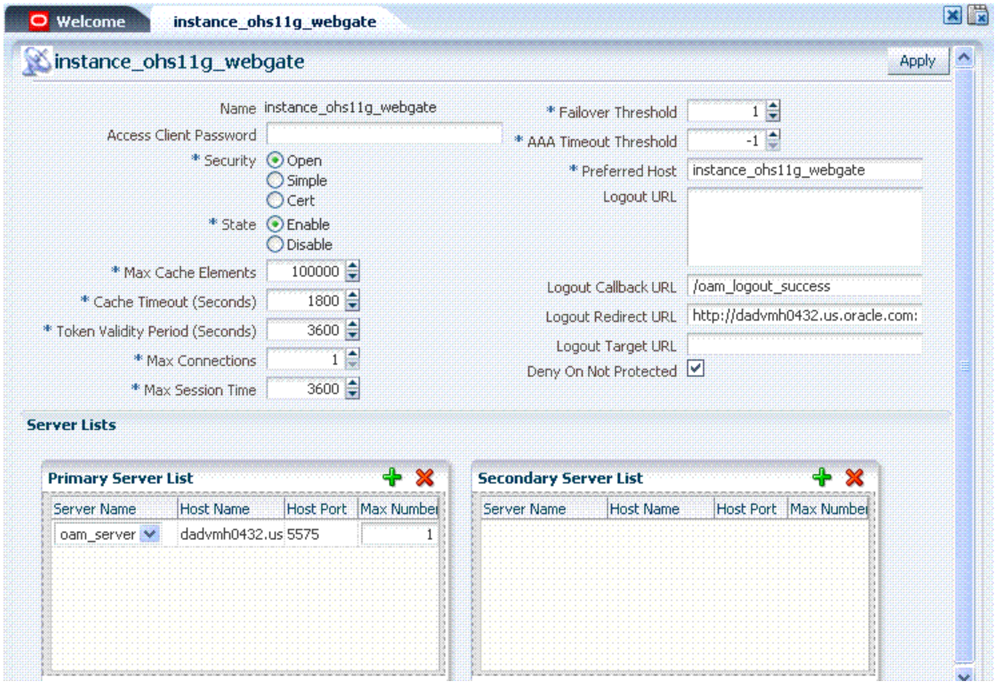
OAM Agent Element	Description
Security	Level of communication transport security between the Agent and the OAM Server (this must match the level specified for the OAM Server): ■ Open--No transport security ■ Simple--SSL v3/TLS v1.0 secure transport using dynamically generated session keys ■ Cert--SSL v3/TLS v1.0 secure transport using server side x.509 certificates. Choosing this option displays a field where you can enter the Agent Key Password, discussed separately within this table. **Note**: For more information on Simple and Cert modes, and encrypting the private key using the DES algorithm, see Appendix E.
Host Identifier	This identifier represents the Web server host.
Auto Create Policies	During agent registration, you can have authentication and authorization policies created automatically. This option is checked (enabled) by default. Default: Enabled **Note**: If you already have a domain and policies registered, you can simply add new resources to it. If you clear this option (no check), no application domain or policies are generated automatically.

Protected Resource (URI) List	URIs for the protected application: /myapp/login, for example. Each URI for the protected application should be specified in a new row of the table for the Protected Resource List.
	Default: 2 resources are protected by default.
	`/.../*`
	`/`
	The default matches any sequence of characters within zero or more intermediate levels spanning multiple directories.
	See Also: "About the Resource URL" on page 9-14.

Public Resource (URI) List	Each public application should be specified in a new row of the table for the Public Resource List. Add a field and enter URI values for the public applications and resources. Each URI should be specified in a new row of the table for the Public Resource List.

To help streamline Agent registration, additional elements are concealed and default values are applied. When you view or edit an Agent's page in the Administration Console, all elements and values are revealed, as shown in Figure 5–4. Most elements are the same as those you define when using the remote registration tool with the expanded OAM template, as described in Chapter 6.

Figure 5–4 OAM 11g Webgate Page with Defaults



instance_ohs11g_webgate

Name	instance_ohs11g_webgate
Access Client Password	
* Security	● Open ○ Simple ○ Cert
* State	● Enable ○ Disable
* Max Cache Elements	100000
* Cache Timeout (Seconds)	1800
* Token Validity Period (Seconds)	3600
* Max Connections	1
* Max Session Time	3600
* Failover Threshold	1
* AAA Timeout Threshold	-1
* Preferred Host	instance_ohs11g_webgate
Logout URL	
Logout Callback URL	/oam_logout_success
Logout Redirect URL	http://dadvmh0432.us.oracle.com:
Logout Target URL	
Deny On Not Protected	☑

Server Lists

Primary Server List

Server Name	Host Name	Host Port	Max Number
oam_server	dadvmh0432.us	5575	1

Secondary Server List

Server Name	Host Name	Host Port	Max Number

Table 5–6 summarizes elements on an expanded registration. Additional settings revealed here are used by the OAM Proxy. ObAccessClient.xml is populated with values after agent registration, whether you use the OAM Administration Console as described here or the remote registration tool as described in Chapter 6.

Table 5–6 Expanded OAM 11g and 10g WebGate Agent Elements and Defaults

OAM Agent Element	Description
Agent Name	Name of the OAM Agent.
Access Client Password	Optional, unique password for the OAM Agent. When the agent connects to an OAM Server, it uses the password to authenticate itself to the server. This prevents unauthorized agents from connecting and obtaining policy information.
Primary Cookie Domain 10g WebGate only.	This parameter describes the Web server domain on which the OAM Agent is deployed, for instance,*.acompany.com*. You must configure the cookie domain to enable single sign-on among Web servers. Specifically, the Web servers for which you configure single sign-on must have the same Primary Cookie Domain value. The OAM Agent uses this parameter to create the ObSSOCookie authentication cookie. This parameter defines which Web servers participate within the cookie domain and have the ability to receive and update the ObSSOCookie. This cookie domain is not used to populate the ObSSOCookie; rather it defines which domain the ObSSOCookie is valid for, and which Web servers have the ability to accept and change the ObSSOCookie contents. Default: If the client side domain can be determined during registration, the Primary Cookie Domain is populated with that value. However, if no domain is found, there is no value and WebGate uses the host-based cookie. **Note**: The more general the domain name, the more inclusive your single sign-on implementation will be. For example, if you specify b.com as your primary cookie domain, users will be able to perform single sign-on for resources on b.com and on a.b.com. However, if you specify a.b.com as your primary cookie domain, users will have to re-authenticate when they request resources on b.com.
State Set only in the OAM Administration Console.	Specifies whether the OAM Agent is enabled or disabled. Default = Enabled

Table 5–6 (Cont.) Expanded OAM 11g and 10g WebGate Agent Elements and Defaults

OAM Agent Element	Description
Max Cache Elements	Number of elements maintained in the cache. Cache elements are the following: ■ URLs—The URL cache maintains information about a URL, including if it is protected and the authentication scheme used if it is protected. ■ Authentication schemes—This cache stores authentication scheme information for a specific authentication scheme ID. The value of this setting refers to the maximum consolidated count for elements in both of these caches. Default = 100000
Cache Timeout (seconds)	Amount of time cached information remains in the OAM Agent cache when the information is neither used nor referenced. Default = 1800 (seconds)
Token Validity Period 11g WebGate only	Maximum valid time period for an agent token (the content of OAMAuthnCookie for 11g WebGate). The equivalent to tokenValidityPeriod field in 10g WebGates is called cookieSessionTime. Default = 3600 (seconds) Note: For OAM 10g WebGates, use Cookie Session Time to set the Token Validity Period.
Max Connections	The maximum number of connections that this OAM Agent can establish with the OAM Server. This number must be the same as (or greater than) the number of connections that are actually associated with this agent. Default = 1
Max Session Time	Maximum amount of time in seconds that a user's authentication session is valid, regardless of their activity. At the expiration of this session time, the user is re-challenged for authentication. This is a forced logout. Default = 24 (hours) A value of 0 disables this timeout setting.
Failover Threshold	Number representing the point when this OAM Agent opens connections to a Secondary OAM Server. Default = 1 For example, if you type 30 in this field and the number of connections to primary OAM Server falls to 29, this OAM Agent opens connections to secondary OAM Server.

Table 5–6 (Cont.) Expanded OAM 11g and 10g WebGate Agent Elements and Defaults

OAM Agent Element	Description
AAA Timeout Threshold	Number (in seconds) to wait for a response from the OAM Server. If this parameter is set, it is used as an application TCP/IP timeout instead of the default TCP/IP timeout. Default = -1 (default network TCP/IP timeout is used) A typical value for this parameter is between 30 and 60 seconds. If set to a very low value, the socket connection can be closed before a reply from Access Server is received, resulting in an error. For example, suppose an OAM Agent is configured to talk to one primary OAM Server and one secondary OAM Server. If the network wire is pulled from the primary OAM Server, the OAM Agent waits for the TCP/IP timeout to learn that there is no connection to the primary OAM Server. The WebGate tries to reestablish the connections to available servers starting with the primary Access Server. Again, the OAM Agent waits for the TCP/IP timeout to determine if a connection can be established. If it cannot, the next server in the list is tried. If a connection can be established to another OAM Server (either a primary or secondary), the requests are re-routed. However this can take longer than desired. When finding new connections, OAM Agent checks the list of available servers in the order specified in its configuration. If there is only one primary OAM Server and one secondary OAM Server specified, and the connection to the primary OAM Server times out, the OAM Agent still tries the primary OAM Server first. As a result, the OAM Agent cannot send requests to an OAM Server for a period greater than twice the setting in the OAM Server Timeout Threshold. If the OAM Server takes longer to service a request than the value of the timeout threshold, the OAM Agent abandons the request and retries the request on a new connection. Note that the new connection that is returned from the connection pool can be to the same OAM Server, depending on your connection pool settings. Also, other OAM Server may also take longer to process the request than the time specified on the threshold. In these cases, the OAM Agent can continue to retry the request until the OAM Server is shut down.
Idle Session Timeout 10g WebGates only	Default: 3600 Release 7.0.4 WebGates enforced their own idle session timeout only. 10.1.4.0.1 WebGates enforced the most restrictive timeout value among all WebGates the token had visited. With 10*g* (10.1.4.3), the 7.0.4 behavior was reinstated as the default with this element. To set idleSessionTimeoutLogic: ■ The default value of `leastComponentIdleTimeout` instructs the WebGate to use the "most restrictive" timeout value for idle session timeout enforcement. ■ A value of `currentComponentIdleTimeout` instructs the WebGates to use the "current WebGate" timeout value for idle session timeout enforcement.
Preferred Host	Specifies 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 it was defined in a user's HTTP request. The Preferred Host function prevents security holes that can be inadvertently created if a host's identifier is not included in the Host Identifiers list. However, it cannot be used with virtual Web hosting. For virtual hosting, you must use the Host Identifiers feature.
Deny on Not Protected	Denies access to all resources to which access is not explicitly allowed by a rule or policy. Always enabled in 11g WebGate registration, and cannot be changed. On a 10g WebGate registration page, you can choose to disable this.

Table 5–6 (Cont.) Expanded OAM 11g and 10g WebGate Agent Elements and Defaults

OAM Agent Element	Description
Logout URL	The Logout URL triggers the logout handler, which removes the cookie (ObSSOCookie for 10g WebGates; OAMAuthnCookie for 11g WebGates) and requires the user to re-authenticate the next time he accesses a resource protected by Oracle Access Manager. ■ If there is a match, the WebGate logout handler is triggered. ■ If Logout URL is not configured the request URL is checked for "logout." and, if found (except "logout.gif" and "logout.jpg"), also triggers the logout handler. Default = Note: This is the standard OAM 10g WebGate configuration parameter used to trigger initial logout. See Also: Chapter 11 for additional steps required for configuring logout for OAM 10g WebGates registered with OAM 11g.
Additional Logout for 11g WebGate Only	For OAM 11g WebGate single sign-off behavior, specific logout elements and values automate the redirect to a central logout URL, callback URL, and end_URL. This replaces 10g WebGate single sign-off only through a customized local logout page.
Logout Callback URL 11g WebGate only	The URL to oam_logout_success, which clears cookies during the call back. This can be a URI format without host:port (recommended), where the OAM Server calls back on the host:port of the original resource request. For example: Default = /oam_logout_success This can also be a full URL format with a host:port, where OAM 11g server calls back directly without reconstructing callback URL. When the request URL matches the Logout Callback URL, WebGate clear its cookies and streams an image gif in the response. This is similar to OSSO agent behavior. When WebGate redirects to the server logout page, it records an "end" URL as a query parameter (`end_url=http://host:port/..."`), which becomes the landing page that the OAM 11g Server redirects back to after logout. Other OAM 11g services support the central logout page on the server. The end_url relies on the target URL query parameter passed from OPSS integrated applications.
Logout Redirect URL 11g WebGate only	This parameter is automatically populated after agent registration completes.By default, this is based on the OAM Server host name with a default port of 14200. For example: Default = http://*OAMServer_host*:14200/oam/server/logout The Logout URL triggers the logout handler, which removes the OAMAuthnCookie_<*host:port*>_<*random number*> and requires the user to re-authenticate the next time he accesses a resource protected by Oracle Access Manager. See Also: Chapter 11, "Configuring Centralized Logout for 11g WebGate with OAM 11g Server"

Table 5–6 (Cont.) Expanded OAM 11g and 10g WebGate Agent Elements and Defaults

OAM Agent Element	Description
Logout Target URL 11g WebGate only	The value is the name for the query parameter that the OPSS applications passes to WebGate during logout; the query parameter specifies the target URL of the landing page after logout completes. Default: end_url See Also: Chapter 11, "Configuring Centralized Logout for 11g WebGate with OAM 11g Server"
Primary Server List	Identifies Primary Server details for this Agent. The default is based on the OAM Server: ■ Server Name ■ Host Name ■ Host Port ■ Max Number (maximum connections this WebGate will establish with the OAM Server (not the maximum total connections the WebGate can establish with all OAM Servers).)
Secondary Server List	Identifies Secondary OAM Server details for this agent, which must be specified manually: ■ Server Name ■ Host Name ■ Host Port ■ Max Number (maximum connections this WebGate will establish with the OAM Server (not the maximum total connections the WebGate can establish with all OAM Servers).)

5.3.2 Searching for a WebGate Agent Registration

Users with valid OAM Administrator credentials can perform the following procedure to search for an OAM Agent using the Administration Console.

Prerequisites

The WebGate must be a registered agent of Oracle Access Manager 11g.

To search for an OAM Agent registration

1. Activate the System Configuration tab.
2. From the search type list, choose the relevant type (10g Webgates or 11g Webgates) to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   ```
   my_OAM_Agent
   ```

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit command button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.

6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

5.3.3 Registering a WebGate Agent

You can register a WebGate agent before you install it. Users with valid OAM Administrator credentials can perform the following task to register an OAM Agent using the Administration Console.

See Also:

- About the Create OAM Agent Page
- Provisioning a 10g WebGate with OAM 11g

Note: During agent registration, at least one OAM Server instance must be running in the same mode as the agent. Otherwise, agent registration fails.

After agent registration, you can change the communication mode of the OAM Server if needed. Communication between the agent and server continues to work as long as the WebGate mode is at least at the same level as the OAM Server mode or higher. See Appendix E.

Prerequisites

Confirm that at least one OAM Server is running in the same mode as the agent to be registered.

To register a WebGate agent

1. Log in to the OAM Administration Console as usual.
2. From the OAM Administration Console Welcome page, Agent Configuration panel, click one of the following links to open a fresh page:
   - Add OAM 11g Agent
   - Add OAM 10g Agent (see also Chapter 17)

   Alternatively: From the System Configuration tab, expand the Agents node, the OAM Agents node, and either the 11g Webgates or 10g Webgates node, then click the Create command button in the tool bar.
3. On the Create: OAM Agent page, enter required details (those with an *) to register this OAM Agent, as shown in Table 5–5:
4. **Protected Resource List**: In this table, enter individual resource URLs to be protected by this OAM Agent, as shown in Table 5–5.
5. **Public Resource List**: In this table, enter individual resource URLs to be public (not protected), as shown in Table 5–5.
6. Click Apply to submit the registration (or close the page without applying changes).
7. Check the Confirmation window for the location of generated artifacts and then close the window.
8. In the navigation tree, confirm the Agent name is listed.

Note: If you are provisioning an OAM 10g WebGate, skip Step 9 for now and go to Chapter 17.

9. Perform the following steps to copy the artifacts to the WebGate installation directory (or install WebGate and then copy these artifacts):

   a. On the OAM Administration Console host, locate the updated OAM Agent ObAccessClient.xml configuration file (and any certificate artifacts). For example:

      $DOMAIN_HOME/output/$*Agent_Name*/ObAccessClient.xml

   b. On the OAM Agent host, copy artifacts (to the following WebGate directory path). For example:

      11g WebGate: *11gWebGate_instance_dir*/webgate/config/ObAccessClient.xml (for instance *WebTier_Middleware_Home*/Oracle_WT1/instances1/config/OHS/ohs1/webgate/config/ObAccessClient.xml)

      10g WebGate: $*WebGate_install_dir*/oblix/lib/ObAccessClient.xml

   c. Restart the OAM Server that is hosting the Agent and proceed to Part III, "Single Sign-on, Policies, and Testing".

5.3.4 Viewing or Editing a WebGate Agent Registration

Users with valid OAM Administrator credentials can change any setting for a registered OAM Agent using the Administration Console, as described in the following procedure. For example, you might want to revise the timeout threshold or other settings used by the OAM Proxy.

After changes, updated details are propagated through a runtime configuration update process. There is usually no need to copy the artifacts over to WebGate configuration area.

Note: Artifacts need only be copied to the WebGate directory path if the agent name, access client password, or security mode is changed.

Prerequisites

The agent must be registered and available in the Oracle Access Manager Administration Console.

See Also:

- Searching for a WebGate Agent Registration
- About the Create OAM Agent Page

To view or modify details for a registered WebGate Agent

1. From the System Configuration tab, navigation tree, expand the following nodes as needed:

   Agents
   OAM Agents
   11g Webgates (or 10g Webgates)

2. Double-click the desired agent's name to display the registration page.
3. Modify Agent details, and Primary or Secondary Server details, as needed (see Table 5–5 and Table 5–6).
4. Click Apply to submit changes and dismiss the Confirmation window (or close the page without applying changes).
5. Perform the following steps to copy artifacts if needed:

> **Note:** Artifacts need only be copied to the WebGate directory path if the agent name, or agent client password, or security mode is changed. See Chapter 17 if you are provisioning an OAM 10g WebGate.

   a. On the OAM Administration Console host, locate the updated OAM Agent ObAccessClient.xml configuration file (and any certificate artifacts). For example:

      $DOMAIN_HOME/output/$*Agent_Name*/ObAccessClient.xml

   b. On the OAM Agent host, copy artifacts (to the following WebGate directory path). For example:

      11g WebGate: *11gWebGate_instance_dir*/webgate/config/ObAccessClient.xml *(for instance, WebTier_Middleware_Home*/Oracle_WT1/instances1/config/OHS/ohs1/webgate/config/ObAccessClient.xml)

      10g WebGate: $*WebGate_install_dir*/oblix/lib/ObAccessClient.xml

   c. Restart the OAM Server that is hosting the Agent and proceed to Part III, "Single Sign-on, Policies, and Testing".

   d. See Appendix E, "Securing Communication with OAM 11g", if needed.

5.3.5 Deleting a WebGate Agent Registration

Users with valid OAM Administrator credentials can perform the following procedure to delete a registered OAM Agent from the Administration Console.

> **Note:** Deleting an agent registration remove only the registration (not the associated host identifier, application domain, resources, or the agent itself).

See Also:

- Searching for a WebGate Agent Registration
- About the Create OAM Agent Page
- Removing a 10g WebGate from the OAM 11g Deployment in Chapter 17

Prerequisites

Evaluate the application domain, resources, and policies associated with this agent and ensure that these are configured to use another agent or that they can be removed.

To delete an OAM agent registration

1. From the System Configuration tab, navigation tree, expand the:

   Agents
   OAM Agents
   11g Webgates (or 10g Webgates)

2. Optional: Double-click the desired agent's name to view the registration, then close the page.
3. Click the desired agent's name, click the Delete button in the tool bar, and confirm the removal in the Confirmation window.
4. Confirm the Agent name is no longer listed in the navigation tree.
5. Remove WebGate Instance: Perform the following steps and also refer to "Removing a 10g WebGate from the OAM 11g Deployment" on page 17-25, if needed.

   a. Shut down the Web server.

   b. Remove WebGate software using the utility provided in the following directory path:

      $*WebGate_install_dir*/oui/bin

      ```
      Windows: setup.exe -d
      Unix: runInstaller -d
      ```

   c. Revert to the httpd.conf version before updates for WebGate. For example:

      Copy: httpd.conf.ORIG

      To: httpd.conf

   d. Restart the Web server.

   e. On the agent host, manually remove the WebGate instance directory. For example:

      11g WebGate: *11gWebGate_instance_dir*/webgate/config/ObAccessClient.xml *WebTier_Middleware_Home*/Oracle_WT1/instances1/config/OHS/ohs1/webgate/

      10g WebGate: $*WebGate_install_dir*/oblix/lib/ObAccessClient.xml

5.4 Registering and Managing OSSO Agents Using the Administration Console

This section describes how to manage OSSO Agent registrations (mod_osso) using the Administration Console. For details, see:

- About OSSO Agents and the OSSO Proxy
- About the Create OSSO Agent Page
- Searching for an OSSO Agent (mod_osso) Registration
- Registering an OSSO Agent (mod_osso)
- Viewing or Editing OSSO Agent (mod_osso) Registration
- Deleting an OSSO Agent (mod_osso) Registration

5.4.1 About OSSO Agents and the OSSO Proxy

An OSSO Agent is any mod_osso module deployed on an Oracle HTTP Server that is acting as a partner application for the OSSO server and protecting resources.

The OSSO Proxy supports interoperability between OAM and OSSO agents (using an OSSO agent to access a valid SSO session created for an OAM Agent, and vice versa).

OSSO Proxy Supports	Description
SSO login	From an OSSO Agent to the OAM Server (and OSSO-specific tokens)
SSO logout	From the OSSO Agent to the OAM Server
OSSO Agent requests and protocols	OSSO Proxy translates the OSSO protocol into a protocol for Oracle Access Manager 11g services.

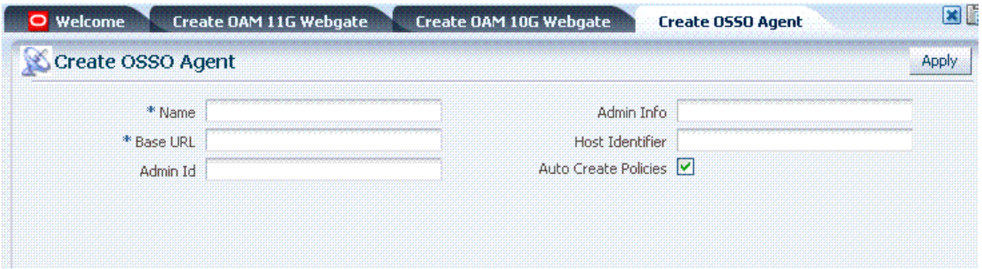
5.4.2 About the Create OSSO Agent Page

This topic describes OSSO Agent registration using the Administration Console.

Note: Before you register an OSSO Agent, ensure that the Oracle HTTP Server is installed on the client computer and that the Web server is configured for mod_osso.

Figure 5–5 shows a Create OSSO Agent page, under the System Configuration tab in the Administration Console.

Figure 5–5 Create OSSO Agent Page



On the Create OSSO Agent page, required information is identified by the asterisk (*). Table 5–7 describes the required and optional details that you can specify when you register a new agent.

Table 5–7 Create OSSO Agent Page Elements

Element	Description
Agent Name	The identifying name for this mod_osso Agent.
Agent Base URL Required for OSSO agents.	The required protocol, host, and port of the computer on which the Web server for the agent is installed. For example, http://*host:port* or https://*host:port*. Note: The host and port are used as defaults for the expanded registration. See Table 5–8.
Admin ID	Optional administrator log in ID for this mod_osso instance. For example, *SiteAdmin*.
Admin Info	Optional administrator details for this mod_osso instance. For example, *Application Administrator*.
Host Identifier	The host identifier is filled in automatically based on the Agent name

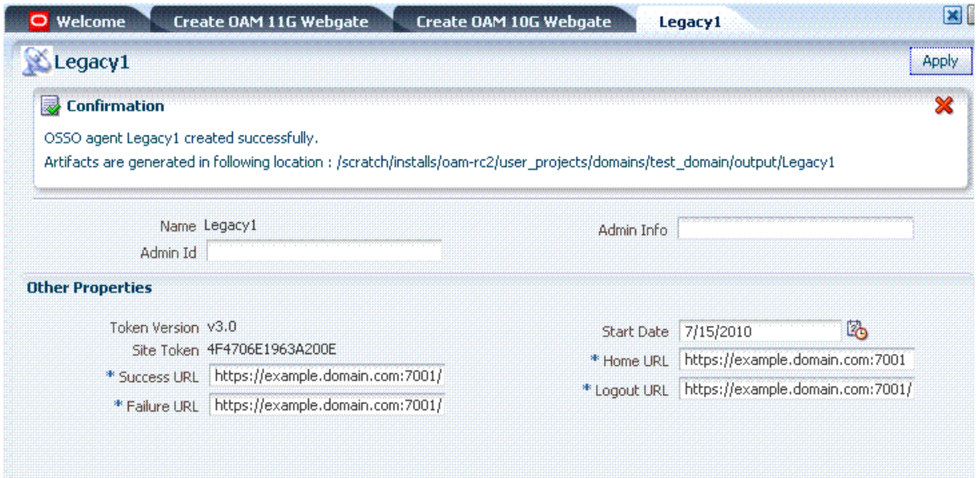
Table 5–7 (Cont.) Create OSSO Agent Page Elements

Element	Description
Auto Create Policies	During agent registration, you can have authentication and authorization policies created automatically. This option is checked (enabled) by default. The OSSO Proxy requires an application domain that includes a resource with the generic url(/.../*) protected by a policy based on the LDAP scheme (default). This is why a generic URL is used at the server side. Default: Enabled **Note**: If you already have a domain and policies registered, you can simply add new resources to it. If you clear (uncheck) this option, no application domain or policies are generated automatically.

To help streamline Agent registration, several elements are concealed and default values are used during registration with the console. When you view an agent's registration page in the Administration Console, all elements and values are revealed.

Figure 5–6 shows the full Agent page as viewed in the Administration Console. The Confirmation window is still visible. All details specified, and defaults, are shown.

Figure 5–6 OSSO Agent Page and Confirmation Window



OSSO agent Legacy1 created successfully.

Artifacts are generated in following location : /scratch/installs/oam-rc2/user_projects/domains/test_domain/output/Legacy1

Name Legacy1

Admin Id

Admin Info

Token Version v3.0

Site Token 4F4706E1963A200E

* Success URL https://example.domain.com:7001/

* Failure URL https://example.domain.com:7001/

Start Date 7/15/2010

* Home URL https://example.domain.com:7001

* Logout URL https://example.domain.com:7001/

Table 5–8 summarizes the expanded elements and defaults that are used by the OSSO Agent.

Table 5–8 Expanded OSSO Agent Elements

Element	Description
Token Version	The version of the token is 3.0. This cannot be edited.
Site Token	The Application Token used by the partner when requesting authentication. This cannot be edited.
Success URL	The redirect URL to be used upon successful authentication. By default, osso_login_success on the fully qualified host and port specified with the Agent Base URL are used. For example: Default: http://*myhost:5678*/osso_login_success
Failure URL	The redirect URL to be used if authentication fails.By default, osso_login_failure on the fully qualified host and port specified with the Agent Base URL are used: Default: http://*myhost:5678*/osso_login_failure
Start Date	First month, day, and year for which log in to the application is allowed by the server. Default: The date the Agent was registered.

Table 5–8 (Cont.) Expanded OSSO Agent Elements

Element	Description
Home URL	The redirect URL to be used for the Home page after authentication. By default, the fully qualified host and port specified with the Agent Base URL are used: Default: http://*myhost:5678*
Logout URL	The redirect URL to be used when logging out. This redirects the user to the global logout page on the server: osso_logout_success. By default, the fully qualified host and port specified with the Agent Base URL are used: Default: http://*myhost:5678*/osso_logout_success See Also: "Introduction to OAM 11g Centralized Logout" on page 11-2.

5.4.3 Searching for an OSSO Agent (mod_osso) Registration

Users with valid OAM Administrator credentials can perform the following procedure to search for an OSSO Agent using the Administration Console.

Prerequisites

The OSSO Agent must be registered and available in the Oracle Access Manager Administration Console.

To search for an OSSO Agent registration

1. Activate the System Configuration tab.
2. From the search type list, choose the OSSO agents type to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   ```
   my_OSSO_Agent
   ```

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit command button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.
6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

5.4.4 Registering an OSSO Agent (mod_osso)

Users with valid OAM Administrator credentials can perform the following procedure to register an OSSO Agent using the Administration Console.

Prerequisites

Ensure that the Oracle HTTP Server is installed and running on the client computer, and is configured for mod_osso.

See Also:

- About the Create OSSO Agent Page
- Chapter 6 for details about using the remote command-line tool

To register an OSSO Agent

1. Log in to the OAM Administration Console as usual.
2. From the Welcome page, Agent Configuration panel, click the following link to open a fresh page:

   - Add OSSO Agent

   Alternatively: From the System Configuration tab, expand the Agents node and the OSSO Agents node, then click the Create button in the tool bar.
3. Click the Create button in the tool bar.
4. On the Create: OSSO Agent page, enter required details, as shown in Table 5–7:

   - Agent Name
   - Agent Base URL
5. On the Create: OSSO Agent page, enter optional details as desired (Table 5–7):
6. Click Apply to submit the registration (or close the page without applying changes).
7. In the Confirmation window, check the path to generated artifacts and then close the window. For example:

   ```
   Artifacts are generated in following location : /.../base_domain/output/OSSO1
   ```
8. Copy the updated osso.conf file to the OSSO Agent host:

   a. On the OAM Administration Console host, locate the updated OSSO Agent osso.conf file. For example:

      $DOMAIN_HOME/output/$*Agent_Name*/osso.conf

   b. On the OSSO Agent host, copy osso.conf to the mod_osso directory path.

      $*OHS_dir*/osso.conf

      for instance, WebTier_Middleware_Home/Oracle_WT1/instances1/config/OHS/ohs1/config/osso.conf

   c. Restart the OAM Server that is hosting the OSSO Agent.
9. Proceed to Part III, "Single Sign-on, Policies, and Testing".

5.4.5 Viewing or Editing OSSO Agent (mod_osso) Registration

Users with valid OAM Administrator credentials can change any setting for a registered OSSO Agent using the Administration Console, as described in the following procedure. For example, you might want to revise the end date or add administrator information.

Prerequisites

Ensure that the Oracle HTTP Server is installed and running on the client computer, and is configured for mod_osso.

See Also:

- Searching for an OSSO Agent (mod_osso) Registration
- About the Create OSSO Agent Page

To view or modify an OSSO Agent registration

1. From the System Configuration tab, navigation tree, expand the Agents node.
2. Expand the OSSO Agents node, and double-click the desired agent's name to display the registration page.
3. On the registration page, view or modify details as needed based on details in Table 5–7 and Table 5–8.
4. Click Apply to submit the changes (or close the page without applying changes), and close the Confirmation window.
5. Copy the updated osso.conf file to the OSSO Agent host:
   a. On the OAM Administration Console host, locate the updated OSSO Agent osso.conf file. For example:

      $DOMAIN_HOME/output/$*Agent_Name*/osso.conf

   b. On the OSSO Agent host, copy osso.conf to the mod_osso directory path.

      $*OHS_dir*/osso.conf

      for instance, WebTier_Middleware_Home/Oracle_WT1/instances1/config/OHS/ohs1/config/osso.conf

   c. Restart the OAM Server that is hosting the OSSO Agent and proceed to Part III, "Single Sign-on, Policies, and Testing":

5.4.6 Deleting an OSSO Agent (mod_osso) Registration

Users with valid OAM Administrator credentials can perform the following procedure to delete a registered OSSO Agent from the Administration Console.

> **Note:** Deleting an agent registration removes only the registration (not the associated host identifier, application domain, resources, or the agent instance itself).

Prerequisites

Evaluate the application domain, resources, and policies associated with this agent and ensure that these are configured to use another agent or that they can be removed.

See Also: Searching for an OSSO Agent (mod_osso) Registration

To delete an OSSO Agent registration

1. From the System Configuration tab, navigation tree, expand the Agents node.
2. Expand the OSSO Agents node.
3. Optional: Double-click the desired agent's name to display the registration page; confirm this is the agent to remove, and close the page.
4. Click the Agent name, click the Delete button in the tool bar, and confirm removal in the Confirmation window.

6

Registering Partners (Agents and Applications) Remotely

Oracle Access Manager 11g provides a command-line utility to streamline partner registration. Any administrator inside the network can use the remote registration tool to specify WebGate parameters and values using a template. Administrators outside the network can use the utility to provide information to administrators within the network.

This chapter focuses on using the command-line utility to perform partner registration. This chapter includes the following topics:

- Prerequisites
- Introduction to Remote Partner Registration
- Acquiring and Setting Up the Registration Tool
- Creating the Registration Request
- Performing In-Band Remote Registration
- Performing Out-of-Band Remote Registration
- Validating Remote Registration and Resource Protection

6.1 Prerequisites

Before you can perform tasks in this chapter, ensure that an OAM Administration Console and a managed OAM Server are running.

6.2 Introduction to Remote Partner Registration

Supported policy enforcement agents must be registered with Oracle Access Manager 11g to communicate with OAM authentication and authorization services. A partner application (one that delegates the authentication function to the OAM SSO provider to spare users from re-authenticating when accessing multiple resources) must also be registered.

Protecting applications with Oracle Access Manager 11g requires an OAM Agent (WebGate) or OSSO Agent (mod_osso) that is registered with the OAM Administration Console, and an application domain that is configured to protect the application with specific authentication and authorization policies.

The following command-line registration functionality is supported:

- Secure registration and creation of an application domain by:

   - In-band Administrators (administrators within the network who manage the Web server that hosts the agent)

     In-band Administrators can use either the registration tool or the Oracle Access Manager Administration Console for registration tasks. This chapter focuses on command-line registration.

   - Out-of-band Administrators (those outside the network)

     Administrators outside the network must submit registration requests to an Administrator within the network. After processing the request, the in-band administrator returns the files required by the out-of-band Administrator who uses the files to configure his environment.

- Symmetric key generation per Partner Application

  One key is generated and used per registered mod_osso or 11g WebGate. However, one single key is generated for all 10g WebGates.

- Registration of earlier Oracle Access Manager WebGate and OSSO Agents for backward compatibility with legacy systems is provided. For more information, see the certification matrix on Oracle Technology Network:

  ```
  http://www.oracle.com/technology/software/products/ias/files/fusion_
  certification.html
  ```

Functionality in the following list is not supported:

- Persistence of the Key and Agent Information
- Generation of Keys used by internal Oracle Access Manager components
- API support for reading Agent information

For more information, see:

- About In-Band Remote Registration
- About Out-of-Band Remote Registration
- About Key Use, Generation, Provisioning, and Storage
- About the Remote Registration Tool
- About Remote Registration Requests
- About Out-of-Band Registration Responses

6.2.1 About In-Band Remote Registration

Following is a brief overview of in-band Web server administrator tasks for provisioning a partner application using the registration tool. The tasks are the same whether you have an OAM Agent (WebGate) or OSSO Agent (mod_osso) protecting resources.

> **Note:** mod_osso is an Oracle HTTP Server module that provides OracleAS applications with authentication. This module resides on the Oracle HTTP Server that enables applications protected by OracleAS Single Sign-On to accept HTTP headers in lieu of a user name and password once the user has logged into the OracleAS Single Sign-On server. The values for these headers are stored in a mod_osso cookie.

The mod_osso module replaces the single sign-on SDK that was used in earlier releases of OracleAS Single Sign-On to integrate partner applications. Located on the application server, mod_osso simplifies the authentication process by serving as the sole partner application to the single sign-on server. In this way, mod_osso renders authentication transparent to OracleAS applications. The administrator for these applications is spared the burden of integrating them with an SDK. After authenticating a user, mod_osso transmits the simple header values that applications may use to authorize the user:

- User name
- User GUID (global user identity)
- Language and territory

In this overview, the term "Administrator" refers to any user within the network who is part of the LDAP group that is designated for OAM administrators in the primary user identity store that is registered with OAM.

Task overview: In-band administrators performing remote registration

1. Acquire the Oracle Access Manager 11g Release 1 (11.1.1) registration tool as described in "Acquiring and Setting Up the Registration Tool" on page 6-19.
2. Update the input file with unique values for the agent and application domain as described in "Creating the Registration Request" on page 6-19.
3. Run the registration tool to configure the Agent and create a default application domain for the resources, as described in "Performing In-Band Remote Registration" on page 6-20.
4. Validate the configuration as described in "Validating Remote Registration and Resource Protection" in on page 6-22.
5. Perform access checks to validate that the configuration is working, as described in "Validating Authentication, Resource Protection, and Access After Remote Registration" on page 6-23.

6.2.2 About Out-of-Band Remote Registration

The term *out-of-band registration* refers to manual registration that involves coordination and actions by both the in-band Administrator and the out-of-band Administrator.

Task overview: Out-of-band remote registration (Agent outside the network)

1. Out-of-band Administrator creates a starting request input file containing specific application and agent details and submits it to the in-band Administrator.
   - Acquire the Oracle Access Manager 11g registration tool as described in "Acquiring and Setting Up the Registration Tool" on page 6-19.
   - Copy and edit a template to input unique values for the agent and application domain as described in "Creating the Registration Request" on page 6-19.
   - Submit the starting request input file to the in-band administrator using a method you choose (email or file transfer).
2. In-band Administrator:
   - Acquire the Oracle Access Manager 11g registration tool as described in "Acquiring and Setting Up the Registration Tool" on page 6-19.

- Use the out-of-band starting request with the registration tool to provision the agent and create the following files to return to the out-of-band Administrator. See "Performing In-Band Remote Registration" on page 6-20 for details of this and more on the following files:
  - *agentName*_Response.xml is generated for the out of band administrator to use in Step 3.
  - For WebGate Agents, a modified ObAccessClient.xml file is created (and the 11g WebGate cwallet.sso file), which the out-of-band administrator can use to bootstrap the WebGate.

    SSO wallet creation applies only to OAM 11g WebGates (not to OAM 10g agents or OSSO agents).
  - For OSSO Agents, a modified osso.conf file is created for the out-of-band administrator to bootstrap the OSSO module.

3. Out-of-band Administrator uses the registration tool with the *agentName*_Response.xml file and copies the Agent configuration and any other generated artifacts to the appropriate file system directory.
4. In-band Administrator validates the configuration as described in "Validating Remote Registration and Resource Protection" on page 6-22.
5. Out-of-band Administrator performs several access checks to validate that the configuration is working, as described in "Validating Authentication, Resource Protection, and Access After Remote Registration" on page 6-23.

6.2.3 About Key Use, Generation, Provisioning, and Storage

Each registered agent has a symmetric key, regardless of the registration method (Administration Console versus remote registration).

Each application will have a symmetric key whether it is protected through mod_osso, or an OAM Agent. This key is generated by the registration tool. Storage of the application mapping, key, and type of Agent persists in the system configuration for retrieval as needed.

Key Use

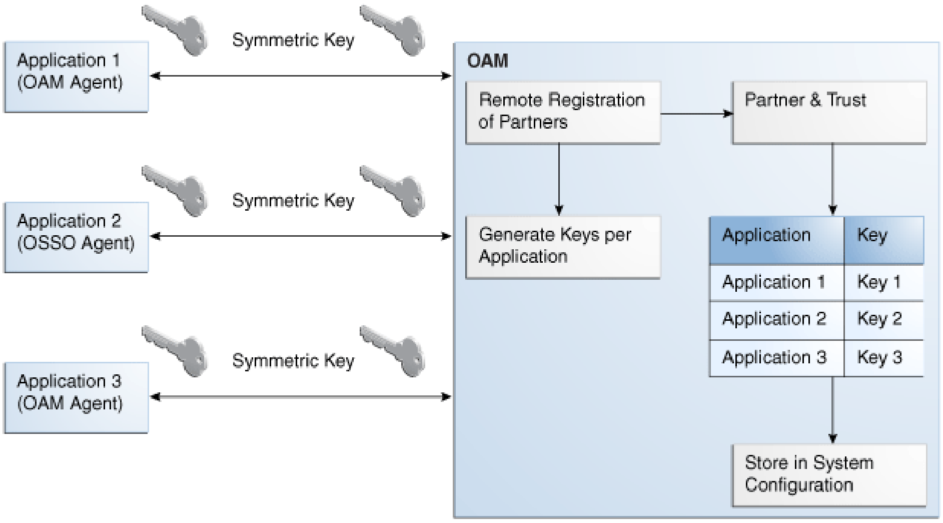
Each WebGate agent has its own secret key that is shared between the agent and the OAM 11g server. If one WebGate is compromised, other WebGates are unaffected. The following presents an overview:

- Encrypt/Decrypt the host-based WebGate-specific OAMAuthnCookie_<*host:port*>_<*random number*>.
- Encrypt/Decrypt the data that is redirected between WebGate and OAM 11g server.

Key Generation

Figure 6–1 illustrates the process of key generation, which occurs automatically when the agent is registered, regardless of the method used (Administration Console versus remote registration). There is one symmetric key per agent.

Figure 6–1 Key Generation


Application 1 (OAM Agent)
Symmetric Key
Application 2 (OSSO Agent)
Symmetric Key
Application 3 (OAM Agent)
Symmetric Key
OAM
Remote Registration of Partners
Partner & Trust
Generate Keys per Application
Application
Key
Application 1
Key 1
Application 2
Key 2
Application 3
Key 3
Store in System Configuration

Key Accessibility and Provisioning

Each Agent specific key must be accessible to the corresponding WebGate through a secure local storage on the client machine. Cryptographic keys are not stored in the data store. Instead, an alias to an entry in a Java KeyStore or CSF repository is stored; the partner and trust management API obtain the actual key when it is requested. The agent specific secret key:

- Is provisioned during remote registration (either in-band mode or out-of-band mode)
- Is unique so that it can uniquely identify each agent.
- Is distributed securely back to the agent (either through the wire during in-band mode or through a separate secure channel during out-of-band mode).
- Is saved in the Oracle Secret Store, in the SSO wallet. SSO wallet creation applies only to OAM 11g WebGates (not to OAM 10g agents or OSSO agents).

> **Note:** The Oracle Secret Store is a container that consolidates the storage of secret keys and other security-related secret information inside the Oracle Wallet, not in plain-text. The SSO wallet relies on underlying file system security to protect its data. Opening this wallet does not require a password. The SSO wallet depends on the operating system and file permissions for its security.

- Is saved in the Oracle Secret Store, in an auto-login editable SSO wallet, upon completion of Partner Registration.

Key Storage

The SSO wallet containing the agent key must be located in cwallet.sso, in the directory with ObAccessClient.xml in *WebGate_instance_dir*/webgate/config (for example, WebTier_Middleware_Home/Oracle_WT1/instances).

The SSO wallet does not require a user password, and should be protected with the proper file permission (700) or registry on Windows.

6.2.4 About the Remote Registration Tool

This topic provides an overview of the registration tool, requirements, usage, and results.

Location

<MiddlewareHome>/Oracle_IDM1/oam/server/rreg/

The registration tool, oamreg, is located in <*OAM_HOME*>/oam/server/rreg/client/RREG.tar.gz. After you untar the file on the computer hosting the Agent, the tool resides in the following path.

Platform	Path to oamreg
Linux	rreg/bin/oamreg.sh
Windows	rreg\bin\oamreg.bat

Requirements

Before using the script, two environment variables must be set within it as described here. The JDK home should point to JDKL 1.6:

Environment Variable	Description
OAM_REG_HOME	The directory under which RREG.tar was exploded, followed by /rreg.
JDK_HOME	The location where Java is located on the client computer. For example: *WLS_HOME*/Middleware/jdk160_11.

In addition, you must modify several tags in the Registration request. For details, see "About Remote Registration Requests" on page 6-8.

Registration Administrators

The user can be a part of any group that is mapped against the OAM Administrator's Role in the primary user-identity store. For more information, see Table 3–2, " Required User Identity Store Elements".

Remote Registration Modes and Request Files

The command to run the script requires two arguments:

- mode: `inband` or `outofband`
- input/*`file`*: Either the absolute path to the input file (*request.xml or an *agentName*_Response.xml), or the path relative to the value of OAM_REG_HOME. The preferred location is $*OAM_REG_HOME*/input.

Agent Types and Associated Request Files

Both `inband` and `outofband` modes use a request file with the input argument, as follows.

Table 6–1 Remote Registration Request Files

Agent Type	Request File
10g WebGates	$*OAM_REG_HOME*/input/OAMRequest_short.xml $*OAM_REG_HOME*/input/OAMRequest.xml
11g WebGates	$*OAM_REG_HOME*/input/OAM11GRequest.xml $*OAM_REG_HOME*/input/OAM11GRequest_short.xml
OSSO Agents (mod_osso)	$*OAM_REG_HOME*/input/OSSORequest.xml

See Also: "About Remote Registration Requests" on page 6-8

Generated Files

In `outofband` mode, the in-band administrator uses the starting request file submitted by the out-of-band administrator, and returns a generated *agentName_*Response.xml file to the out-of-band administrator for additional processing. The out-of-band administrator runs the remote registration tool with *agentName_*Response.xml as input to generate agent configuration files.

See Also: "About Out-of-Band Registration Responses" on page 6-18

Sample Remote Registration Commands and Results

Sample commands are shown in Table 6–2 and presume the location of the tool to be $*OAM_REG_HOME* on a Linux system.

Table 6–2 Remote Registration Sample Commands

Command Type	Sample (on Linux)
In-band Administrator using In-band Request	`./bin/oamreg.sh inband input/*Request.xml`
In-band Administrator using Submitted Request	`./bin/oamreg.sh outofband input/`*`starting_request.xml`*
Out-of-band Administrator using Returned Response	`./bin/oamreg.sh outofband input/agentName_Response.xml`

After launching the script, administrators are prompted for a username and password. After running the script, messages inform of success or failure. Based on the input file and the mode in which you are running the registration tool, you can expect the results described in Table 6–3.

Table 6–3 Results of Remote Registration

Server Side Results	Client Side Results
■ The oam-config.xml file contains an entry for the newly registered agent based on the <agentName> tag in the *Request.xml file. ■ The oam-policy.xml file on the server includes the following new entries: An application domain to protect resources created and named after the Agent based on the <agentName> tag in the *Request.xml file.	**inband Client-Side Results**: The Agent's native configuration file is generated and stored in a directory based on the <agentName> tag in the *Request.xml file (for example, RREG_Home/output/*agentName*/). The generated configuration file must replace the earlier agent configuration file. Either: ■ osso.conf, modified for the OSSO Agent ■ 11g WebGate: cwallet.sso ■ OAM Agents: ObAccessClient.xml, modified for the OAM WebGate The appropriate native configuration output file created during registration must be copied to the agent-installed location: 11g WebGate: Copy ObAccessClient.xml (and cwallet.sso) to *WebGate_instance_dir*/webgate/config (for example, WebTier_Middleware_Home/Oracle_WT1/instances/instance1/config/OHS/ohs1/webgate/config) 10g WebGate: Copy ObAccessClient.xml to *WebGate_install_dir*/webgate/config For mod_osso, copy osso.conf file to *OHS_webserver_install_dir*/oracle/product/11.1.1/as_1/instances/*instance1*/config/OHS/*ohs1*/osso/ ■ OAM Agents: Password.xml file and certificate files for Simple or Cert mode are also generated and must be copied.
	outofband Client-Side Results: Depending on the input file you use (starting request or a generated *agentName*_Response.xml file) the following results occur: ■ `input/starting_Request.xml`: Created by the out-of-band administrator and used by the in-band administrator to generate a response file (*agentName*_Response.xml) based on the <agentName> tag. The response file is sent to the out-of-band administrator using any method. ■ `input/`*agentName*_Response.xml: Sent to the out-of-band administrator and used to create the agent's native configuration file in a directory based on the <agentName> tag in the response file. For example: osso.conf, modified for the OSSO Agent cwallet.sso for 11g WebGate ObAccessClient.xml, modified for the OAM Agent (WebGate) The appropriate native configuration output file created during registration must be copied to the agent-installed location: 11g WebGate: Copy ObAccessClient.xml to *WebGate_instance_dir*/webgate/config (for example, WebTier_Middleware_Home/Oracle_WT1/instances/instance1/config/OHS/ohs1/webgate/config) 10g WebGate: Copy ObAccessClient.xml to *WebGate_install_dir*/webgate/config For mod_osso, copy osso.conf file to OHS_webserver_dir/oracle/product/11.1.1/as_1/instances/instance1/config/OHS/ohs1/osso/

See Also:

- "Performing In-Band Remote Registration" on page 6-20
- "Performing Out-of-Band Remote Registration" on page 6-21

6.2.5 About Remote Registration Requests

This topic describes the registration request files that are available for use with the registration tool, and the elements that are common between them:

- OSSO Remote Registration Request
- Short, Simplified OAM Remote Registration Requests

- Common Elements of Remote Registration Requests
- Full OAM Remote Registration Requests

6.2.5.1 OSSO Remote Registration Request

Example 6–1 provides an updated version of the OSSO Registration Request for use with the registration tool oamreg.sh (Linux) or oamreg.bat (Windows). The information highlighted in bold must be modified for a mod_osso agent. However, all other fields can use the default values.

> **See Also:** "Common Elements of Remote Registration Requests" on page 6-10

Example 6–1 OSSORequest.xml

```
...
<OSSORegRequest>
.
    <serverAddress>http://sample.us.oracle.com:7001</serverAddress>
    <hostIdentifier>RREG_Webdomain</hostIdentifier>
    <agentName>Remote_Reg_OSSO</agentName>
    <agentBaseUrl>http://sample.us.oracle.com:7777</agentBaseUrl>
    <oracleHomePath>$ORACLE_HOME</oracleHomePath>
        <virtualHost></virtualHost>
        <updateMode></updateMode>
        <adminInfo></adminInfo>
        <adminId></adminId>
        <protectedResourcesList>
              <resource>/access/resource1.html</resource>
              <resource>/access/resource2.html</resource>
         </protectedResourcesList>
         <publicResourcesList>
              <resource>/public/index.html</resource>
         /<publicResourcesList>

</OSSORegRequest>
```

6.2.5.2 Short, Simplified OAM Remote Registration Requests

Example 6–2 provides an updated sample of the short OAM registration request for use with the agent registration tool: oamreg.sh (Linux) or oamreg.bat (Windows). The only difference between the short OAM remote registration request for OAM 10g WebGates versus OAM 11g WebGates is the container:

- OAMRegRequest
- OAM11GRegRequest

> **Note:** While the short OAM remote registration request is nearly identical for both OAM 10g WebGates and OAM 11g WebGates, be sure to copy the appropriate template for your WebGate release.

Within Example 6–2, only the information highlighted in bold must be modified with values for your environment. All other fields in this file can use the default values. When you run oamreg, default values are provided automatically for all other Agent definitions which can be found in the full OAM remote registration requests.

Example 6–2 Sample Simplified Request: OAMRequest_short.xml

```
<OAMRegRequest>
.
    <serverAddress>http://sample.us.oracle.com:7001</serverAddress>
    <hostIdentifier>RREG_HostId11G</hostIdentifier>
    <agentName>Remote_Reg_OAM</agentName>
        <protectedResourcesList>
              <resource>/</resource>
              <resource>/.../*</resource>
         </protectedResourcesList>
         <publicResourcesList>
              <resource>/public/index.html</resource>
         </publicResourcesList>

</OAMRegRequest>
```

6.2.5.3 Common Elements of Remote Registration Requests

Unless otherwise stated, Table 6–4, explains the global elements within all request files.

Table 6–4 Elements Common to Remote Registration Requests

Element	Description	Example
<serverAddress>	Points to a running instance of the Oracle Access Manager Administration Console, including the host and port.	`<serverAddress>http://{oam_admin_server_host}:{oam_admin_server_port}</serverAddress>`
<agentName>	Defines a unique identifier for the agent on the OAM (Administration) Server. For every agent on the same server instance, this tag must be unique to avoid re-registering the same agent. Re-registering an agent on the same server instance is not supported without first deleting the existing agent.	`<agentName>RREG_OAM</agentName>`
<hostIdentifier>	This identifier represents the Web server host. The field is filled in automatically when you specify a value for the OAM Agent Name. If the agent name or host identifier of the same name already exists, an error occurs during registration. Note: If the <hostIdentifier> tag is specified, its value must be modified for your environment. To use a default value during registration, omit the <hostIdentifier> tag. If a host identifier of the same name already exists, the new agent Web server host:port, if specified, is added to the existing host identifier.	`<hostIdentifier>RREG_HostId11G</hostIdentifier>`

<protectedResourcesList>	Specifies the resource URLs that you want the OAM Agent to protect with some authentication scheme. The resource URLs should be relative paths to the agentBaseUrl.	`<protectedResourcesList>`
		`<resource>/</resource>`
		`<resource>/.../*</resource>`
		`</protectedResourcesList>`

<publicResourcesList>	Specifies the resource URLs that you want to keep public (not protected by the OAM Agent). The resource URLs should be relative paths to the agentBaseUrl. For instance, you might want to specify the Home page or the Welcome page of your application	`<publicResourcesList>`
		`<resource>/public/index.html</resource>`
		`</publicResourcesList>`

6.2.5.4 OSSO-Specific Elements in a Remote Registration Request

Table 6–5 describes the remote registration elements that are OSSO-specific.

Table 6–5 OSSO-Specific Elements in a Remote Registration Request

Element	Description	Example

<OracleHomePath>	The absolute file system directory path to the mod_osso agent.	`<oracleHomePath>`
		`$ORACLE_HOME`
		`</oracleHomePath>`

<virtualHost>	Default: None specified	`<virtualHost></virtualHost>`
<updateMode>	Default: None specified	`<updateMode></updateMode>`
<adminInfo>	Optional administrator details for this mod_osso instance. For example, *Application Administrator*. Default: None specified >	`<adminInfo></adminInfo>`
<adminInfo>	Optional administrator log in ID for this mod_osso instance. For example, *SiteAdmin*. Default: None specified >	`<adminId></adminId>`

6.2.5.5 Full OAM Remote Registration Requests

Table 6–6 provides information on individual elements in full OAM remote registration requests, which are in addition to those in the short version of the request (in Table 6–4):

- OAM11gRequest.xml (11g WebGates)
- OAMRequest.xml (10g WebGates)

> **Note:** Unless explicitly stated, each element appears in both 10g and 11g WebGate requests. Element names might differ slightly from their counterparts in the Console.

Table 6–6 Elements Common to Full Remote Registration Requests

Element	Description	Example
<agentBaseUrl>	Defines the Web server host:port on the computer that the agent is intended to protect. All URLs to protect are taken to be relative to this base URL, which should be specified as: http://*host:port*. Note: Each agentBaseUrl can be registered once only. There is a one-to-one mapping from the Agent's Base URL to the Web Server domain on which the WebGate is installed (as specified with the <hostIdentifier> element). However, there is a one-to-many mapping from the specified domain to the Agent's Base URL (one domain can have multiple Agent's Base URLs)	`<agentBaseUrl>http://{`*`web_server_host`*`):(`*`web_server_port`*`}</agentBaseUrl>`
<applicationDomain>	Defines the name of the application domain, which is based on the specified Agent Name (see Table 6–4).	`<applicationDomain>RREG_OAM11G</applicationDomain>`
<autoCreatePolicy>	During agent registration, you can have authentication and authorization policies created automatically. This option is checked (enabled) by default. Default: true (enabled) **Note**: If you already have a domain and policies registered, you can simply add new resources to it. If you clear (uncheck) this option, no application domain or policies are generated automatically.	`<autoCreatePolicy>true</autoCreatePolicy>`
<primaryCookieDomain> 10g Request Only In OAMRequest.xml (for 10g WebGates) <hostIdentifier> is also used as the preferred HTTP host.	Describes the Web server domain (client domain) on which the OAM 10g Agent is deployed, for instance,*.acompany.com*. You must configure the cookie domain to enable single sign-on among Web servers. Specifically, the Web servers for which you configure single sign-on must have the same Primary Cookie Domain value. The OAM Agent uses this parameter to create the ObSSOCookie authentication cookie. This parameter defines which Web servers participate within the cookie domain and have the ability to receive and update the ObSSOCookie. This cookie domain is not used to populate the ObSSOCookie; rather it defines which domain the ObSSOCookie is valid for, and which Web servers have the ability to accept and change the ObSSOCookie contents. Default: If the client side domain can be determined during registration, the Primary Cookie Domain is populated with that value. However, if no domain is found, there is no value and WebGate uses the host-based cookie. **Note**: The more general the domain name, the more inclusive your single sign-on implementation will be. For example, if you specify b.com as your primary cookie domain, users will be able to perform single sign-on for resources on b.com and on a.b.com. However, if you specify a.b.com as your primary cookie domain, users will have to re-authenticate when they request resources on b.com.	`<primaryCookieDomain>{client_domain}</primaryCookieDomain>`

Table 6–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example
<maxCacheElems>	Number of elements maintained in the cache. Cache elements are the following: ■ URLs—The URL cache maintains information about a URL, including if it is protected and the authentication scheme used if it is protected. ■ Authentication schemes—This cache stores authentication scheme information for a specific authentication scheme ID. The value of this setting refers to the maximum consolidated count for elements in both of these caches. Default = 100000	`<maxCacheElems>100000 </maxCacheElems>`
<cacheTimeout>	Amount of time cached information remains in the OAM Agent cache when the information is neither used nor referenced. Default = 1800 (seconds)	`<cacheTimeout>1800</cacheTimeout>`
<tokenValidityPeriod> 11g Request Only	Maximum valid time period for an agent token (the content of OAMAuthnCookie for 11g WebGate). Default = 3600 (seconds)	`<tokenValidityPeriod>3600 </tokenValidityPeriod>`
<cookieSessionTime> 10g Request Only	Maximum amount of time in seconds that a user's authentication session is valid, regardless of their activity. At the expiration of this session time, the user is re-challenged for authentication. This is a forced logout. Default = 3600 (seconds) A value of 0 disables this timeout setting.	`<cookieSessionTime>3600 </cookieSessionTime>`
<maxConnections>	The maximum number of connections that this OAM Agent can establish with the OAM Server. This number must be the same as (or greater than) the number of connections that are actually associated with this agent. Default = 1	`<maxConnections>1</maxConnections >`
<maxSessionTime>	Maximum duration, in hours, for a connection between WebGate and the OAM Server. Default = 24 (hours) A value of 0 disables this timeout setting.	`<maxSessionTime>24</maxSessionTim e>`
<ssoServerVersion>	SSO Token version values: ■ v3.0: Most secure token using AES encryption standard for encrypting tokens exchanged between OAM 11g server and mod_osso. This is the default value. This was supported by OSSO 10.1.4.3 patchset. ■ v1.4: This is supported by OSSO 10g prior to OSSO 10.1.4.3 patchset. Uses DES encryption standard. ■ v1.2: This used to be version of tokens exchanged between OSSO partners prior to OSSO 10.1.4.0.1. Uses DES.	<ssoServerVersion> >...</ssoServerVersion> >
<idleSessionTimeout> 10g Request Only	Amount of time in seconds that a user's authentication session remains valid without accessing any AccessGate protected resources. Default = 3600 A value of 0 disables this timeout setting.	`<idleSessionTimeout>3600> </idleSessionTimeout`

Table 6–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example
<failoverThreshold>	Number representing the point when this OAM Agent opens connections to a Secondary OAM Server. Default = 1 For example, if you type 30 in this field and the number of connections to primary OAM Server falls to 29, this OAM Agent opens connections to secondary OAM Server.	`<failoverThreshold>1 </failoverThreshold>`
<aaaTimeoutThreshold>-	Number (in seconds) to wait for a response from the OAM Server. If this parameter is set, it is used as an application TCP/IP timeout instead of the default TCP/IP timeout. Default = -1 (default network TCP/IP timeout is used) A typical value for this parameter is between 30 and 60 seconds. If set to a very low value, the socket connection can be closed before a reply from Access Server is received, resulting in an error. Both the WaitForFailover parameter and the aaaTimeoutThreshold must use the same value. See Also: Table 5–6 for more information on this parameter.	`<aaaTimeoutThreshold>-1 </aaaTimeoutThreshold>`
<sleepFor>	The frequency with which the Access Server checks its connections to the directory server. For example, if you set a value of 60 seconds, the Access Server checks its connections every 60 seconds from the time it comes up.	`<sleepFor>60</sleepFor>`
<debug>	Turns debugging on or off. Default: false (off)	`<debug>false</debug>`
<security>	Level of communication transport security between the Agent and the OAM Server (this must match the level specified for the OAM Server): ■ Open--No transport security ■ Simple--SSL v3/TLS v1.0 secure transport using dynamically generated session keys ■ Cert--SSL v3/TLS v1.0 secure transport using server side x.509 certificates Note: For more information, see Appendix E.	`<security>open</security`
<denyOnNotProtected>	Denies access to all resources to which access is not explicitly allowed by a rule or policy. Always enabled in 11g WebGate registration, and cannot be changed. On a 10g WebGate registration page, you can choose to disable this. When enabled, you must create an anonymous authentication method and allow access to content using an anonymous access policy.	`<denyOnNotProtected>1 </denyOnNotProtected>`

Table 6–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

<cachePragmaHeader>
<cacheControlHeader>
	These settings apply only to WebGates and control the browser's cache.
	By default, CachePragmaHeader and CacheControlHeader are set to no-cache. This prevents WebGate from caching data at the Web server application and the user's browser.
	However, this may prevent certain operations such as downloading PDF files or saving report files when the site is protected by a WebGate.
	You can set the Access Manager SDK caches that the WebGate uses to different levels. See http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html section 14.9 for details.
	All of the cache-response-directives are allowed. For example, you may need to set both cache values to public to allow PDF files to be downloaded.
		`<cachePragmaHeader>no-cache </cachePragmaHeader>`
		`<cacheControlHeader>no-cache </cacheControlHeader>`

<ipValidation>	IP address validation is specific to WebGates and is used to determine whether a client's IP address is the same as the IP address stored in the ObSSOCookie (10g WebGate) or OAMAuthnCookie (11g WebGate) generated for single sign-on.	`<ipValidation>0</ipValidation>`

<logOutUrls>	The Logout URL triggers the logout handler, which removes the cookie (ObSSOCookie for 10g WebGates; OAMAuthnCookie for 11g WebGates) and requires the user to re-authenticate the next time he accesses a resource protected by Oracle Access Manager.
	■ If there is a match, the WebGate logout handler is triggered.
	■ If Logout URL is not configured the request URL is checked for "logout." and, if found (except "logout.gif" and "logout.jpg"), also triggers the logout handler
	Note: This is the standard OAM 10g WebGate configuration parameter used to trigger initial logout.
	See Also: Chapter 11 for steps to configure logout for OAM 10g WebGates registered with OAM 11g.
		`<logOutUrls>`
		`<url>/logout1.html</url>`
		`<url>/logout2.html</url>`
		`</logOutUrls>`

<logoutCallbackUrl> 11g Request Only	The URL to oam_logout_success, which clears cookies during the call back. This can be a URI format without host:port (recommended), where the OAM Server calls back on the host:port of the original resource request. For example: /oam_logout_success This can also be a full URL format with a host:port, where OAM 11g server calls back directly without reconstructing callback URL. See Also: Chapter 11 for steps to configure logout for OAM 11g WebGates.	`<logoutCallbackUrl>/oam_logout_success </logoutCallbackUrl>`
<logoutTargetUrlParamName > 11g Request Only	The value is the name for the query parameter that the OPSS applications passes to WebGate during logout; the query parameter specifies the target URL of landing page after logout completes. Default: end_url Note: The end_url value is configured using param.logout.targeturl in jps-config.xml. See Also: Chapter 11 for steps to configure logout for OAM 11g WebGates.	`<logoutTargetUrlParamName>end_url </logoutTargetUrlParamName>`

Table 6–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

User-Defined Parameters	The following user-defined parameters are available for configuration in the remote registration request files only.
	Note: Each parameter can have only one value. User-defined parameters cannot be set using the OAM Administration Console.
		`<userDefinedParameters>`
		`   <userDefinedParam>`
		`      <name>...</name>`
		`      <value>...</value>`
		`</userDefinedParam>`

MaxPostDataLength	Determines the length of POST data.
	Oracle recommends that you do not set the value to less than 100. By default, or if this parameter is set to a value beyond the specified range, POST data length is limited to the default size of 0.75MB.
	Default: 750000
	See Also: Chapter 19 for more information about configuring IIS Web servers for WebGate.
		`<name>MaxPostDataLength</name>`
		`<value>750000</value>`

maxSessionTimeUnits	Allows the `MaxSessionTime` parameter to be interpreted as a number of minutes instead of the default (hours).
	Some firewalls forcefully disconnect OAM Server connections over a certain age or idle time. If you cannot modify firewall time-out settings, you can use `maxSessionTimeUnits`. The effect of lowering Maximum Client Session Time does increase the frequency with which access clients close and re-open connections to the OAM Server, which increases network traffic. Therefore, the `maxSessionTimeUnits` value should be as high as possible within the limits of the firewall settings.
	Default: hours
	Possible values: minutes
		`<name>maxSessionTimeUnits</name>`
		`<value>hours</value>`

RetainDownstreamPostData	Adding this user-defined parameter and setting the value to `true` resolves a problem that occurs when WebGate for Apache 2.0 or Apache 2.2 prevents POST data from being read by downstream applications. Form-based authentication schemes using the "passthrough" challenge-parameter and policies using the "Query String Variable(s)" option are affected.
	Default: false
		`<name>RetainDownstreamPostData`
		`</name>`
		`<value>false</value>`

useIISBuiltinAuthentication	Set to true only if you are using Microsoft Passport or Integrated Windows Authentication on the OAM Server on which the Agent is installed. It is used only for IIS, and is ignored if the WebGate is installed for another type of Web server.
	Default: false
		`<name>useIISBuiltinAuthentication`
		`</name>`
		`<value>false</value>`

idleSessionTimeoutLogic
10g WebGates only
	Release 7.0.4 WebGates enforced their own idle session timeout only.
	10.1.4.0.1 WebGates enforced the most restrictive timeout value among all WebGates the token had visited.
	With 10*g* (10.1.4.3), the 7.0.4 behavior was reinstated as the default with this element.
	To set idleSessionTimeoutLogic:
	■ The default value of `leastComponentIdleTimeout` instructs the WebGate to use the "most restrictive" timeout value for idle session timeout enforcement.
	■ A value of `currentComponentIdleTimeout` instructs the WebGates to use the "current WebGate" timeout value for idle session timeout enforcement.
		`name>idleSessionTimeoutLogic`
		`</name>`
		`<value>leastComponentIdleTimeout`
		`</value>`

Table 6–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

URLInUTF8Format	In an environment that uses Oracle HTTP Server 2, this parameter must be set to true to display latin-1 and other character sets.
	Default: true
		`<name>URLInUTF8Format</name>`
		`<value>true</value>`

inactiveReconfigPeriod
Shared secret applies to only 10g WebGate
Configuration applies to only 11g WebGate.
	In the idle state the WebGate reads the shared secret (configuration) from the OAM Server using the InactiveReconfigPeriod value. If this value is not set, the WebGate polls the OAM Server for the shared secret (configuration) value at an interval of 1 minute even though the updated shared secret (configuration) value will be returned only after 10 minutes.
	Default: 10 (minutes)
		`<name>inactiveReconfigPeriod</name>`
		`<value>10</value>`

WaitForFailover
10g WebGate only
	Used only for backward compatibility with NetPoint 5.x systems, this parameter has been replaced by aaaTimeoutThreshold.
	Both the WaitForFailover parameter, and the aaaTimeoutThreshold parameter, control the TCP/IP timeout between the WebGate and OAM Servers. The default value is "-1," which means the network default TCP/IP timeout value is used.
	Both the WaitForFailover parameter and the aaaTimeoutThreshold must use the same value.
		`<name>WaitForFailover</name>`
		`<value>-1</value>`

proxySSLHeaderVar	This parameter is used when the WebGate is located behind a reverse proxy, SSL is configured between the client and the reverse proxy, and non-SSL is configured between the reverse proxy and the Web server. It ensures that URLs are stored as https rather than http. The proxy ensures that URLs are stored in https format by setting a custom header variable indicating whether it is servicing an SSL or non-SSL client connection. The value of the ProxySSLHeaderVar parameter defines the name of the header variable the proxy must set. The value of the header variable must be "ssl" or "`nonssl`". If the header variable is not set, the SSL state is decided by the SSL state of the current Web server.
	Default: `IS_SSL`
		`<name>proxySSLHeaderVar</name>`
		`<value>IS_SSL</value>`

client_request_retry_attempts	WebGate-to-OAM Server timeout threshold specifies how long (in seconds) the WebGate waits for the OAM Server before it considers it unreachable and attempts the request on a new connection.
	This is the same for both 10g and 11g WebGates (OAM Agents) with OAM 11g.
	If the OAM Server takes longer to service a request than the value of the timeout threshold, the OAM Agent abandons the request and retries the request on a new connection.
	Note that the new connection that is returned from the connection pool can be to the same OAM Server, depending on your connection pool settings. The OAM Agent will first try the Primary OAM Server if one is available and then Secondary OAM Servers if one is available.
	Also, other OAM Servers might require more time to process the request than the time specified on the timeout threshold. In some cases, the OAM Agent can retry the request until the OAM Servers are shut down.
	You can configure a limit on the number of retries that the OAM Agent performs for a non-responsive server using the client_request_retry_attempts parameter.
	Setting the value to -1 (or not setting it at all) allows an infinite number of retries.
		`<name>client_request_retry_attempts`
		`</name>`
		`<value>1</value>`

Table 6–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

ContentLengthFor401Response	To set the Content-Length for all 401 responses, add the following as a user defined parameter and value: `ContentLengthFor401Response` 0.
	Zero (0) is the only value you can use. Any other value will be ignored. If you do not use this parameter and value, a mismatch between the content and content length might occur. This would result in either no data displayed in the browser or an error message in the browser.
		`<name>ContentLengthFor401Response </name>`
		`<value>0</value>`

SUN61HttpProtocolVersion	SUN v6.1 Web server might have a problem with redirection after reading POST data. If the connection uses the keepAlive (HTTP/1.1) protocol, data is not flushed properly. Thus, redirection might not work consistently.
	The SUN 6.1 Web server can be forced to use the HTTP/1.0 protocol when you assign a value of `1.0`.
	Default: 1.0
	Any value other than 1.0 will be ignored.
		`<name>SUN61HttpProtocolVersion </name>`
		`<value>1.0</value>`

impersonationCredentials	Default: cred	`<name>impersonationCredentials< /name>`
		`<value>cred</value>`

UseWebGateExtForPassthrough	This IIS Web server-specific parameter for use only with IIS v6 and v7 in Worker Process Isolation Mode.
	Default: false
	Set the value to true for IIS version 6.x (running in worker process isolation mode) and IIS 7.x in the following situations:
	■ To achieve Pass-through functionality
	■ For form login (if form login action is other than /access/oblix/apps/webgate/bin/webgate.dll)
	Note: You must also configure webgate.dll as an ISAPI extension (besides configuring it as an ISAPI filter). See Chapter 19, "Configuring the IIS Web Server for 10g WebGates".
	Also: For IIS 5.0 or IIS6.0 running in IIS 5.0 Isolation Mode this parameter should not be defined (or should be set to false). In this case, postgate.dll must be configured as an ISAPI filter to achieve pass-through functionality. For more information, see "Enabling Pass-Through Functionality for POST Data" on page 19-9.
		`<name>UseWebGateExtForPassthrough </name>`
		`<value>false</value>`

syncOperationMode	Default: false	`<name>syncOperationMode</name>`
		`<value>false</value>`

filterOAMAuthnCookie
11g Request only.
	When set to true, the OAMAuthnCookie is always filtered and not accessible to downstream applications.
	Default: true
		`<name>filterOAMAuthnCookie</name>`
		`<value>true</value>`

6.2.6 About Out-of-Band Registration Responses

After performing requested operations, in-band administrators send the following files to out-of-band Administrators for additional processing:

- *agentName*_Response.xml, which must be used as is by the out-of-band administrator.

This is not shown because Oracle recommends that you do not open or edit an *agentName*_Response.xml.

- Native Web server configuration files, which must be used by the out-of-band administrator to update their Web server.

6.3 Acquiring and Setting Up the Registration Tool

You can use the following procedure to acquire and update the oamreg script for your operating system:

Windows: oamreg.bat

Linux: oamreg.sh

and to update the appropriate *Request*.xml file that provides input for the specific agent you want to register.

For remote registration, two variables are required: JAVA_HOME and OAM_REG_HOME, as described in Table 6–7.

Table 6–7 Variables Required for Remote Registration

Location	Variable	Description
Client Side	JDK_HOME	The Java 1.6 location on the computer that relies on $JAVA_HOME already set in the environment.
	OAM_REG_HOME	The exploded directory for RREG.tar/rreg, which must be explicitly set in the script by the user.
rreg folder location as opposed to the RREG.tar.gz	JDK_HOME	Relies on $JAVA_HOME already set in the environment.
	OAM_REG_HOME	Is already set in the script during the installation.

See Also: "About the Remote Registration Tool" on page 6-6

To acquire the tool and update the script for your environment

1. Locate RREG.tar.gz file in the following path:

```
WLS_home/Middleware/domain_home/oam/server/rreg/client/RREG.tar.gz
```

2. Untar RREG.tar.gz file.
3. In the oamreg script, in rreg/bin/oamreg, set the JAVA_HOME environmental variable to JDK 1.6 (Table 6–7).
4. In the oamreg script, set the OAM_REG_HOME environmental variable to the *exploded_dir_for_RREG*.tar/rreg variables based on your environment (client side or server side Table 6–7).
5. Proceed with "Creating the Registration Request".

6.4 Creating the Registration Request

You can use the following procedure to create an appropriate *Request*.xml file to provide input for the specific agent you want to register.

See Also:

- "About the Remote Registration Tool" on page 6-6
- "About Remote Registration Requests" on page 6-8

To create the registration request

1. Locate the required *Request*.xml input file for the agent you want to register (Table 6–1).

   `WLS_home/Middleware/`*`domain_home`*`/oam/`*`server`*`/rreg/`input

2. Copy the request file to a new name. For example:

   From: OAMRequest.xml
   To: *myagent_request*.xml

3. In the Request file, modify information to reflect details for this agent and the resources to protect (Table 6–4 and Table 6–6):
4. Proceed with:
   - Performing In-Band Remote Registration
   - Performing Out-of-Band Remote Registration

6.5 Performing In-Band Remote Registration

This section provides steps you can use to perform in-band remote registration.

Prerequisites:

- Acquiring and Setting Up the Registration Tool
- Creating the Registration Request

You can use the following procedure to perform remote registration within the network.

> **Note:** In this situation, the Administrator within the network performs all tasks. The tasks are the same whether you have an OAM Agent (WebGate) or OSSO Agent (mod_osso) protecting resources.

This example illustrates registering an OAM Agent using the short registration request on a Linux system. **Alternatively**, you can use the OAM Administration Console to register the Agent and add an Application domain, as described in Chapter 5 and Chapter 9, respectively.

To perform in-band remote registration

1. On the computer hosting the Agent, run the registration command and specify your own *Request*.xml as the input file. For example:

   ```
   ./bin/oamreg.sh inband input/myagent_request.xml
   ```

2. Provide the registration Administrator user name and password when asked.
3. Read the messages on-screen to confirm:
   - Success: On-screen message confirms

     In-band registration process completed successfully!

Native Configuration File Location: "... created in output folder ..."

The output folder is in the same location where RREG.tar.gz was expanded: /rreg/output/*AgentName*/

4. Review the native configuration file, ObAccessClient.xml, created for the Agent in the output folder, and replace the earlier agent configuration file if it is not already replaced.
5. Finalize Agent Registration: Perform the following steps to finalize this agent registration:

   > **See Also:** Chapter 17, "Managing OAM 10g WebGates with OAM 11g".

   a. Copy ObAccessClient.xml to the OAM Agent host computer to manually update the WebGate configuration.

      10g WebGate: $*WG_install_dir*/oblix/lib/ObAccessClient.xml

      11g WebGate: $*WebGate_instance_dir*/webgate/config (also cwallet.sso) For example:

      $WebTier_Middleware_Home/Oracle_WT1/instances/instance1/config/OHS/ohs1/webgate/config

   b. Restart the Managed Server that is hosting the OAM Agent.
6. Proceed with "Validating Remote Registration and Resource Protection" on page 6-22.

6.6 Performing Out-of-Band Remote Registration

This section provides steps for administrators outside the network and those inside the network as they work together to register the remote Agent and set up a default application domain to protect resources.

Prerequisites:

- Acquiring and Setting Up the Registration Tool
- Creating the Registration Request

> **Note:** In this situation, the in-band Administrator and the out-of-band Administrator perform different tasks. Tasks are the same regardless of agent type: OAM Agent or OSSO Agent (mod_osso).

In the following procedure, steps illustrate the procedure to register an OAM Agent on a Linux system only:

- In-Band refers to a task performed by the Web server administrator within the network.
- Out-of-Band refers to a task performed the Web server administrator who is outside the network

To perform out-of-band remote registration

1. Out-of-Band Administrator: Create and send your *starting_request*.xml file to the in-band Administrator for processing (see "Creating the Registration Request"):

```
WLS_Home/Middleware/Oracle_
IDM1/oam/server/rreg/client/rreg/output/agentName/starting_request.xml
```

2. In-Band Administrator:

   a. Run the registration command and specify the out-of-band Administrator's *starting_request*.xml as the input file. For example:

      ```
      ./bin/oamreg.sh outofband input/starting_request.xml
      ```

   b. Provide the Registration Administrator user name and password when asked.

   c. Read messages on-screen to confirm:

      Success: "... registration process completed successfully!

      Response.xml location: "... created in input folder ..."

      The input folder is in the same location where RREG.tar.gz was expanded: /rreg/input/*AgentName*/

   d. Return the *agentName*_Response.xml file to the out-of-band Administrator along with any other artifacts. For example:

      *agentName*_Response.xml

3. Out-of-Band Administrator: Updates the environment, as follows.

   a. On the computer hosting the Agent, run the remote registration command and specify the received *agentName*_Response.xml as the input file. For example:

      ```
      ./bin/oamreg.sh outofband input/agentName_Response.xml
      ```

      ObAccessClient.xml and cwallet.sso (for 11g agents) are generated in the output folder location /rreg/output/AgentName/.

   b. Copy ObAccessClient.xml to the OAM Agent host computer to manually update the WebGate configuration.

      10g WebGate: $*WG_install_dir*/oblix/lib/ObAccessClient.xml

      11g WebGate: $*WebGate_instance_dir*/webgate/config (also cwallet.sso). For example:

      $WebTier_Middleware_Home/Oracle_WT1/instances/instance1/config/OHS/ohs1/webgate/config

   c. Restart the Web Server that is hosting the OAM Agent.

   d. Proceed with "Validating Remote Registration and Resource Protection" on page 6-22.

6.7 Validating Remote Registration and Resource Protection

This section provides the following topics:

- Validating Remote Registration
- Validating Authentication, Resource Protection, and Access After Remote Registration

6.7.1 Validating Remote Registration

You can use the following steps as a guide to validate the registration of an Agent and application regardless of the Agent type.

You must be an in-band Administrator to perform tasks using the OAM Administration Console. Out-of-band Administrators must test authentication and access remotely.

See Also:

- Chapter 5, "Registering Partners (Agents and Applications) by Using the Console"
- Chapter 9, "Managing Policies to Protect Resources and Enable SSO"

To validate agent and application registration

1. Agent Registration: Confirm Agent details under the System Configuration tab in the OAM Administration Console, as described in Chapter 5 and:
   - OAM Agent: Ensure that the modified ObAccessClient.xml resides in the WebGate installation directory to bootstrap communication between WebGate and the OAM Server.

     WebGate_install_dir\access\
   - OSSO Agent: Ensure that the modified osso.conf resides in the same directory as the Agent's Web server. Use it to bootstrap communication between OSSO Agent and OAM Server.
2. Shared Components, Host identifier: Confirm that the host identifier is defined in the OAM Administration Console.
3. Application Domain: Under the Policy Configuration tab, confirm there is a new default application domain, which is named after the registered Agent.
4. Resources in the application domain are associated with the host identifier.
5. Proceed with "Validating Authentication, Resource Protection, and Access After Remote Registration".

6.7.2 Validating Authentication, Resource Protection, and Access After Remote Registration

After registration, protected resource should be accessible with proper authentication without restart of admin server or Managed Server.

The procedure here provides several methods for confirming that registration, authentication, and authorization are properly configured and operational. The procedures is nearly identical for both OAM Agents (WebGates) and OSSO Agents (mod_osso).

Oracle recommends that the out-of-band administrator perform these verifications.

To verify authentication and access after registration

1. Enter the URL for an application protected by the registered OAM Agent to confirm that the log in page appears (proving that the authentication redirect URL was specified appropriately). For example:

   ```
   http://myWebserverHost.us.abc.com:8100/resource1.html
   ```
2. On the Log In page, enter a valid username and password when asked, and click Login.
3. Check the OAM specific cookies are created in the browser session. For example:

ObSSOCookie:

```
Set-Cookie:
ObSSOCookie=GGVEuvjmrMe%2FhbItbjT24CBmJo1eCIfDIwQ1atdGdnY4mt6kmdSekSFeAAFvFrZZZ
xDfvpkfS3ZLZFbaZU2rAn0YYUM3JUWVYkYFwB%2BBK7V4x%2FeuYHj%2B8gwOyxhNYFna3iSx1MSZBE
y51KTBfsDYOiw6R%2BCxUhOO8uZDTYHI3s0c7AQSyrEiQTuUV3nv1omaFZlk1GuZa4J7ycaGbIUyqwX
rM0cKuBJNd6sX1LiRj9HofYQsvUV7ToqeAOpDS7z9qs5LhqU5Vq60bBn12DTX6zNX6Lcc0L5tVwvh7%
2BnOAkz2%2BoDkLs%2BBTkeGcB3ppgC9;httponly; path=/; domain=.us.oracle.com;
```

OAM_ID Cookies:

```
Set-Cookie:
OAM_ID=v1.0~0~E1EBBC9846E09857060A68E79AEEB608~AA79FC43C695162B6CDE3738F40E94DA
6408D58B879AC3B467EBBD4800743C899843672B3511141FFABCF58B2CDCB700C83CC734A913625
7C4ABDA6913C9EF5A4E05C5D03D3514F2FECACD02F1C1B9314D76B4A68CB7A8BE42AEB09AFB98B8
EB; path=/; HttpOnly
```

4. Proceed as follows:

   - **Success**: If you authenticated successfully and were granted access to the resource; the configuration is working properly. Proceed with Steps 5 through 12 for further validations.
   - **Failure**: If you received an error during login or were denied access to the resource, check the following:
     - **Login Error**: Confirm that you provided a valid user id and password.
     - **Unavailable Resource**: Confirm that the resource is available.
     - **Wrong Redirect URL**: Verify the redirect URL in the OAM Administration Console.

5. **User Variations:** Perform steps 1 through 4 again with user variations to confirm appropriate behavior (either success for authorized users or failure for unauthorized users).

6. **Request Cancellation:** Perform a partial log in and click Cancel to confirm that the resource is not accessed.

7. **Modified Authentication URL:** Enter a nearly identical authentication URL as you perform Steps 1 through 5 to confirm appropriate response. For example, add a character to the URL string.

8. **Updated Resource:** Perform the following steps to ensure the resource is accessible. For example:

   Original Resource: /abc/test.html

   Updated Resource: /abc/xyz/test.html

   Without restarting the Oracle WebLogic Server:

   - Access the updated resource and confirm the user is asked to authenticate and the resource is accessible.
   - Access the original resource and confirm that the resource is accessible and the user is not asked for authentication.

9. **Various URL Patterns:** Verify authentication for various URL patterns as you perform steps 1 through 5.

10. **New Authentication Scheme:** Perform the following steps to confirm authentication operations without restarting the WebLogic Server.

    - Add a new authentication policy that uses a different Authentication Scheme.

- Protect the resource using the new policy.
- Without restarting the Oracle WebLogic Server, perform steps 1 through 4.

  See Also: Chapter 9, "Managing Policies to Protect Resources and Enable SSO"

11. **CGI Resource Header Variable and Cookies:** Perform the following steps to confirm authentication operations without having to restart the WebLogic Server.
    - Add a new authentication policy to protect a Common Gateway Interface (CGI) resource and set the Response for "Authentication Successful".
    - Protect the resource using the new policy.
    - Access the CGI resource.
    - Check for the header values configured for the response in a CGI data dump.
12. **Agent Disabled**: Perform the following steps to validate accessibility and authentication if WebGate is disabled in ObAccessClient.xml (WebGate should pick up the enabled value from oam-config.xml).
    - Disable the Agent (OAM Agent (WebGate) or OSSO Agent (mod_osso)).
    - Start the Web server and OAM Server.
    - Access an application protected by the OAM Agent and confirm that you are asked to authenticate.

Part III

Single Sign-on, Policies, and Testing

Part III provides information to help you understand single-sign on with OAM 11g as well as how to configure OAM policies and logout. Testing your single sign-on connection and policies is also described.

Part III contains the following chapters:

7

Introduction to the OAM Policy Model, Single Sign-On

Login is the action the user takes to authenticate and gain access to a desired application. Single sign-on (SSO) is enabled by Oracle Access Manager to eliminate the need for additional or different logins to access other applications during the same user session.

This chapter provides an introduction to Oracle Access Manager 11g single sign-on to lay some ground work before developing policies and the components that these require. This chapter includes the following topics:

> **Note:** Unless explicitly stated, information in this chapter is the same for OAM Agents and OSSO Agents.
>
> For details about single log-out, see Chapter 11, "Configuring Centralized Logout for OAM 11g".

7.1 Prerequisites

A fully functional Oracle Access Manager 11g system, including at least two registered Agents, is required.

This section identifies knowledge-based requirements for tasks in this chapter.

- Learn more about agent registration from Chapter 5, "Registering Partners (Agents and Applications) by Using the Console"

7.2 Comparing the OAM 11g Policy Model with OAM 10g

Oracle Access Manager 11g distills the policy models of both Oracle Access Manager and OSSO 10g into a single model that provides simplicity, flexibility, and a future growth path.

Table 7–1 compares the OAM 11g policy model with other models.

Table 7–1 Comparing OAM 11g Policy Model with OAM 10g

Policy Elements	OAM 11g	OAM 10g
Policy Authoring	OAM Administration Console	OAM Policy Manager
Policy Store	Database	LDAP directory server
Domain	Application Domain	Policy Domain
Resources	1. No resource prefixes. Resource definitions are treated as complete URLs. 2. Pattern matching (with limited features) for: ' * ' and '...' are supported 3. Resources need not be unique across domains. 4. No query-string or per-operation protection for HTTP URLs. 5. Each HTTP resource is defined as a URL path, and associated with a host identifier. However, resources of other types are associated with a specific name (not a host identifier. 6. Non-HTTP resource types are supported, without definition of specific operations. Non-HTTP resource types are never associated with a host identifier.	1. Resource prefixes are defined in domains 2. Pattern matching for: { } * … 3. Resources need not be unique across domains. 4. http resources can be protected based on URL query string contents and/or HTTP operation. This combination must be unique across domains. 5. Non-HTTP resource types and operations can be defined.
Policies	1. Authentication policies include resources, success responses, and an authentication scheme. 2. Authorization policies can also contain success responses, and time based, IP based and user-based constraints. 3. Only one authentication policy and one authorization policy can be associated with any resource. 4. Authentication and Authorization policies can evaluate to Success or Failure. 5. No Query Builder and no support for LDAP filters for (for retrieving matches based on an attribute of a certain display type, for example). 6. There is no notion of default policy in an application domain. However, you can define a policy for resource: /…/* which can be used as a default policy within a determined scope).	1. Authentication policies are simple and contain only authentication- scheme-based rule. 2. One resource can be associated with a set of Authorization policies. Evaluation of these policies can be based on an expression that combines the policies within the set using logical operators as desired. A resource can also be associated with multiple authentication policies and authorization policy sets. However, only one set applies. 3. An Authorization policy can evaluate to Success or Failure, or Inconclusive. 4. Users can be specified using LDAP filters. 5. Default authentication policy and authorization policy set can be defined for a policy domain. This policy is only applicable if there are no other applicable policies for a runtime resource in that domain.
Responses	1. Authentication and Authorization success Responses can be defined within the policies. These are applied after evaluation of policies. 2. Cookie, Header, and Session responses are supported. 3. URL redirection can be set. 4. Response definitions are part of each policy.	1. Authentication and Authorization Responses can be defined within the policies for Success, Failure, and Inconclusive events. These are returned to the caller after evaluation of policies. 2. HTTP_HEADER and Cookie based variables can be set. 3. Redirect URLs can be set for Success and Failure events of authentication and authorization policy evaluations.
Authentication Schemes	Authentication Schemes are defined globally and can be referenced within authentication policies.	Authentication Schemes can be defined outside of policies and can be referenced within authentication policies.

7.3 Introduction to the OAM 11g Policy Model

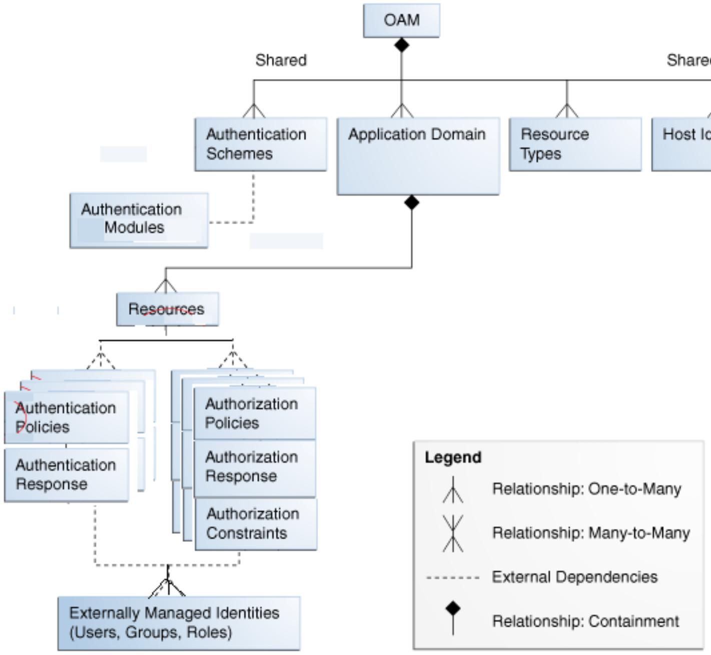
This section introduces the Oracle Access Manager 11g policy model and the global shared components within it.

The Oracle Access Manager 11g policy model supports both authentication and authorization services within the context of an OAM application domain. The policy model relies on external user identity stores and on authentication modules, which are a part of the overall system configuration.

> **Note:** Earlier releases of Oracle Access Manager provided authentication and authorization services within the context of an OAM policy domain. OracleAS SSO 10g provides only authentication.

Figure 7–1 illustrates the different elements within the Oracle Access Manager 11g policy model.

Figure 7–1 Oracle Access Manager 11g Policy Model and Shared Components



Application Domains and Policies

The top-level construct of the Oracle Access Manager 11g policy model is the OAM application domain. Each application domain provides a logical container for resources, and the associated authentication and authorization policies that dictate who can access these.

The size and number of application domains is up to the administrator; the decision can be based on individual application resources or any other logical grouping as needed. An application domain is automatically created during Agent registration. Also, administrators can protect multiple application domains using the same agent by manually creating the application domain and adding the resources and policies. For details, see:

- About Application Domains and Policies
- About Resources and Resource Definitions
- About Authentication Policies, Responses, and Resources
- About Authorization Policies, Resources, Constraints, and Responses

Shared Policy Components

Global policy components that can be used in one or more application domains. The following topics provide more information:

- About Resource Types
- About Host Identifiers
- About Authentication, Schemes, and Modules

7.3.1 About Resource Types

A resource type describes the kind of resource to be protected.

Each resource is defined using a single resource type. However, you can define any number of resources using that type.

Before you can add resources to an application domain for protection, *their* resource type must be defined. Administrators typically use the default resource type, HTTP, but non-HTTP types can be defined.

For more information about resource types and management, see "Managing Resource Types" on page 8-2.

7.3.2 About Host Identifiers

A host can be known by multiple names. To ensure that OAM recognizes the URL for a resource, OAM must know the various ways used to refer to that resource's host computer.

Table 7–2 illustrates the different host names under which a Web server might be accessible to employees. Creating a single Host Identifier using all of these names allows you to define a single set of policies to appropriately protect the application, regardless of how the user accesses it.

Table 7–2 Host Identifiers Examples

Host Identifier	Description
hrportal.intranet.company.com	A friendly name employees can remember. This is a load-balanced proxy, and requests to this could actually utilize one of several servers hosting the HR application.
hr-sf-02.intranet.company.com	A single machine hosting the application, which can be accessed directly.
hrportal.company.com	The same application is also accessible externally to the corporate firewall, primarily for use by ex-employees to check benefits, 401k info, and so on. This is also a load-balanced reverse proxy.

With OAM, all possible variations are stored together. Administrators enter the canonical name for the host and every other name by which the host can be addressed by users. A request sent to any address on the list is mapped to the official host name.

Host identifiers are created automatically during Agent registration and are used to seed the Resource definition and default authentication and authorization policies in the new application domain. **Alternatively**: an administrator can create a host identifier definition for use in one or more application domains.

Authentication and authorization policies in an application domain protect resources based on host identifiers. Host identifiers are used to identify resources or an application at run time and can be used to formulate policies for application resources at design time.

For more information, see "About Host Identifiers" on page 8-5.

See Also: The following chapters for more information about registering agents and applications:

- Chapter 5, "Registering Partners (Agents and Applications) by Using the Console"
- Chapter 6, "Registering Partners (Agents and Applications) Remotely"

7.3.3 About Authentication, Schemes, and Modules

Authentication is the process of proving that a user is who he or she claims to be. Authenticating a user's identity with Oracle Access Manager refers to running a pre-defined set of processes to verify the digital identity of the user.

Each authentication policy can be assigned only one authentication scheme. However, one authentication scheme can be assigned to multiple authentication policies.

One authentication policy can protect many resources. However, each resource can be protected by only one authentication policy.

See the following topics:

- Authentication Schemes and Modules
- Authentication Event Logging and Auditing

7.3.3.1 Authentication Schemes and Modules

Using OAM, a resource or group of resources can be protected by a single authentication process known as an authentication scheme. Authentication schemes rely on pre-defined authentication modules.

Authentication Scheme: A named component that defines the challenge mechanism, level of trust, and the underlying authentication module required to authenticate a user. It also contains some general information about itself. Authentication schemes are defined globally, to ensure that a small number of Security Administrators define them in a consistent, secure way. There are several default authentication schemes provided with Oracle Access Manager 11g.

Authentication Modules: The smallest executable unit of an authentication scheme. Several pre-defined modules are provided. Each module contains standard plug-ins. The authentication module determines the exact procedure to be followed and the method for challenging the user for credentials. For more information about these modules, see "Managing Authentication Modules" on page 8-12.

For more information, see "Managing Authentication Schemes" on page 8-18.

Multi-level Authentication: Oracle Access Manager 11g enables administrators to assign different authentication levels to different authentication schemes, and then choose which scheme protects which application. A highly sensitive application might require a user certificate and a less sensitive application might require a user name and password. For example, if a user is granted access to a resource that has a Basic Over LDAP authentication scheme defined as having a level of 2, the user can access other resources that have schemes with the same or a lower level. However, if the user tries to access a resource with a more stringent authentication challenge, such as a scheme called Client Certificate with a level of 5, they must re-authenticate.

Windows Native Authentication: Integrated Windows Native Authentication is supported for both OSSO and WebGate protected applications, as described in *Oracle Fusion Middleware Integration Guide for Oracle Access Manager*.

Other Authentication Types: Authentication features required by Oracle Fusion Middleware applications are supported, including:

- Weak authentication, typically a user name and password, no certificates
- Auto-login with third-party self-service user provisioning
- HTTP header support for user context information. For instance, host identifiers are used to create a host context for the resource. This is useful when adding resources that have the same URL paths on different computers.

If you use different authentication schemes for two WebGates, users can go from a higher authentication scheme to a lower one without re-authentication, but not from a lower level to a higher level.

> **Note:** During single sign-on, users might pass the authentication tests but might fail the authorization tests when attempting to access a second or third resource. Each resource in the domain might have a unique authorization policy.

For details about configuring and using authentication schemes with Oracle Access Manager11g, see "Managing Authentication Schemes" on page 8-18.

7.3.3.2 Authentication Event Logging and Auditing

Authentication Success and Failure events are audited, in addition to administration events. Auditing covers creating, modifying, viewing, and deleting authentication schemes, modules, and policies. Information that is collected about the user who is authenticating includes:

- IP address
- User Login ID
- Time of Access

During logging (or auditing), user information, user sensitive attributes are not recorded. Secure data (user passwords, for example) are removed to avoid misuse.

> **See Also:**
>
> - Chapter 13, "Logging Component Event Messages"
> - Chapter 14, "Auditing OAM Administrative and Run-time Events."

Several monitoring and diagnostic metrics are collected during authentication. For more information, see Chapter 15, "Monitoring OAM Metrics by Using Oracle Access Manager".

7.3.4 About Application Domains and Policies

OAM 11g default behavior is to deny access when a resource is not protected by a policy that explicitly allows access. In contrast, OAM 10g default behavior allowed access when a resource was not protected by a rule or policy that explicitly specified access.

The Oracle Access Manager 11g policy model enables you to control who can access resources when you define an application domain that discriminates between authenticated users who are authorized and those who are not authorized for access to a particular resource.

There are several ways that users can attempt to access a resource that is protected by an application domain, for example, by entering a URL in a browser, by running an application, or by calling some other external business logic. When users request access to resources protected by an application domain, their requests are evaluated according to the domain's policies (for authentication and authorization).

An application domain logically groups resources and policies in a flexible way. Each Application domain can be made to contain policy elements related to an entire application deployment, a particular tier of the deployment, or a single host. Application domains do not have any hierarchical relationship to one another.

Within the application domain, specific resources are identified as well as the policies that govern each resource. Authentication and authorization policies include administrator-configured responses that are applied on successful evaluation. Authorization policies include administrator-configured constraints that define how evaluation is performed.

Each application domain must have a unique name (a brief description is optional). Each domain is seeded with a resource container and policy containers where administrators can define resources and policies.

7.3.5 About Resources and Resource Definitions

Resources represent a document, or entity, or pieces of content stored on a server and available for access by a large audience. Clients communicate with the server and request the resource using a particular protocol (HTTP or HTTPS, for example) that is defined by an existing Resource Type.

Note: To protect pieces of content on a page, Oracle recommends using Oracle Entitlements Server.

With Oracle Access Manager, resource definitions are created within an application domain. Each resource is defined as a URL path. Every HTTP Resource Type must be associated with a host identifier. However, non-HTTP Resource Types (non-HTTP resources), are associated with a specific name (not a host identifier).

Note: Only resources defined within an application domain can be associated with policies defined for the application domain.

For more information, see "Adding and Managing Resource Definitions for Use in Policies" on page 9-11.

7.3.6 About Authentication Policies, Responses, and Resources

Authentication is the process of proving that a user is who he or she claims to be. To authenticate a user, Oracle Access Manager presents the user's browser with a request for authentication credentials in the form of a challenge. The challenge is referred to as a challenge method.

Administrators can create one or more authentication policies in an application domain to apply to specific resources within that domain.

> **Note:** Authentication is provided by OAM 11g Authentication Policies regardless of Agent type.

Authentication Policy Evaluation

Authentication policies specify the authentication methodology to be used for authenticating the user for whom the access must be provided on a given resource. Policies define the way in which the resource access is to be protected.

After a policy has been evaluated, two standard actions are performed:

- The result is returned
- The user is shown something based on that result: either the requested URL requested (on Success, allow) or the URL of a generic error page (on Failure, deny)

  Either or both results can be overridden on a policy-by-policy basis.

Policy Responses for SSO

Administrator-defined policy responses declare optional actions to be taken in addition to the above. Policy responses provide the ability to insert information into a session and pull it back out at any later point. This is more robust and flexible than OAM 10g, which provided data passage to (and between) applications by redirecting to URLs in a specific sequence. For details, see "Introduction to Policy Responses for SSO" on page 9-28.

> **Note:** Policy responses must be configured by an administrator and applied to specific resources defined within the same application domain.

For more information, see "Anatomy of an Application Domain and Policies" on page 9-3.

7.3.7 About Authorization Policies, Resources, Constraints, and Responses

Authorization is the process of determining if a user has a right to access a requested resource.

Administrators can create one or more authorization policies to specify the conditions under which a subject or identity has access to a resource. A user might want to see data or run an application program protected by a policy. The requested resource must belong to an application domain and be covered within that domain by a specific authorization policy.

> **Note:** OracleAS SSO 10g does not provide authorization; OSSO Agents do not use OAM 11g Authorization Policies.

Authorization Responses for SSO

Administrator-defined policy responses declare optional actions to be taken in addition to the above. Policy responses provide the ability to insert information into a session and pull it back out at any later point. This is more robust and flexible than OAM 10g, which provided data passage to (and between) applications by redirecting to URLs in a specific sequence.

For more information, see "Introduction to Policy Responses for SSO" on page 9-28.

Authorization Constraints

An authorization constraint is a rule that grants or denies access to a particular resource based on the context of the request for that resource. Authorization constraints determine if the authorization succeeds or fails for the request.

Administrators must define the constraints that apply to the resources assigned to the authorization policy. For details, see "Introduction to Authorization Constraints" on page 9-35.

Authorization Policy Evaluation

Evaluation of the authorization policy results in one of two outcomes: SUCCESS (allow) or FAILURE (deny). If the data is insufficient to evaluate the policy, the outcome is always FAILURE. For example, a constraint verifies that the user is a member of a group before allowing access; however, if the group does not actually exist in the LDAP, the outcome is FAILURE.

For more information, see "Defining Authorization Policies for Specific Resources" on page 9-24.

7.4 Introduction to Configuring OAM Single Sign-On

To begin the introduction to OAM 11g SSO, the following task overview summarizes how to configure single sign-on with OAM 11g, and where to find additional information related to specific tasks.

Task overview: Configuring single sign-on with OAM 11g

1. Review the following topics:
   - Comparing the OAM 11g Policy Model with OAM 10g
   - Introduction to the OAM 11g Policy Model
   - Introduction to SSO Components
   - Introduction to OAM 11g Single Sign-On Implementation Types
   - Introduction to OAM 11g SSO Processing
2. Configure a single sign-on logout URL for each partner application using documentation for your application.
3. Install an OAM policy-enforcement Agent on each Web server that is hosting an application that you want to protect:
   - Chapter 5, "Registering Partners (Agents and Applications) by Using the Console"
   - Chapter 6, "Registering Partners (Agents and Applications) Remotely"

Note: Registering an Agent with OAM 11g automatically creates a default host identifier and application domain seeded with basic policies.

4. Confirm that the desired resource type is available, (or create one yourself), as described in "Managing Resource Types" on page 8-2.
5. Confirm that you have the proper authentication modules and schemes, as described in:
   - "Managing Authentication Modules" on page 8-12
   - "Managing Authentication Schemes" on page 8-18
6. Confirm that a host identifier definition named for the agent was created automatically, (or create one yourself) as described in "Managing Host Identifiers" on page 8-5.
7. Add resource definitions and configure OAM 11g policies in the application domain as described in Chapter 9, "Managing Policies to Protect Resources and Enable SSO".
8. Configure the Server Common SSO Engine, as described in "Managing the Common SSO Engine" on page 9-45.
9. Validate the configuration, as described in "Validating Global Sign-On and Centralized Logout" on page 11-15.

7.5 Introduction to SSO Components

This section provides a brief overview of single sign-on with Oracle Access Manager 11g. It includes the following topics:

- About Single Sign-On Components
- About Single Sign-On Cookies
- About Single Sign-On Cookies

7.5.1 About Single Sign-On Components

This topic introduces key components to implementing and enforcing Oracle Access Manager 11g policies for single sign-on.

Table 7–3 summarizes key single sign-on components.

Table 7–3 OAM 11g SSO versus OSSO 10g Component Summary

Component Description	OAM 11g	OSSO 10g
Servers **Note**: Non-administrative users first gain access to the single sign-on server by entering the URL of a partner application, which returns the SSO login page. See Also: "Introduction to OAM 11g SSO Processing" on page 7-18.	■ OAM Server ■ OAM Administration Console (installed on the WebLogic Administration Server) **Note**: Administrative users access the administration console home page by typing the URL: https://*host:port*/oamconsole. See Also: "Logging In to and Signing Out of Oracle Access Manager 11g" on page 2-9.	■ OracleAS SSO server (OSSO server) See Also: *Oracle Application Server Single Sign-On Administrator's Guide*.
Proxy Provides support for legacy systems:	■ OAM Proxy supports legacy Oracle Access Manager implementations by acting as a legacy Access Server.	■ OSSO Proxy supports legacy SSO implementations by acting as the legacy OSSO Server.
Policy Enforcement Agents Resides with the relying parties and delegate authentication and authorization tasks to OAM Servers.	■ 11g OAM Agents (WebGates) ■ 10g OAM Agents (WebGates/AccessGates) ■ 10g OSSO Agents (mod_osso)	■ mod_osso (partner) Note: The mod_osso module is an Oracle HTTP Server module that provides authentication to OracleAS applications.
Oracle Identity Management Infrastructure	Enables secure, central management of enterprise identities.	Enables secure, central management of enterprise identities.
Policy Store	Database	mod_osso and partner application
Partner Applications **Note**: External applications do not delegate authentication. Instead, these display HTML login forms that ask for application user names and passwords. For example, Yahoo! Mail is an external application that uses HTML login forms.	An application that delegates authentication and authorization to OAM and accepts headers from a registered Agent.	An application that delegates authentication to mod_osso and the OracleAS Single Sign-On server. **Note**: After registering mod_osso with OAM 11g, mod_osso delegates authentication to OAM. The mod_osso module enables partner applications to accept authenticated user information once the user is logged in. Re authenticating is avoided by accepting headers from the registered OSSO Agent. The partner application is responsible for determining whether the authenticated user is authorized to use the application.
SSO Engine Manages the user session life cycle, facilitates global logout across all relying parties in the valid user session, and provides consistent service across multiple protocols.	With OAM Agents: ■ Authentication (credential collection) occurs across the HTTP (HTTPS) channel ■ Authorization occurs across the Oracle Access Protocol (OAP) channel	■ mod_osso delegates authentication only and communicates exclusively through the HTTP channel.
Partner Keys	■ During 11g agent registration, a partner key is generated for the agent and also shared with the OAM Server The key is used for encrypting and decrypting SSO cookies ■ During 10g agent registration, a global shared secret key is generated across all of OAM 11g (all WebGates and OAM Server).	■ One key per partner shared between mod_osso and OSSO server
Server Keys	■ During OAM Server installation, one OAM Server key is generated	■ OSSO server's own key ■ One global key per OSSO setup for the GITO domain cookie

Table 7–3 (Cont.) OAM 11g SSO versus OSSO 10g Component Summary

Component Description	OAM 11g	OSSO 10g
Key Storage	▪ **Agent side**: A per agent key is stored locally in the Oracle Secret Store ▪ **OAM 11g server side:** A per agent key, and server key, are stored in the Java Keystore on the server side	▪ **mod_osso side**: partner keys and GITO global key stored locally in obfuscated configuration file ▪ **OSSO server side**: partner keys, GITO global key, and server key are all stored in the directory server
Cookies See Also: "About Single Sign-On Cookies"	Host-based authentication cookie: ▪ **11g WebGate, One per agent**: OAMAuthnCookie_<host:port>_<random number> set by WebGate using the authentication token received from the OAM Server after successful authentication **Note**: A valid OAMAuthnCookie is required for a session. ▪ **10g WebGate**, One ObSSOCookie for all 10g WebGates. ▪ **One for the OAM Server**: OAM_ID	▪ Host-based authentication cookie: **one per partner**: OHS-*host-port* **one for OSSO server**: (but not with OAM 11g) ▪ Domain-level session cookie for global inactivity timeout (GITO) if enabled
Policies	OAM Agents use OAM 11g authentication and authorization policies to determine who gets access to protected applications (defined resources).	mod_osso uses only OAM 11g authentication policies to determine who gets access to defined resources. mod_osso provides authentication only.
Client IP	▪ Maintain this client age, and include it in the host-based cookie: OAMAuthnCookie for 11g WebGate (or ObSSOCookie for 10g WebGate)	▪ Include the original clientage inside the host cookie. In later authentication requests, when the cookie is presented, the original clientIP is compared with the presenter's IP. Rejection occurs if there is no match

See Also: "Introduction to OAM 11g SSO Processing" on page 7-18

7.5.2 About Single Sign-On Cookies

Table 7–4 describes the cookies that can be set or cleared during user login.

Table 7–4 SSO Cookies

SSO Cookie Set at User Login	Description
OAM_ID cookie	Set by the OAM Server. Protected with keys known to the OAM Server only. When a user attempts to access a protected application, the request comes to the SSO Engine and the controller checks for the existence of the cookie: ▪ If the cookie does not exist, user authentication begins. After successful authentication, the user context and token are set by the SSO Engine. The cookie is set with the global user ID (GUID), creation time, and idle timeout details. Information in the cookie is encrypted with the SSO Server key and can be decrypted only by the SSO Engine. ▪ If the cookie exists, then the cookie is decrypted and the sign in flow completes with the authenticated user.
OAMAuthnCookie	Set by each 11g WebGate that is contacted. Protected by the key known to the respective 11g WebGate and the OAM Server. A valid OAMAuthnCookie is required for a session. Note: If the user accesses applications protected by different 11g WebGates, you will have multiple OAMAuthnCookies. See "OAMAuthnCookie for 11g OAM WebGates" on page 7-13.

Table 7–4 (Cont.) SSO Cookies

SSO Cookie Set at User Login	Description
ObSSOCookie	A domain-based cookie for 10g WebGates is set only when a 10g WebGate is contacted. Protected with keys known to the OAM Server only. One global shared secret key for all WebGates. **Note**: This cookie enables backward compatibility and interoperability between OAM 11g and older agents.
OAM_REQ	A transient cookie that is set or cleared by the OAM Server if the Authentication request context cookie is enabled. Protected with keys known to the OAM Server only. Note: This cookie is configured as a high availability option to store the state about user's original request to a protected resource while his credentials are collected and authentication performed.
OAMRequestContext	A transient cookie that is set or cleared by the 11g WebGate. Protected by the key known to the respective 11g WebGate and the OAM Server. Note: This cookie is configured as a high availability option to store the state about user's original request to a protected resource while his credentials are collected and authentication performed.
OHS-*host-port*	Set only when OSSO Agents (mod_osso) are contacted on Oracle HTTP Server (OHS). Protected with the key known to the respective mod_osso agent and the OAM Server. See "mod_osso Cookies". **Note**: This cookie enables backward compatibility and interoperability between OAM 11g and older agents.
GITO cookie	Provides backward compatibility and interoperability between OSSO 10g and OAM 11g. The cookie is created by the OAM Server and accessed or modified by the OAM Server or mod_osso agent.

For details about configuring authentication and authorization policies, see Chapter 9, "Managing Policies to Protect Resources and Enable SSO".

7.5.3 About Single Sign-On Cookies

- OAMAuthnCookie for 11g OAM WebGates
- ObSSOCookie for 10g OAM WebGates
- OAM_REQ Cookie
- mod_osso Cookies

7.5.3.1 OAMAuthnCookie for 11g OAM WebGates

There is one OAMAuthnCookie_<host:port>_<random number> set by each 11g WebGate using the authentication token received from the OAM Server after successful authentication. A valid OAMAuthnCookie is required for a session.

SSL Connections: Administrators can ensure the ObSSOCookie is only sent over an SSL connection and prevents the cookie from being sent back to a non-secure Web server by configuring SSL and then specifying Simple or Cert mode for Agents and Servers. For details, see "About Communication Between OAM Servers and WebGates" on page 4-4.

Cookie Expiration: For 11g WebGate and OAMAuthnCookie, expiration is controlled by the "tokenValidityPeriod" parameter, which controls the valid token (or cookie) time.

This key is known to both the 11g WebGate and SSO Engine and is used for encrypting OAMAuthnCookie. The SSO engine key (only known to the SSO Engine) is used for encrypting the OAM_ID OAM Server cookie.

Similar to ObSSOCookie.

7.5.3.2 ObSSOCookie for 10g OAM WebGates

Oracle Access Manager 11g sets a key-based cookie *ObSSOCookie* for each user or application that accesses a resource protected by a 10g WebGate. The key is set up during agent registration and is known to both the agent and SSO Engine (shared between them). This key is different from the OAM Server (or SSO Engine) key.

Removing the ObSSOcookie causes the 10g WebGate to log the user out and requires the user to re-authenticate the next time he or she requests a resource that is protected by the Access System.

The WebGate sends the ObSSOCookie to the user's browser upon successful authentication. This cookie can then act as an authentication mechanism for other protected resources that require the same or a lower level of authentication. When the user requests access to a browser or another resource, the request flows to the OAM Server. The user is logged in, and the ObSSOCookie is set. The OAM Server generates a session token with a URL that contains the ObSSOCookie. Single sign-on works when the cookie is used for subsequent authorizations in lieu of prompting the user to supply authorization credentials.

When the cookie is generated, part of the cookie is used as an *encrypted session token*. The single sign-on cookie does not contain user credentials such as user name and password.

SSL Connections: Administrators can ensure the ObSSOCookie is only sent over an SSL connection and prevents the cookie from being sent back to a non-secure Web server by configuring SSL and then specifying Simple or Cert mode for Agents and Servers. For details, see "About Communication Between OAM Servers and WebGates" on page 4-4.

Cookie Expiration: Administrators can specify the desired Cookie Session Time in the OAM Agent registration. For more information, see "Registering and Managing WebGate Agents Using the Administration Console" on page 5-9.

7.5.3.3 OAM_REQ Cookie

In high availability configurations, the Request Cache type must be changed from BASIC to COOKIE using Infrastructure Security custom WLST commands.

Note: You must invoke the WLST script from the Oracle Common home. See "Using Custom WLST Commands" in the *Oracle Fusion Middleware Administrator's Guide.*

See Also:

- "Running WLST Commands for OAM Operations" on page F-4
- Table 7–4, " SSO Cookies"

7.5.3.4 mod_osso Cookies

The mod_osso module is the Oracle HTTP Server module that provides authentication to OracleAS applications. This module resides on the Oracle HTTP Server that enables applications protected by OracleAS Single Sign-On to accept HTTP headers in lieu of a user name and password once the user has logged into the OracleAS Single Sign-On server. The values for these headers are stored in a mod_osso cookie.

Located on the application server, mod_osso simplifies the authentication process by serving as the sole partner application to the single sign-on server. In this way, mod_ osso renders authentication transparent to OracleAS applications. The administrator for these applications is spared the burden of integrating them with an SDK. After authenticating a user, mod_osso transmits the simple header values that applications may use to authorize the user:

GITO Cookie: Needed in special cases to support timeout when multiple types of agents (mod_osso and WebGate) are working with OAM 11g. Server side session managers can check the validity of the cookie for expiry and timeout during session validation. Global logout is required for OSSO Agents (mod_osso) to ensure that logging out of a session on any entity propagates the logout to all entities.

When a user is authenticated by OSSO 10g, the OSSO Server sets GITO cookie. Once the partner cookie (OHS cookie) is set, OHS does not route the request to the server. Instead, on every access, OHS decrypts the GITO cookie and updates the last activity timestamp. During request processing, if any partner detects that current time has surpassed GITO timeout (last activity time + GITO timeout), the request is sent to OSSO 10g in forced authentication mode. When a request reaches OSSO server in forced authentication mode, server chooses to ignore SSO_ID cookie and challenges user for credentials, considering it as a fresh request. After successful authentication, SSO_ID and GITO cookie are updated.

This is enabled (using the `editGITOValues` WLST command), as described in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference.*

OssoSecureCookies Directive: Add the OssoSecureCookies directive to set the Secure flag on all cookies. This tells the browser to only transmit those cookies on connections secured by HTTPS. An example of this directive in a mod_osso configuration (mod_osso.conf), is as follows:

```
<IfModule mod_osso.c>
OssoIpCheck off
OssoIdleTimeout off
OssoSecureCookies on
OssoConfigFile osso/osso.conf
<Location /j2ee/webapp>
require valid-user
AuthType Basic
</Location>
</IfModule>
```

For more information, see *Oracle Application Server Single Sign-On Administrator's Guide*.

7.6 Introduction to OAM 11g Single Sign-On Implementation Types

This section provides the following topics to introduce various single sign-on implementation types:

- Application SSO
- Single Sign-On with OAM 11g
- Cross-Network Domains and Oracle Access Manager 11g

7.6.1 Application SSO

Oracle Access Manager enables administrators to create a web of trust in which a user's credentials are verified once and are provided to each application the user runs. Using these credentials, the application does not need to re-authenticate the user with its own mechanism.

Application single sign-on allows users who have been authenticated by Oracle Access Manager to access applications without being re-authenticated.

There are two ways to send a user's credentials:

- **Using Cookies:** A specific value is set on the browser's cookie that the application must extract to identify a user.
- **Using Header Variables:** An HTTP header set on the request by the agent and visible to the application.

> **Note:** Both forms require administrators to enter the appropriate responses within the policy. For more information, see "Introduction to Policy Responses for SSO" on page 9-28.

Header response values are inserted into a request by an OAM Agent, and can only be applied on Web servers that are protected by an agent. registered with OAM 11g If the policy includes a redirect URL that is hosted by a Web server not protected by OAM, header responses are not applied.

For example, when a user authenticates, she might be redirected to a portal index page:

```
http://mycompany.com/authnsuccess.htm
```

For authentication failure, an authentication action might redirect the user to an error page or a self-registration script:

```
http://mycompany.com/authnfail.htm
```

7.6.2 Single Sign-On with OAM 11g

This section introduces single sign-on processing using OAM 11g.

> **See Also:** "Introduction to OAM 11g SSO Processing" on page 7-18

Oracle Access Manager provides a proprietary multiple network domain SSO capability that predates Oracle Identity Federation. If this is implemented in your OAM 10g deployment, you can register OAM 10g Agents with OAM 11g to continue this support.

SSO with Mixed Release Agents

After registering agents with Oracle Access Manager 11g, OAM Servers provide seamless support for OAM 10g and 11g Agents and 10g OSSO Agents (mod_osso) in any combination.

Reverse-Proxy SSO

If you are going to use a reverse proxy in a single sign-on configuration, be sure either to set the IPvalidation parameter to false or to add the proxy IP address to the IPValidationExceptions list in the WebGate registration. Otherwise, the reverse proxy hides the client's IP address.

In some situations the Reverse Proxy does not pass the 10g WebGate ObSSOCookie to Oracle WebLogic after a successful authentication. To avoid this issue, use Form Based authentication instead of Basic Over LDAP when using Reverse Proxy with Oracle WebLogic. For 11g WebGate, a user-defined parameter (filterOAMAuthnCookie (default true)) can be used to prevent the OAMAuthnCookie from being passed to downstream applications for security consideration. If you do want to pass the cookie on, then set the parameter to false.

Multiple WebLogic Server Domain SSO

OAM 11g supports SSO in multiple WebLogic administration domains. You can define multiple WebLogic administration domains based on different system administrators' responsibilities, application boundaries, or the geographical locations of WebLogic servers. Conversely, you can use a single domain to centralize all WebLogic Server administration activities.

> **Note:** All Managed Servers in a cluster must reside in the same domain; you cannot split a cluster over multiple domains. All Managed Servers in a domain must run the same version of the Oracle WebLogic Server software. The Administration Server can run either the same version as the Managed Servers in the domain, or a later service pack.

There are two basic types of WebLogic administration domains:

- Domain with Managed Servers: A simple production environment can consist of a domain with several Managed Servers that host applications, and an Administration Server to perform management operations. In this configuration, applications and resources are deployed to individual Managed Servers; similarly, clients that access the application connect to an individual Managed Server.

  Production environments that require increased application performance, throughput, or availability may configure two or more of Managed Servers as a cluster. Clustering allows multiple Managed Servers to operate as a single unit to host applications and resources. For more information about the difference between a standalone and clustered Managed Servers, see Managed Servers and Clustered Managed Servers.

- Standalone WebLogic Server domain: For development or test environments, you may want to deploy a single application and server independently from servers in a production domain. In this case, you can deploy a simple domain consisting of a single server instance that acts as an Administration Server and also hosts the applications you are developing. The examples domain that you can install with WebLogic Server is an example of a standalone WebLogic Server domain.

All Managed Servers in a cluster must reside in the same domain; you cannot split a cluster over multiple domains. All Managed Servers in a domain must run the same version of the Oracle WebLogic Server software. The Administration Server can run either the same version as the Managed Servers in the domain, or a later service pack.

Each domain's configuration is stored in a separate configuration file (config.xml), which is stored on the Administration Server along with other files such as logs and security files. When you use the Administration Server to perform a configuration task, the changes you make apply only to the domain managed by that Administration Server. To manage another domain, use the Administration Server for that domain. For this reason, the servers instances, applications, and resources in one domain should be treated as being independent of servers, applications, and resources in a different domain.You cannot perform configuration or deployment tasks in multiple domains at the same time.

Each domain requires its own Administration Server for performing management activities. When you use the Administration Console to perform management and monitoring tasks, you can switch back and forth between domains, but in doing so, you are connecting to different Administration Servers.

If you have created multiple domains, each domain must reference its own database schema. You cannot share a configured resource or subsystem between domains. For example, if you create a JDBC data source in one domain, you cannot use it with a Managed Server or cluster in another domain. Instead, you must create a similar data source in the second domain. Furthermore, two or more system resources cannot have the same name.

7.6.3 Cross-Network Domains and Oracle Access Manager 11g

Unlike OAM 10g, OAM 11g supports cross-network-domain single sign-on out of the box. During single sign-off with OAM 11g:

- The SSO cookie set by 11g OAM server is a host cookie that works across the network domains. The WebGate clears its standalone Agent cookie and then redirects to the OAM 11g Server for session clearing.
- In the case of OAM 10g WebGates, which do not have a standalone Agent cookie, logout occurs only on the server side with no redirection required.
- In the case of 11g WebGates and OSSO agents that support a standalone agent cookie, the agent logout callback URL is called in parallel. The agents accessed in a session and agents from multiple domains are all called in parallel, depending on the number of concurrent connections supported in the browser.

> **Note:** Oracle recommends Oracle Identity Federation for a standards-based, multi-protocol, cross-network-domain single sign-on.

7.7 Introduction to OAM 11g SSO Processing

This section provides the following topics:

- About SSO Log In Processing
- About SSO Log In Processing with OAM Agents
- About SSO Login Log In Processing with OSSO Agents (mod_osso)
- About Single Sign-On Processing with Mixed Release Agents

7.7.1 About SSO Log In Processing

Single Sign On login and logout processing determines whether the user is a valid user and whether the user state is valid or invalid (either a first time user OR the user session has expired). Session management support locates, persists, and cleans up the user session context and user token.

The following topics provide more information:

- Login
- Login with Self-Service Provisioning Applications
- Login and Auto Login for Applications Using Oracle ADF Security

7.7.1.1 Login

The first time a user attempts to access a protected resource, she is prompted for her credentials based on the authentication scheme and level for the resource (typically a user id and password is needed).

Authentication fails if the wrong user ID or password is entered. In this case, the user is not authenticated and another prompt for credentials appears.

Following successful authentication, a check of authorization policies is made to confirm this user is authorized to access the particular resource. Upon successful authorization, information is passed to the application. The user is not asked to sign in again until her session expires or if she requests a resource with a higher level of authentication.

7.7.1.2 Login with Self-Service Provisioning Applications

Provisioning does not create the session in Oracle Access Manager. When a new user uses a self-service provisioning application to create an account, he is prompted for his user id and password again when accessing an application.

The protected application is directed to Oracle Access Manager 11g, which requests the user’s credentials. For example if Oracle Identity Manager is protected by OAM 11g, the user request is redirected to Oracle Access Manager from which a request to enter credentials is made.

7.7.1.3 Login and Auto Login for Applications Using Oracle ADF Security

Oracle Platform Security Services (OPSS) comprise Oracle WebLogic Server’s internal security framework. On the Oracle WebLogic Server, you can run a Web application that uses Oracles Application Development Framework (Oracle ADF) security, integrates with Oracle Access Manager 11g SSO, and uses OPSS SSO for user authentication.

For more information, see Appendix C, "Integrating Oracle ADF Applications with Oracle Access Manager 11g SSO".

7.7.2 About SSO Log In Processing with OAM Agents

Oracle Access Manager authenticates each user with a customer-specified authentication method to determine the identity and leverages information stored in the user identity store. Oracle Access Manager authentication supports several authentication methods and different authentication levels. Resources with varying degrees of sensitivity can be protected by requiring higher levels of authentication that correspond to more stringent authentication methods.

When a user tries to access a protected application, the request is received by OAM which checks for the existence of the SSO cookie.

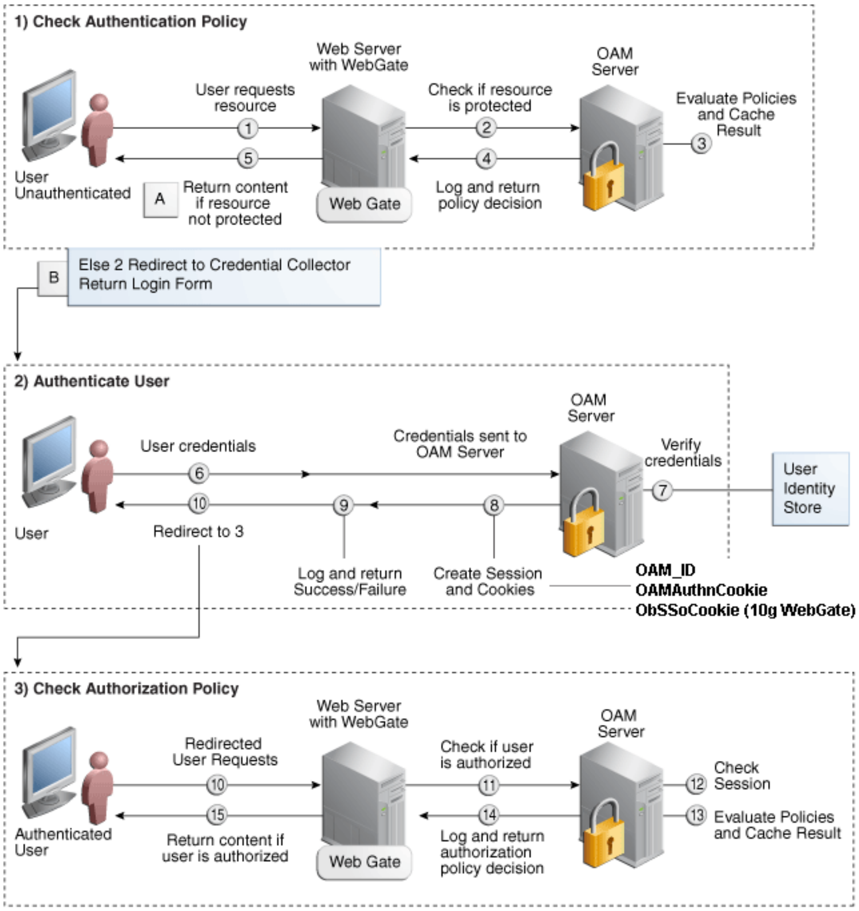
After authenticating the user and setting up the user context and token, OAM sets the SSO cookie and encrypts the cookie with the SSO Server key (which can be decrypted only by the SSO Engine).

Depending on the actions (responses in OAM 11g) specified for authentication success and authentication failure, the user may be redirected to a specific URL, or user information might be passed on to other applications through a header variable or a cookie value.

Based on the authorization policy and results of the check, the user is allowed or denied access to the requested content. If the user is denied access, she is redirected to another URL (specified by the administrator in WebGate registration).

Figure 7–2 shows the processes involved in evaluating policies, validating a user's identity, authorizing the user for a protected resource, and serving the protected resource. This example shows the OAM Agent flow (WebGate/AccessGate 10g or WebGate 11g). There are slight variations with 11g WebGates.

Figure 7–2 SSO Log-in Processing with OAM Agents


1) Check Authentication Policy
Web Server with WebGate
OAM Server
User requests resource
Check if resource is protected
Evaluate Policies and Cache Result
1
5
2
4
3
User Unauthenticated
A
Return content if resource not protected
Web Gate
Log and return policy decision
B
Else 2 Redirect to Credential Collector Return Login Form
2) Authenticate User
OAM Server
User credentials
Credentials sent to OAM Server
Verify credentials
6
10
9
8
7
User Identity Store
User
Redirect to 3
Log and return Success/Failure
Create Session and Cookies
OAM_ID
OAMAuthnCookie
ObSSoCookie (10g WebGate)
3) Check Authorization Policy
Web Server with WebGate
OAM Server
Redirected User Requests
Check if user is authorized
Check Session
10
15
11
14
12
13
Evaluate Policies and Cache Result
Authenticated User
Return content if user is authorized
Web Gate
Log and return authorization policy decision

Process overview: SSO Log-in Processing with OAM Agents

1. The user requests a resource.
2. WebGate forwards the request to OAM for policy evaluation.
3. OAM:
   - Checks for the existence of an SSO cookie.
   - Checks policies to determine if the resource protected and if so, how?
4. OAM Server logs and returns decisions.
5. WebGate responds as follows:

   a. **Unprotected Resource**: Resource is served to the user.

   b. **Protected Resource**:

   Request is redirected to the credential collector.

   The login form is served based on the authentication policy.

Authentication processing begins

6. User sends credentials.
7. OAM verifies credentials.
8. OAM starts the session and creates the following host-based cookies:
   - **One per partner**: OAMAuthnCookie set by 11g WebGates (ObSSOCookie set by 10g WebGate) using the authentication token received from the OAM Server after successful authentication.

     Note: A valid cookie is required for a session.
   - **One for OAM Server**: OAM_ID
9. OAM logs Success or Failure.
10. Credential collector redirects to WebGate and authorization processing begins.
11. WebGate prompts OAM to look up policies, compare them to the user's identity, and determine the user's level of authorization.
12. OAM logs policy decision and checks the session cookie.
13. OAM Server evaluates authorization policies and cache the result.
14. OAM Server logs and returns decisions
15. WebGate responds as follows:
    - If the authorization policy allows access, the desired content or applications are served to the user.
    - If the authorization policy denies access, the user is redirected to another URL determined by the administrator.

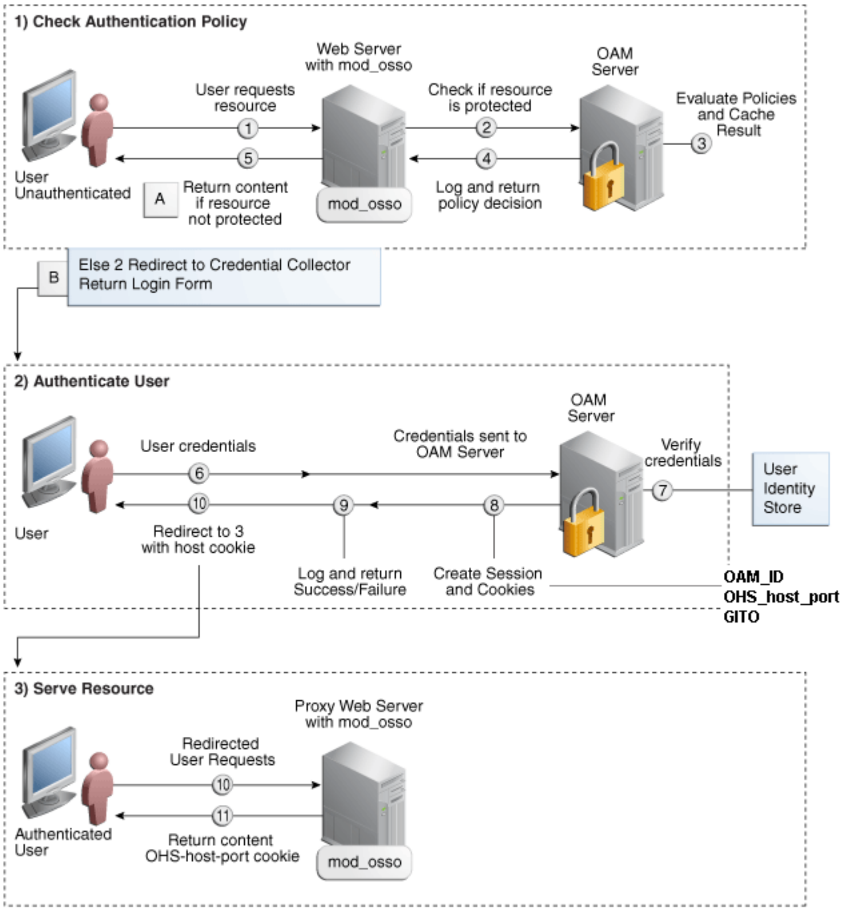
7.7.3 About SSO Login Log In Processing with OSSO Agents (mod_osso)

SSO login processing with registered OSSSO Agents (mod_osso) is similar to login processing with WebGates. However, mod_osso provides only authentication using OAM 11g authentication policies.

> **Note:** mod_osso does not support authorization either on its own or using OAM 11g policies.

Figure 7–3 illustrates the login processing with mod_osso and OAM 11g.

Figure 7–3 SSO Login Processing with OSSO Agents


1) Check Authentication Policy
Web Server with mod_osso
OAM Server
User requests resource
Check if resource is protected
Evaluate Policies and Cache Result
1
2
3
4
5
User Unauthenticated
A
Return content if resource not protected
mod_osso
Log and return policy decision
B
Else 2 Redirect to Credential Collector
Return Login Form
2) Authenticate User
OAM Server
User credentials
Credentials sent to OAM Server
Verify credentials
User Identity Store
6
10
9
8
7
User
Redirect to 3 with host cookie
Log and return Success/Failure
Create Session and Cookies
OAM_ID
OHS_host_port
GITO
3) Serve Resource
Proxy Web Server with mod_osso
Redirected User Requests
10
11
Authenticated User
Return content OHS-host-port cookie
mod_osso

Process overview: SSO Log-in Processing with OSSO Agents

1. The user requests a resource.
2. mod_osso forwards the request to OAM for policy evaluation.
3. OAM:
   - Checks for the existence of an SSO cookie.
   - Checks policies to determine if the resource protected and if so, how?
4. OAM Server logs and returns decisions.
5. mod_osso responds as follows:
   a. **Unprotected Resource**: Resource is served to the user.
   b. **Protected Resource**:

      Request is redirected to the credential collector.

      The login form is served based on the authentication policy.

      Authentication processing begins
6. User sends credentials.
7. OAM verifies credentials.

8. OAM starts the session, passes an authentication token to the application, and creates the following cookies:
   - **One per partner**: OHS_host_port
   - **One for the OAM Server**: OAM_ID
   - **Global Inactivity Out**: A domain-level cookie GITO
9. OAM logs Success or Failure.
10. Credential collector redirects to mod_osso, which transmits the simple header values that applications cab use to authorize the user:
11. Resource is served upon authentication success and the OHS-*host-port* cookie is set.

7.7.4 About Single Sign-On Processing with Mixed Release Agents

OAM 11g Servers provide similar SSO run-time processing regardless of the Agent type or release.

8 Managing Policy Components

Shared policy components can be used in any OAM policy. This chapter describes how administrators can manage policy components.

This chapter includes the following topics:

- Prerequisites
- Introduction to Managing Policy Components
- Managing Resource Types
- Managing Host Identifiers
- Managing Authentication Modules
- Managing Authentication Schemes

8.1 Prerequisites

An OAM Administration Console and at least one OAM Server must be installed and running within a WebLogic Server domain.

Oracle recommends that you review the following topics before performing activities in this chapter.

- Learn more about the policy model and components from "Introduction to the OAM 11g Policy Model" on page 7-3
- Review a comparison of the current policy model versus other models in "Comparing the OAM 11g Policy Model with OAM 10g" on page 7-1
- Learn more about the Administration Console and controls from Chapter 2, "Getting Started with OAM Administration and Navigation"

8.2 Introduction to Managing Policy Components

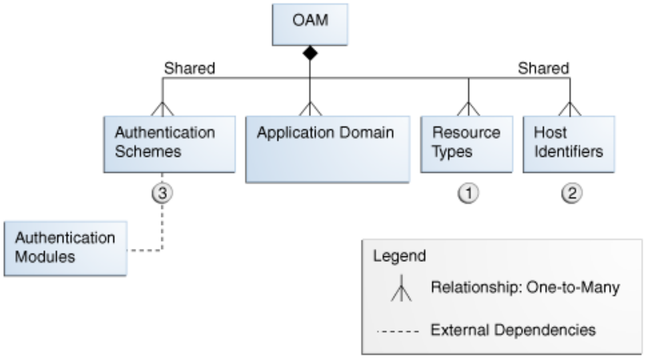
This section introduces the Oracle Access Manager 11g policy model and the global components within it.

The Oracle Access Manager 11g policy model provides both authentication and authorization services within the context of an OAM application domain. The policy model relies on external user identity stores and on authentication modules, which are a part of the overall system configuration.

> **Note:** Earlier releases of Oracle Access Manager provided authentication and authorization services within the context of an OAM policy domain. OracleAS SSO 10g provides only authentication.

Figure 8–1 illustrates the different elements within the policy model for Oracle Access Manager 11g. The top-level construct of the Oracle Access Manager 11g policy model is the OAM application domain. Additional information follows the figure.

Figure 8–1 Policy Components: Relationship to an Application Domain



Policy Components: Global authentication schemes, resource types, and host identifiers that can be used in any application domain. Managing policy components is described throughout this chapter

Application Domains: A logical container for resources (or sets of resources), and the associated authentication and authorization policies that dictate who can access specific resources. The size and number of application domains is up to the administrator. For details, see Chapter 9, "Managing Policies to Protect Resources and Enable SSO".

8.3 Managing Resource Types

This section includes the following topics:

- About Resource Types and Their Use
- About the Resource Type Page
- Creating a Non-HTTP Resource Type
- Searching for a Specific Resource Type
- Deleting Resource Types

8.3.1 About Resource Types and Their Use

When adding a resource to an application domain, administrators must choose from a list of defined Resource Types. then enter a specific URL. For HTTP type resources, include a host identifier. For non-HTTP resource types, use the type name.

The default resource type, HTTP, is used with HTTP and HTTPS protocols. Operations associated with the HTTP resource type need not be defined by an administrator. Instead, policies developed and applied to the resource apply to all operations.

When adding an HTTP type resource to an application domain, administrators must choose from a list of existing host identifiers and add the resource URL.

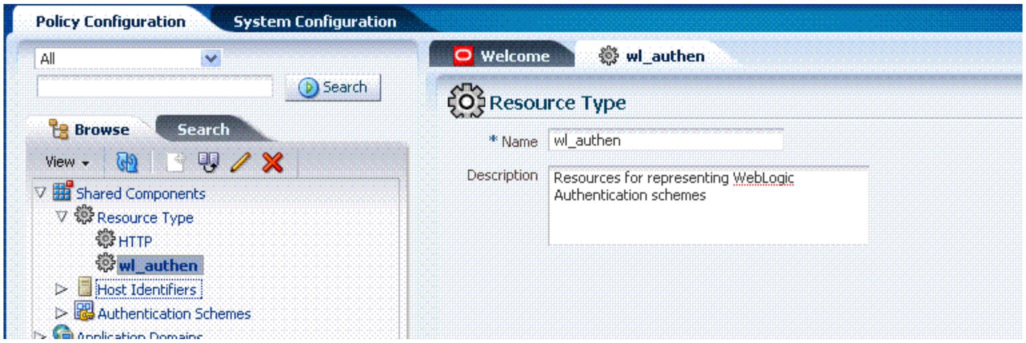
Administrators can define a resource type for non-HTTP resources. Non-HTTP resource types have no associated host identifier. When adding non-HTTP resources to an application domain, administrators must enter the type name into the Resource URL field as a pointer. The name cannot match any host Identifier (and vice versa). This is not a relative HTTP URL.

For instance, a non-HTTP resource type named wl_authen is available to use with resources deployed in a WebLogic container. Resources of type wl_authen, require a custom AccessGate. The protected resource is accessed using its URL on the Oracle WebLogic Server.

8.3.2 About the Resource Type Page

In the OAM Administration Console, resource types are organized with other Components under the Policy Configuration tab, as shown in Figure 8–2. The navigation tree on the left shows two default resource types: HTTP for internet protocols and wl_authen for resources deployed in a WebLogic container.

Figure 8–2 The Default wl_authen Resource Type Definition



* Name: wl_authen

Description: Resources for representing WebLogic Authentication schemes

The HTTP resource type is used for Web applications protected by Oracle Access Manager 11g.

The `wl_authen` resource type is used for Fusion Middleware application scenarios in combination with in Oracle Access Manager 11g and one of the following OAM Authentication Provider configurations, as described in the *Oracle Fusion Middleware Application Security Guide*.

- Authenticator
- Identity Asserter with Oracle Web Services Manager

Table 8–1 describes the elements in the resource type definition.

Table 8–1 Resource Type Definition

Element	Description
Name	Required. A unique name of up to 30 alpha or numeric characters. **Note**: A non-HTTP Resource Type name cannot match a Host Identifier (and vice versa).
Description	Optional. Use this field to describe the purpose of this resource type using up to 200 alpha or numeric characters. For example: Resources representing WebLogic Authentication schemes.

Following topics describe how to create, modify, and delete a resource type.

8.3.3 Creating a Non-HTTP Resource Type

Users with valid OAM Administrator credentials can use the following procedure to create a new resource type definition for non-HTTP resources, as explained in "About Resource Types and Their Use" on page 8-2.

In this case, there is no host identifier associated with the resource type. Instead, you must enter a name into the field. Any resource type with name other than HTTP is considered non-HTTP.

See Also: About the Resource Type Page

To create a non-HTTP resource type

1. From the Policy Configuration tab, navigation tree, Shared Components node, click Resource Type, then click the Create command button in the tool bar.
2. Fill in fields on the Resource Type page to identify this new type:
   a. Name:
   b. Description (optional)
3. Click Apply to submit the new resource type (or close the page without applying changes).
4. Close the Confirmation window.
5. Close the Resource Type page.

8.3.4 Searching for a Specific Resource Type

Users with valid OAM Administrator credentials can use the following procedure to locate a non-HTTP resource type.

To search for a resource type

1. Activate the Policy Configuration tab.
2. From the search type list, choose Resource Type to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   ```
   my_resource_type
   ```

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.
6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

8.3.5 Deleting Resource Types

Users with valid OAM Administrator credentials can use the following procedure to remove a non-HTTP resource type only. A validation error occurs if you attempt to delete a resource type that is being used by a resource in an application domain.

Prerequisites

Each resource in an application domain has an assigned type. If you intend to delete the assigned resource type you must first modify the resource definition in an application domain that uses this resource type

Note: You cannot remove default resource types HTTP or wl_authen.

To remove a resource type

1. From the Policy Configuration tab, Shared Components node, double-click the name of the Resource Type to confirm the definition, and then close the page.
2. Click the name in the navigation tree, click the Delete button in the tool bar, and confirm removal in the Confirmation window.
3. Confirm that the Resource Type is removed.

8.4 Managing Host Identifiers

This section describes host identifiers and their use as well as how to create, modify, or remove a host identifier. Topics here include:

- About Host Identifiers
- About Virtual Web Hosting
- About the Host Identifier Page
- Creating a Host Identifier
- Searching for a Host Identifier Definition
- Viewing or Editing a Host Identifier Definition
- Deleting a Host Identifier Definition

8.4.1 About Host Identifiers

Policies protect resources on computer hosts. Within Oracle Access Manager, the computer host is specified independently using a host identifier.

Based on a defined host identifier, administrators can add specific resources to an application domain and apply policies to protect those resources.

Registered Agents protect all requests that match the addressing methods defined for the host identifier used in a policy. A request sent to any address on the list is mapped to the official host name and OAM can apply the policies that protect the resource and OAM can apply the policies that protect the resource.

A host identifier is automatically created when an Agent (and application) are registered using either the OAM Administration Console or the remote registration tool. Administrators can manually add a host identifier if an application and resources exist on a host that does not have a mapped host identifier. Also, administrators can modify an existing host identifier to add in the new host name variations. For instance, adding another proxy Web server with a different host name requires a new host name variation.

For more information, see:

- Host Identifier Usage
- Host Identifier Guidelines
- Host Identifier Variations

8.4.1.1 Host Identifier Usage

At design time, the host identifier can be used while defining which resources belong to a specific application domain. Resources are scoped using their host identifier (HTTP) or type (non-HTTP). This combination uniquely identifies them across Oracle Access Manager.

> **Note:** Each resource should be unique across all application domains; each resource and host identifier combination must be unique across all application domains.

Runtime Usage

At run time, Web server host information in the access query from an OAM agent is mapped to a host identifier and associated with the resource that is being accessed by a user. The OAM agent obtains the Web server host information in one of two ways:

- If the Preferred Host parameter is configured for virtual Web hosting support (see"About Virtual Web Hosting" on page 8-7), Web server host information for the given request is obtained from the Web server.
- If the Preferred Host parameter directly specifies the Web server host information, it is always used irrespective of the Web server's own host information.

This allows for the Resources to be specified in terms of logical host names in their Host Identifiers, instead of the host names matching the present deployment of the Web server.

A user accessing aseng-wiki, would enter:

http://aseng-wiki.us.oracle.com/mywikipage

Here, "mywikipage" is the resource URL and "aseng-wiki.us.oracle.com" is the host. Matching this host and port (port is 80) provides the host identifier.

Preferred Host

Web server host information is generally acquired by setting the Preferred Host string of the OAM Agent. If the Agent is actively protecting multiple virtual hosts, this string can be set to server_name to ensure that the actual request hostname is correctly picked up from the Web server's request object. For more information, see "About Virtual Web Hosting" on page 8-7

Authenticating Hosts and Challenge Redirect in Authentication Schemes

When a user attempts to access a protected resource URL, she is redirected to the server specified in the Challenge Redirect field of the authentication scheme. If the authentication challenge is to be processed by another host, the name of that host must be defined to be available in the Host Identifiers list. For example, if a user is redirected to an SSL-enabled server for authentication, that server must be defined as a host identifier.

Note: If you enter a host name in the Challenge Redirect field of an authentication scheme, it must be defined as a Host Identifier.

8.4.1.2 Host Identifier Guidelines

Each host identifier can be defined to represent one or more Web server hosts. Following are several important guidelines for host identifiers:

- Each host name must be unique.
- Each *host name*:*port* pair must be unique.
- Each *host name*:*port* pair must belong to only one host identifier.
- Each *host name*:*port* pair must match the end user's entry exactly.
- A Host Identifier name cannot match a non-HTTP Resource Type name (and vice versa).
- Each resource and host identifier combination must be unique across all application domains.

For more information, see "Host Identifier Variations".

8.4.1.3 Host Identifier Variations

Host identifiers are used to simplify the identification of a Web server host by defining all possible hostname variations. Host identifiers consist of a list of all URL addressing methods. A host identifier must be configured for each Web site or virtual Web site that you want to protect with Oracle Access Manager.

You can identify Web server hosts to Oracle Access Manager in various ways, for example, by providing a computer name or an IP address. The following are examples of how the same host can be addressed:

- site.com
- site.com:80
- www.site.com
- www.site.com:80
- 216.200.159.58
- 216.200.159.58:80

8.4.2 About Virtual Web Hosting

A virtual host referees to the situation where the same host has multiple sites being served either based on multiple NIC cards (IP based) or multiple names (for example, abc.com and def.com resolving to same IP).

Consider a use case where you have two virtual hosts configured on an OHS Server acting as reverse proxy to OAM Server, as follows:

- One virtual host is configured in two-way SSL mode
- One virtual host configured in non-SSL mode

Suppose there are two resources protected with different authentication schemes and application domains:

- /resource1 is protected by a X509Scheme with a Challenge URL (to define the credential collection URL) of https://sslvhost:port/

  When the user accesses /resource1 he is redirected to the OHS Server on the SSL port for authentication and is asked for the X.509 Certificate.

- /resource2 is protected by a LDAPScheme on the second virtual host with a Challenge Redirect of http://*host:port*/

  When user accesses /resource2 he is redirected to second virtual host which is in non-SSL mode (or in one way SSL mode if required). The Login form for LDAP authentication is displayed.

> **Note:** Your deployment can support X.509 and Form authentication with 10g mod_osso. However, mod_osso can be configured for only one SSO Server. In this case, the Agent redirects to Oracle Access Manager on the non-SSL virtual host. The credential collector checks the Authentication Scheme's Challenge URL parameter for the resource and redirects back to the HTTPS virtual host for X509 authentication.

To support virtual Web hosting, you must specify a specific value, described in the following paragraphs, in the Preferred HTTP host field of the OAM Agent registration. The following summarizes configuration for virtual servers:

- To support virtual hosts on most Web servers other than Apache-based servers, you must set the Preferred HTTP Host value to HOST_HTTP_HEADER.

  If you specify this value, when user's browser sends a request, the WebGate sets the value of the Preferred HTTP Host to the host value in the request. For example, suppose a user enters the string myweb2 in a URL:

  http://myweb2

  On the Web server, if one of the Web sites has a host named myweb2, the request is served by the matching virtual site.

- On Apache-based servers, for example, Apache, Apache 2, IBM HTTP Server, Oracle HTTP Server, and so on, the Preferred HTTP Host value must be set to SERVER_NAME.

> **Note:** The SERVER_NAME value is not supported for any host other than an Apache-based server. If you set this value for a non-Apache-based server, users will be unable to access any resources that are protected by WebGate on that Web server. Users will, instead, receive an error that the WebGate configuration is incorrect.
>
> The "ServerName" directive needs to be explicitly set with 7777 along with the hostName. This is irrespective of the "Listen" directive is set correctly. The Server sometimes requires this value explicitly to identify itself, most often it can identify itself automatically.

8.4.3 About the Host Identifier Page

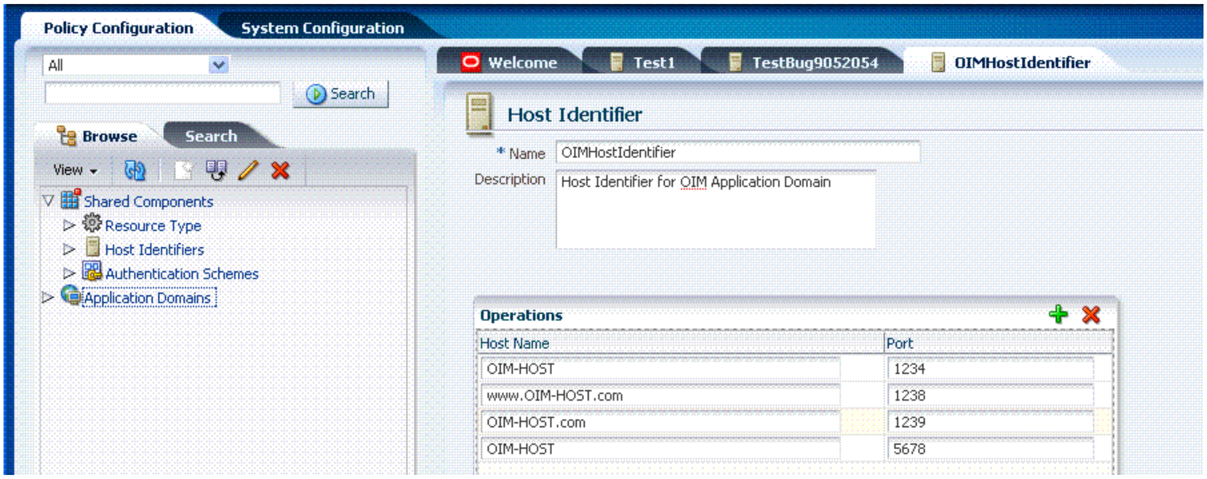
A host identifier is automatically created when an Agent (and application) are registered using either the OAM Administration Console or the remote registration tool. In the application domain that is registered with the Agent, the host identifier is used automatically.

Administrators can use the console to create and manage host identifiers. Within the OAM Administration Console, host identifiers are organized under Shared Components, on the Policy Configuration tab navigation tree. Administrators can manually create a new host identifier definition, modify a definition, delete a definition, or copy an existing definition to use as a template. The name of the copy is based on the original definition name. For example, if you copy a definition named *host3*, the copy is named *copy of host3*.

Figure 8–3 illustrates a typical Host Identifier configuration page in the Administration Console, where you enter the canonical name for the host, and every other name by which the same host can be addressed by users.

> **Note:** Each host identifier must be unique. You cannot use the same host name and port in any other host identifier definition.

Figure 8–3 Host Identifier Page



* Name: OIMHostIdentifier

Description: Host Identifier for OIM Application Domain

Operations

Host Name	Port
OIM-HOST	1234
www.OIM-HOST.com	1238
OIM-HOST.com	1239
OIM-HOST	5678

Table 8–2 describes the host identifier definition.

Table 8–2 Host Identifier Definition

Property	Description
Name	A unique name for this definition. Use only upper- and lower-case alpha characters. No punctuation or special characters are allowed.
Description	The optional description, up to 200 characters, that explains the use of this configuration.
Operations	▪ Host Name: A list of the various host names or permutations that users might use when accessing the application. See also: "Host Identifier Variations" on page 8-7 and "Host Identifier Guidelines" on page 8-7. ▪ Port: The Web server port used by each host or permutation

8.4.4 Creating a Host Identifier

Users with valid OAM Administrator credentials can use the following procedure to create a host identifier definition. This is needed if an application and resources were manually added to a host that has no mapped host identifier.

> **Note:** If you copy an existing definition to use as a template, you must modify all unique identifiers in the copy.

See Also: "About Host Identifiers" on page 8-5

To manually create a Host Identifier

1. From the Policy Configuration tab, navigation tree, expand Shared Components.
2. Click Host Identifiers, then click the Create command button in the tool bar.

   Alternatively: Expand the Host Identifiers node, double-click the name of a definition to use as a template, then click the Duplicate button to create a copy named copyof*name.*
3. On the Host Identifier page, fill in the:
   a. Name
   b. Description
   c. Operations: Add (or remove) host name and port variations in the Operations list.

      Add: Click + and enter a new host name and port combination.

      Remove: Click a host name, then click the Delete button to remove it.
4. Repeat step 3c as needed to identify all variations of this host that users can access.
5. Click Apply to submit the new definition (or close the page without applying changes).
6. Close the Confirmation window, and confirm the new definition is listed in the navigation tree.

8.4.5 Searching for a Host Identifier Definition

Users with valid OAM Administrator credentials can perform the following task to search for a specific host identifier.

See Also: "About Search Controls" on page 2-24

To search for a host identifier

1. Activate the Policy Configuration tab.
2. From the search type list, choose Host Identifiers to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   ```
   my_host_identifier
   ```
4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.

- **View**: Select a View menu item to alter the appearance of the results table.

6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

8.4.6 Viewing or Editing a Host Identifier Definition

Users with valid OAM Administrator credentials can use the following procedure to modify a host identifier definition. This can include adding, changing, or removing individual host identifiers from the definition. For instance, when adding another proxy Web server with a different host name, you might need to modify an existing host identifier definition to add the new host name variation.

Prerequisite: Inventory application domains that refer to the host identifier and

> **Note:** After viewing settings, you can either close the page or modify settings as needed.

See Also: "About the Host Identifier Page" on page 8-8

To view or modify a Host Identifier

1. From the Policy Configuration tab, navigation tree, expand:
   a. Shared Components node
   b. Host Identifiers node
2. Double-click the desired definition name.
3. View the Host Identifier page, and modify information as needed (see Table 8–2):
   a. Name
   b. Description
   c. Operations: Add to or remove host name and port variations in the Operations list.

      Click + to add a new host name and port combination.

      Click an existing host name and click the Delete button to remove it.
4. Repeat step 3c as needed to add or remove variations.
5. Click Apply to submit the changes (or close the page without applying changes).
6. Dismiss the Confirmation window, and close the page when you finish.

8.4.7 Deleting a Host Identifier Definition

Users with valid OAM Administrator credentials can use the following procedure to delete an entire host identifier definition. A validation error occurs if you attempt to delete the host identifier that is being used in a resource.

Prerequisites

Each resource in an application domain is associated with a specific host identifier. If you intend to delete a host identifier you must first modify any resource definitions in an application domain that uses this host identifier

> **See Also:** "Viewing or Editing a Host Identifier Definition" on page 8-11 if you want to remove a single host identifier from an existing definition.

To delete a Host Identifier

1. From the Policy Configuration tab, navigation tree, expand the:
   a. Shared Components node
   b. Host Identifiers node
2. Optional: In the list, double-click the desired name to view the definition, and then close the page.
3. In the navigation tree, click the desired definition name.
4. Click the Delete button in the tool bar, and confirm removal in the Confirmation window.
5. Check the navigation tree to confirm that the definition was removed.

8.5 Managing Authentication Modules

In Oracle Access Manager 11g, each authentication scheme requires an authentication module. This section describes the pre-configured authentication modules that are provided and describes how administrators can define a custom module. It is divided into the following topics:

- About Default Authentication Modules Pages
- Creating a New Authentication Module
- Viewing or Editing Authentication Modules
- Deleting an Authentication Module

8.5.1 About Default Authentication Modules Pages

In the Oracle Access Manager Administration Console, pre-configured authentication modules are organized with other system-level components under the System Configuration tab.

Only the following pre-configured authentication module types are allowed in an authentication scheme. However, you can create new modules of an existing type to use in authentication schemes. For more information, see:

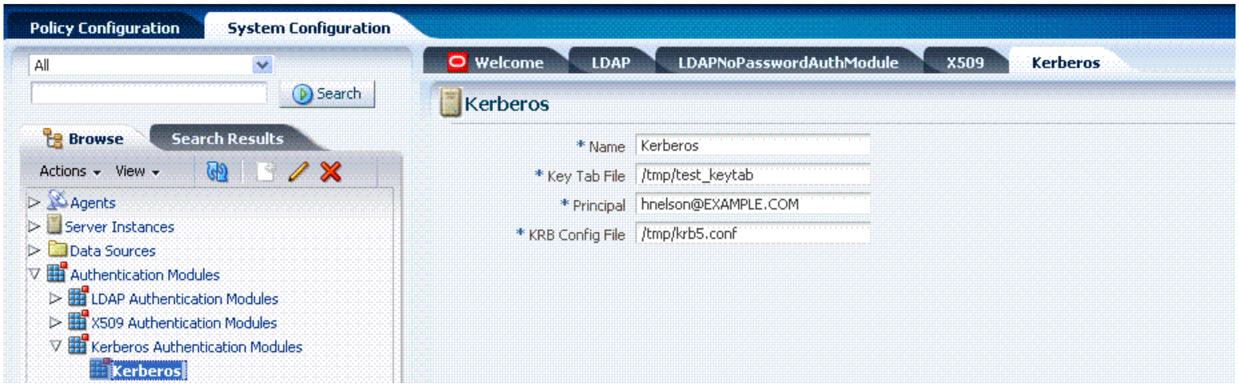
- Kerberos Authentication Module
- LDAP Authentication Modules
- X509 Authentication Module

> **See Also:**
>
> - "About Challenge Methods" on page 8-22
> - "About Authentication Modules" on page 8-25

8.5.1.1 Kerberos Authentication Module

The pre-configured Kerberos authentication module is illustrated in Figure 8–4. Additional details follow the figure.

Figure 8–4 Pre-configured Kerberos Authentication Module



* Name: Kerberos
* Key Tab File: /tmp/test_keytab
* Principal: hnelson@EXAMPLE.COM
* KRB Config File: /tmp/krb5.conf

Table 8–3 describes the definition of the Kerberos authentication module. You can use the existing, pre-configured Kerberos authentication module or create one of your own.

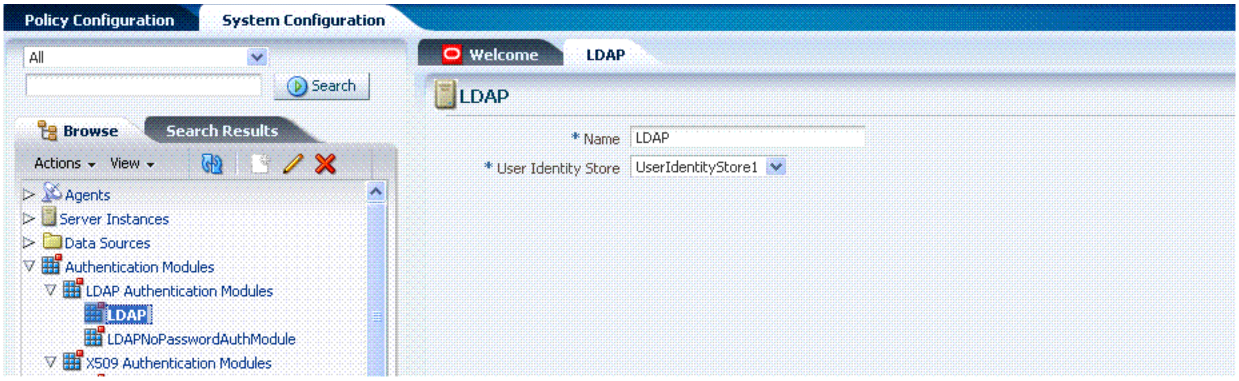
Table 8–3 Kerberos Authentication Module Definition

Element	Description
Name	The unique ID of this module, which can include upper and lower case alpha characters as well as numbers and spaces.
Key Tab File	The path to the encrypted, local, on-disk copy of the host's key, required to authenticate to the key distribution center (KDC). For example: /etc/krb5.keytab. The KDC authenticates the requesting user and confirms that the user is authorized for access to the requested service. If the authenticated user meets all prescribed conditions, the KDC issues a ticket permitting access based on a server key. The client receives the ticket and submits it to the appropriate server. The server can verify the submitted ticket and grant access to the user submitting it. The key tab file should be readable only by root, and should exist only on the machine's local disk. It should not be part of any backup, unless access to the backup data is secured as tightly as access to the machine's root password itself.
Principal	Identifies the HTTP host for the principal in the Kerberos database, which enables generation of a keytab for a host.
Krb Config File	Identifies the path to the configuration file that controls certain aspects of the Kerberos installation. A krb5.conf file must exist in the /etc directory on each UNIX node that is running Kerberos. krb5.conf contains configuration information required by the Kerberos V5 library (the default Kerberos realm and the location of the Kerberos key distribution centers for known realms).

8.5.1.2 LDAP Authentication Modules

The pre-configured LDAP authentication module is illustrated in Figure 8–4. Additional details follow the figure.

Figure 8–5 Pre-Configured LDAP Authentication Module



* Name: LDAP

* User Identity Store: UserIdentityStore1

Table 8–4 describes the elements in an LDAP authentication module. The same elements are also used in LDAPNoPasswordAuthnModule.

Table 8–4 LDAP Authentication Module Definition

Element	Description
Name	A unique name for this module.
User Identity Store	The primary user identity store that contains the user credentials required for authentication by this module. The LDAP store must be registered with OAM 11g to appear in this list. See Also: "Managing User Identity Store and OAM Administrator Registrations" on page 3-4.

8.5.1.3 X509 Authentication Module

Oracle Access Manager provides a pre-configured X509 authentication module as a default. Administrators can also create new X509 authentication modules. In cryptographic terms, X.509 is a standard for digital public key certificates used for single sign-on (SSO). X.509 specifies standard formats for public key certificates, certificate revocation lists, and attribute certificates among other things.

With X.509 digital certificates you can assume a strict hierarchical system of certificate authorities (CAs) issuing the certificates. In the X.509 system, a CA issues a certificate that binds a public key to a particular Distinguished Name, or to an Alternative Name such as an e-mail address or a DNS-entry.

The trusted root certificates of an enterprise can be distributed to all employees so that they can use the company PKI system. Certain Web browsers provide pre-installed root certificates to ensure that SSL certificates work immediately.

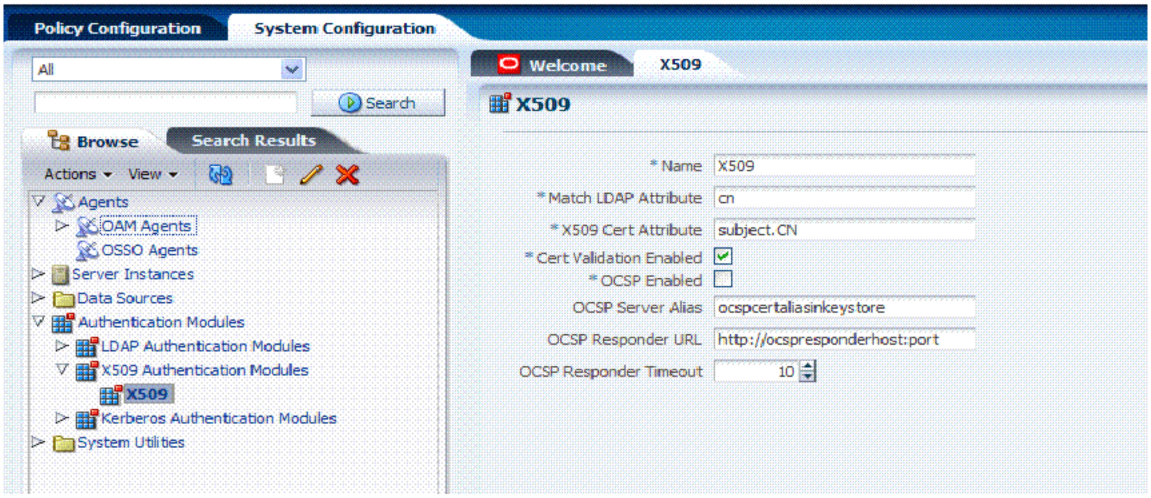
Oracle Access Manager uses the Online Certificate Status Protocol (OCSP) Internet protocol to maintain the security of a server and other network resources. OCSP is used for obtaining the revocation status of an X.509 digital certificate. OCSP specifies the communication syntax used between the server containing the certificate status and the client application that is informed of that status.

When a user attempts to access a server, OCSP sends a request for certificate status information. OCSP discloses to the requester that a particular network host used a particular certificate at a particular time. The server returns a response of "current", "expired," or "unknown." OCSP allows users with expired certificates a configurable grace period, during which they can access servers for the specified period before renewing.

OCSP messages are encoded in ASN.1 and are usually transmitted over HTTP. The request and response characteristic of OCSP has led to the term "OCSP responders" when referring to OCSP servers. With Oracle Access Manager, the computer hosting the OAM Administration Console is the OCSP responder.

An OCSP responder can return a signed response signifying that the certificate specified in the request is 'good', 'revoked' or 'unknown'. If OCSP cannot process the request, it can return an error code.

Figure 8–6 Pre-Configured X509 Authentication Module



X509

* Name: X509

* Match LDAP Attribute: cn

* X509 Cert Attribute: subject.CN

* Cert Validation Enabled: ☑

* OCSP Enabled: □

OCSP Server Alias: ocspcertaliasinkeystore

OCSP Responder URL: http://ocspresponderhost:port

OCSP Responder Timeout: 10

Table 8–5 describes the requirements of the X509 authentication module.

Table 8–5 X509 Authentication Module Definition

Element	**Description**
Name	Identifies this module definition with a unique name.
Match LDAP attribute	Defines the LDAP distinguished name attribute to be used. For example: cn.
X509 Cert Attribute	Defines the certificate attribute to be used to bind the public key. For example: SUBJECT.cn.
Cert Validation Enabled	Enables (or disables when not checked) X.509 Certificate validation.
OCSP Enabled	Enables (or disables when not checked) the Online Certificate Status Protocol. Note: OCSP Server Alias, OCSP Responder URL and OCSP Responder Timeout are required only when OCSP Enabled is selected.
OCSP Server Alias	An aliased name for the OSCSP Responder--a mapping between the aliased name and the actual instance name or the IP address of the OSCSP Responder instance.
OCSP Responder URL	Provides the URL of the Online Certificate Status Protocol responder.
OCSP Responder Timeout	Specifies the grace period for users with expired certificates, which enables them to access OAM Servers for a limited time before renewing the certificate.

8.5.2 Creating a New Authentication Module

Users with valid OAM Administrator credentials can use the following procedure to create a new authentication module of an existing type. You cannot duplicate a pre-configured module to use as a template.

Note: You cannot duplicate a pre-configured module to use as a template.

See Also: About Default Authentication Modules Pages

To create a new authentication module

1. From System Configuration tab, navigation tree, expand the Authentication Modules node.
2. From the navigation tree, click the desired module type.
3. Click the Create button in the tool bar.
4. Add details for the new authentication module:
   - Kerberos Module: See Table 8–3
   - LDAP Module: See Table 8–4
   - X509 Module: See Table 8–5
5. Click Apply to submit the new definition and close the Confirmation window (or close the page without applying changes).
6. Check the navigation tree to confirm the entry, and then close the page when you finish.
7. Add the authentication module to one or more authentication schemes, as described in "Creating an Authentication Scheme" on page 8-28.

8.5.3 Searching for a Specific Authentication Module

Users with valid OAM Administrator credentials can perform the following procedure to search for specific authentication module using the Administration Console.

Prerequisites

The authentication module must be registered in the Oracle Access Manager Administration Console.

See Also: About Default Authentication Modules Pages

To locate a specific authentication module

1. Activate the System Configuration tab.
2. From the search type list, choose one of the following module types to define your search, either:
   - LDAP Authentication Modules
   - X509 Authentication Modules
   - Kerberos Authentication Modules
3. In the text field, enter the exact name of the instance you want to find. For example:

   ```
   my_authn_module
   ```

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:

- **Edit:** Click the Edit button in the tool bar to display the configuration page.
- **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
- **Detach**: Click Detach in the tool bar to expand the table to a full page.
- **View**: Select a View menu item to alter the appearance of the results table.

6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

8.5.4 Viewing or Editing Authentication Modules

Users with valid OAM Administrator credentials can use the following procedure to modify an existing authentication module. This includes changing the name of an existing module as well as changing other attributes.

Prerequisites

Modify each authentication scheme that references the module you will change, to use another authentication module.

See Also: About Default Authentication Modules Pages

To view or edit an authentication module

1. Change to another authentication module in each authentication scheme that references this module.
2. From the System Configuration tab, navigation tree, expand the:
   a. Authentication Modules node
   b. Expand the module type node
3. Double-click the desired module name to display the configuration.
4. Optional: Close the page if you were simply viewing it.
5. On the Authentication Modules page, modify information as needed:
   - Kerberos Module: See Table 8–3
   - LDAP Module: See Table 8–4
   - X509 Module: See Table 8–5
6. Click Apply to submit the changes and close the Confirmation window (or close the page without applying changes).
7. Add the updated authentication module to authentication schemes, as described in "Creating an Authentication Scheme" on page 8-28.

8.5.5 Deleting an Authentication Module

Users with valid OAM Administrator credentials can use the following procedure to delete an authentication module.

Prerequisites

In each authentication scheme that references this module, specify another authentication module.

To delete an authentication module

1. In each authentication scheme that references this module, specify another authentication module.
2. From the System Configuration tab, navigation tree, expand the:
   a. Authentication Modules node
   b. Expand the module type node
3. Optional: Double-click the module name to display the configuration and then close the window.
4. Click the desired module name and then click the Delete button.
5. Confirm removal (or dismiss the confirmation window to retain the module).

8.6 Managing Authentication Schemes

This section is divided into the following topics:

- About the Authentication Schemes Page
- Creating an Authentication Scheme
- Viewing or Editing a Authentication Scheme
- Searching for a Authentication Scheme
- Deleting an Authentication Scheme

8.6.1 About the Authentication Schemes Page

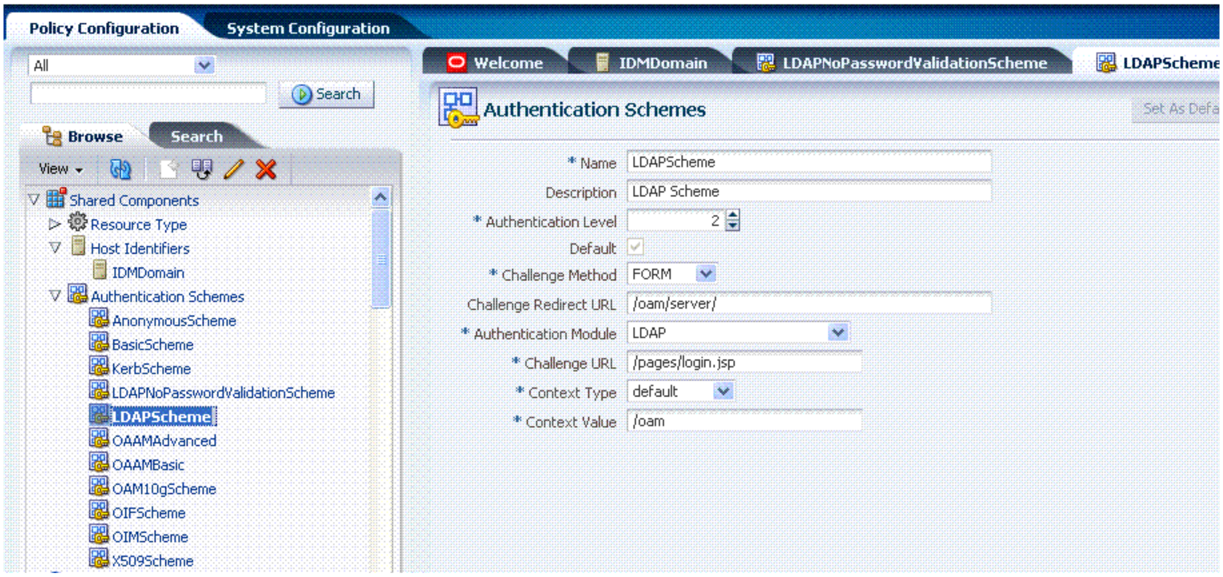
Access to a resource or group of resources can be governed by a single authentication process known as an authentication scheme. An authentication scheme is a named component that defines the challenge mechanism required to authenticate a user. Each authentication scheme must also included a defined authentication module.

When you register a partner (either using the Administration Console or the remote registration tool), the application domain that is created is seeded with a policy that uses the authentication scheme that is set as the default scheme. You can choose any of the existing authentication schemes as the default for use during policy creation.

You can also create a new authentication scheme, copy an existing definition to use as a template, modify a definition, or delete the definition. The copy uses a default name that is based on the original. For example, if you copy the scheme named *KerbScheme*, the copy is named *copyofKerbScheme*.

All authentication schemes include the same elements with differing values. Figure 8–7 shows the default LDAPScheme page as an example. The Authentication Schemes navigation tree lists other default schemes that are delivered.

Figure 8–7 Default LDAPScheme Page



* Name	LDAPScheme
Description	LDAP Scheme
* Authentication Level	2
Default	☑
* Challenge Method	FORM
Challenge Redirect URL	/oam/server/
* Authentication Module	LDAP
* Challenge URL	/pages/login.jsp
* Context Type	default
* Context Value	/oam

Table 8–6 provides information about each of the elements and values in any authentication scheme. Use the Set as Default button to make this the default scheme.

Table 8–6 Authentication Scheme Definition

Element	Description
Name	The unique name for this scheme, which appears in the navigation tree. See Also:"Pre-configured Authentication Schemes" on page 8-21
Description	The optional description, up to 200 characters, that explains the use of this scheme.
Authentication Level	The trust level of the authentication scheme. This reflects the challenge method and degree of trust used to protect transport of credentials from the user. The trust level is expressed as an integer value between 0 (no trust) and 99 (highest level of trust). **Note**: After a user is authenticated for a resource at a specified level, the user is automatically authenticated for other resources in the same application domain or in different application domains, if the resources have the same or a lower trust level as the original resource. See Also: "About Multi-Level Authentication" on page 8-26.
Default	A non-editable box that is checked when the Set as Default button is clicked.
Challenge Method	One challenge method must be selected from those available: ▪ Form ▪ Basic (LDAP) ▪ X509 (Certificate) ▪ WNA (Windows Native Authentication) ▪ None ▪ DAP ▪ OAM10g See Also: "About Challenge Methods" on page 8-22
Challenge Redirect URL	URL of a server specified in the Challenge Redirect field if you want user requests to be redirected to another server for processing. See Also: "About Host Identifiers" on page 8-5.

Table 8–6 (Cont.) Authentication Scheme Definition

Element	Description
Authentication Module	The pre-configured authentication module to be used to challenge the user for credentials. ■ LDAP (under the LDAP Authentication Modules node) ■ X509 Module 1 (under the X509 Authentication Modules node) ■ Kerberos (under the Kerberos Authentication Modules node) See also "About Default Authentication Modules Pages" on page 8-12.
For schemes using Challenge Method FORM, X509, or DAP	Only Schemes with the Challenge Method of FORM, X509, or DAP include these additional elements. Other schemes use defaults that require no change.
Challenge URL	The URL the credential collector will redirect to for credential collection. For a FORM based, out of the box authentication scheme (LDAPScheme and LDAPNoPasswordValidationScheme), the default Challenge URL is "/pages/login.jsp". The context type and context values are used to build the final URL.

Context Type	Used to build the final URL for the credential collector based on the following possible values:
	■ default: The Context Value is used to construct the final URL to forward to for credential collection. For example: with a challenge URL of "/pages/login.jsp" and a context value of /oam, the server forwards to "/oam/pages/login.jsp" for credential collection.
	■ customWar: If a customized credential collector page "customlogin.jsp" is deployed in a WAR file (with context root, "custom") within the same domain, it should be used to collect credentials. Then set the following values to have server forward to the WEB application page "/custom/customlogin.jsp" to collect credentials:
	challenge_url = "/customlogin.jsp"
	contextType="customWar"
	contextValue="/contextroot of custom application"
	■ customHtml: A custom html credential collector page. This file can be placed in a location that is accessible to the application. Set the following values to have the server forward to the custom servlet provided to read the html file and render the page:
	challenge_url = "/CustomReadServlet"
	contextType="customHtml"
	contextValue="html file location"
	■ external: If the login page is external, the file can be placed in a location that is accessible to the application. Set the following values to have the server redirect to the challenge URL (the fully-qualified URL of the external credential collector page) for credential collection. The username and password are collected by the external form (HTML or jsp) and submitted to the OAM Server:
	challenge_url = "/http://host:port/externallogin.jsp"
	contextType="external"
	contextValue=Not applicable
	See Also: "About Custom Login Pages"

Context Value	Used to build the final URL for the credential collector. The default value is /oam.

About Custom Login Pages

Only Schemes with the Challenge Method of FORM, X509, or DAP include additional elements described at the end of Table 8–6. All custom login pages must meet the following requirements:

- Custom login pages require exactly two form fields (username and password). Oracle Access Manager supports authentication forms with two fields only.

- CustomWar and external context types, require logic within the custom login page to perform the following two tasks:
  - Send back the request ID the page received from the Oracle Access Manager server. For example: String reqId = request.getParameter("request_id"); <input type="hidden" name="request_id" value="<%=reqId%>">
  - Submit back to the OAM Server the end point, "/oam/server/auth_cred_submit". For example: <form action="/oam/server/auth_cred_submit"> or "http://oamserverhost:port/oam/server/auth_cred_submit".

For more information, see the following topics:

- Pre-configured Authentication Schemes
- About Challenge Methods
- About Authentication Modules
- About Multi-Level Authentication

8.6.1.1 Pre-configured Authentication Schemes

Table 8–7 identifies the pre-configured authentication schemes available with Oracle Access Manager 11g and some specific details of each.

Table 8–7 Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
AnonymousScheme	Authentication Level: 0 Challenge Method: None Authentication Module: AnonymousModule	Unprotects specific Oracle Access Manager URLs and allows users to access URLs without a challenge. Users are not challenged and do not need to enter their credentials. Note: Authentication Level 0 is for public pages. Oracle recommends that you do not use Level: 0 in a custom authentication scheme.
BasicScheme	Authentication Level: 1 Challenge Method: Basic Authentication Module: LDAP	Protects Oracle Access Manager-related resources (URLs) for most directory types. **Note**: Authentication Level 1 is only one step higher than 0 public pages. Oracle recommends that you do not use Level: 1 in a custom authentication scheme.
KerbScheme	Authentication Level: 2 Challenge Method: WNA Authentication Module: Kerberos	Protects Oracle Access Manager-related resources (URLs) for most directory types based on a Windows Native Authentication challenge method and valid WNA credentials in Active Director.y
LDAPNoPassword ValidationScheme	Authentication Level: 2 Challenge Method: Form Authentication Module: LDAPNoPasswordAuthModule **Note**: LDAPNoPasswordAuthModule is similar to the DAP (asserter) mechanism. **See Also** OAM10gScheme, later in this table.	Protects Oracle Access Manager-related resources (URLs) for most directory types based on a form challenge method. Used with the Identity Asserter for SSO when you have resources in a WebLogic Container. For details, see the *Oracle Fusion Middleware Application Security Guide*.
LDAPScheme	Authentication Level: 2 Challenge Method: Form Authentication Module: LDAP	Protects Oracle Access Manager-related resources (URLs) for most directory types based on a form challenge method.
OAAMAdvanced	Authentication Level: 2 Challenge Method: Form Authentication Module: LDAP	Protects OAAM-related resources with an external context type. This authentication scheme is used when complete integration with OAAM is required. A WebGate must front ending the partner.
OAAMBasic	Authentication Level: 2 Challenge Method: Form Authentication Module: LDAP	Protects OAAM-related resources with a default context type. This scheme should be used when basic integration with OAAM is required. Here, advanced features like OTP are not supported. This is more of an integration when mod_osso is used as the agent.

Table 8–7 (Cont.) Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
OAM10gScheme	Authentication Level: 2 Challenge Method: OAM10g Authentication Module: LDAPNoPasswordAuthModule **Note**: The challenge mechanism OAM10g is similar to that of DAP (asserter) mechanism. The OAM10g mechanism is used specifically for OAM10g coexistence with OSSO and should not be used with any other scheme. **See Also** LDAPNoPasswordValidationScheme, earlier in this table.	Facilitates integration and coexistence with OAM 10g. In the coexistence mode, OAM 10g is the authenticator and OAM 11g is the asserter. This scheme uses a new challenge mechanism: OAM10G.
OIFScheme	Authentication Level: 1 Challenge Method: DAP Authentication Module: DAP	This scheme delegates authentication to OIF, after which, OIF sends back a token that is asserted by the OAM Server. The Delegated Authentication Protocol (DAP) challenge method is used to delegate authentication to a third-party (OIF in this case).
OIMScheme	Authentication Level: 1 Challenge Method: Form Authentication Module: LDAP	Protects Oracle Identity Manager-related resources with a default context type. Note: When integrating OAM and OIM, OAM downgrades the user’s authentication level when any of the following is detected: password expiry forced password change challenge setup not done This enables the user to access the pages only after performing necessary operations in the identity management (OIM) page to which the user is redirected. At Level 1, only public and OIM pages for the required operations can be accessed. **Note**: Authentication Level 1 is only one step higher than 0 public pages. Oracle recommends that you do not use Level: 1 in a custom authentication scheme.
X509Scheme	Authentication Level: 2 Challenge Method: X509 Authentication Module: X509	This authentication scheme is a certificate-based user identification method. To use this method, a certificate must be installed on the user’s browser and the Web server must be SSL-enabled.

8.6.1.2 About Challenge Methods

Authentication involves determining what credentials a user must supply when requesting access to a resource, gathering credentials over HTTP, and returning an HTTP response that is based on the results of credential validation. Oracle Access Manager 11g provides the following credential challenge methods for use in an authentication scheme:

- Form
- Basic
- X509
- WNA
- None
- DAP
- OAM10g

Form

This authentication challenge uses an HTML form with one or more text input fields for user credentials. In a typical form-based challenge, users enter a user name and password in two text boxes on the form. The most common credential choices are user name and password; however, you can use any user attributes: for example, user name, password, and domain.

A Submit button posts the content of the form. When the user clicks the Submit button, the form data is posted to the Web server. OAM and OSSO Agents intercept and process the form data. Upon validation of the user credentials collected in the form, the user is authenticated.

> **Note:** This challenge method relies on an LDAP Authentication Module and the user identity store associated with that module.

You might want to use form-based authentication challenge for reasons such as:

- A consistent user experience: Using form-based login and a standardized logout means that the user experience for login and logout features will be consistent across browsers.
- A Custom Form: You can apply your organization's look and feel in the authentication process.

  For example, a custom form can include a company logo and a welcome message instead of the standard user name and password window used in Basic authentication.
- Additional Information: You can gather additional information at the time of login.
- Additional Functionality: You can provide additional functionality with the login procedure, such as a link to a page for lost password management.

Basic

This built-in Web server challenge mechanism requires a user to enter her login ID and password. The credentials supplied are compared to the user’s definition in the LDAP directory server.

> **Note:** A Basic challenge relies on an LDAP Authentication Module and the user identity store associated with that module.

X509

With the X509 certificate challenge method, a user's browser must supply an X.509 digital certificate over SSL to the OAM Server through the Agent to perform authentication.

> **Note:** X509 is the challenge method for the X509Scheme. The user's organization can determine how to obtain a certificate.

The X.509 client certificate must be verified against the trusted CAs in the key store used by OAM Proxy and OAM Servers to ensure the validity of X.509 Client certificate for authentication.

The following attributes of the X.509 certificate can be validated against the user identity store associated with OAM 11g:

- SubjectDN
- SubjectUniqueID
- Mail
- CN

To acquire the user entry, the X509 Authentication Module takes the attribute name of the X.509 certificate to be validated and the LDAP attribute against which the search will be launched. The expected result is the single user entry matching the criteria. If the search returns no user entry, or more than one entry, authentication fails. Authentication scheme parameters are located in oam-policy.xml.

> **Note:** For X509 authentication, Administrators must configure the Oracle HTTP Server as a reverse proxy (or a server with the wl-proxy plug-in). The Oracle HTTP Server must be configured in two way SSL Mode to acquire X.509 certificate for authentication. Oracle HTTP Server can also be configured for CRL verification.

The online certificate status protocol (OCSP) capabilities are also provided. Any certificate passed for X.509 certificate-based authentication is validated using an OCSP request. Administrators can configure the system to communicate with one or more OCSP servers to retrieve the certificate status.

The X509 Authentication Module configuration for the OCSP responder URL indicates whether OCSP validation is to be done. The value, if specified, indicates the URL for validation of the X.509 client certificate using OCSP. No value indicates no OCSP validation.

WNA

Uses Windows Native Authentication with Active Directory, and the Kerberos Authentication Module.

> **Note:** The KerbScheme relies on the WNA challenge method and Kerberos Authentication Module.

> **See Also:** *Oracle Fusion Middleware Integration Guide for Oracle Access Manager*for details about integration Windows Native Authentication

None

The challenge method of None means that users are not challenged and do not need to enter their credentials. This is used in the AnonymousScheme authentication scheme, which allows users to access Oracle Access Manager-specific URLs that you do not want to protect.

DAP

The Delegated Authentication Protocol (DAP) challenge method is new and required for OIFScheme (Oracle Identity Federation integration) with the DAP authentication module and external context type (Table 8–6). The DAP challenge mechanism indicates that OAM does an assertion of the token that it receives, which differs from the standard challenge "FORM" mechanism with the external option.

DAPModule is a new assertion module, though it is specialized for this one application and does not appear in the list of Authentication Modules in the OAM Administration Console. This integration replaces OSSO 10g with OAM11g, with no changes from the OIF side

The DAP challenge mechanism delegates authentication to a third party (OIF in this case). The challenge_url points to the OIF Server URL. When a resource is protected by this scheme, the OAM Server redirects to the OIF Server URL for credential collection. OAM Server does not perform the credential collection or validation in this case. OIF collects the credentials, authenticates the user against its identity store and returns an assertion token to the OAM Server consisting of the username. OAM receives and decrypts this token, checks whether the user is a valid user in the primary identity store for OAM. If the user is valid, OAM gives access to the resource.

The DAPToken is encrypted and decrypted with a key that is shared between OAM and OIF. The DAPToken is built from the OAM side.

The OIF Administration EM Console provides a way to generate the keystore containing the encryption keys that will be used to secure communications between the OAM 11g and OIF. OAM provides a WLST command (registerOIFDAPPartner), that takes the keystore location generated by the OIF store and retrieves the keys and stores it on the OAM side.

OAM10g

This challenge method is required for OAM10gScheme with the LDAPNoPasswordAuthModule to facilitate trust when you have OAM 10g protecting a domain that also includes an OSSO 10g integrated classic application (Portal, Disco, and so on). This new mechanism is created for OAM 10g coexistence.

OSSO10g is protected with OAM10g through WebGate, so that OAM10g always acts as the authentication/authorization provider.

Facilitating Integration: The OSSO 10g integrated classic applications can be upgraded to OAM 11g, which then acts only as an asserter. OAM 11g creates the tokens that mod_osso can consume so that access can be provided to these applications. The mod_osso applications are protected by the new "OAM10gScheme". There is a WebGate front ending the OAM 11g Server and configured against the OAM 10g Access Server.

Setup: When the resource is accessed, WebGate intercepts the request and sends it to the OAM 10g Access Server for authentication. OAM 10g collects the credentials, validates it against its identity store, and sets the username as a header variable (OAM_REMOTE_USER). The request now goes to the OAM 11g Server which uses the OAM10gScheme to locate the username in the header variable. OAM 11g retrieves the header variable and asserts the presence of the user against the primary identity store. If present, the required cookies (OAM_ID) are generated and redirected to the resource.

8.6.1.3 About Authentication Modules

In Oracle Access Manager 11g, each authentication scheme requires an authentication module. Administrators can create a new authentication module of an existing type. However, several default authentication modules are available for immediate use:

- LDAP: This module matches the credentials (username and password) of the user who requests a resource to a user definition stored in an LDAP directory server. An LDAP module is required for Basic and Form challenge methods.

- X509: Similar to LDAP with additional properties that indicate which attribute of the client's X.509 certificate should be validated against the user attribute in LDAP.
- Kerberos Module: A credential mapping module that matches the credentials (username and password) of the user who requests a resource to the encrypted "ticket".

See Also: "Managing Authentication Modules" on page 8-12

8.6.1.4 About Multi-Level Authentication

Every authentication scheme requires a strength level. The lower this number, the less stringent the scheme. A higher level number indicates a more secure authentication mechanism:

- LDAPScheme authLevel=1
- KerbScheme authLevel=2

Note: Multi-level authentication does not affect, negate, or alter X.509 certificate authentication.

SSO capability enables users to access more than one protected resource or application with a single sign in. After a successful user authentication at a specific level, the user can access one or more resources protected by one or more application domains. However, the authentication schemes used by the application domains must be at the same level (or lower). When a user accesses a resource protected with an authentication level that is greater than the level of his current SSO token, he is re-authenticated. In the step-up case, the user maintains his current level of access even if failing the challenge presented for the higher level. This is "additional authentication".

Note: A user who is authenticated to access resources at level 3, is eligible to access resources protected at levels less than or equal to 3. However, if the user is authenticated to access resources at level 2 and then attempts to access resources protected by level 3, the user is asked to re-authenticate (this is known as step-up authentication).

Oracle Access Manager 11g policies allow different resources of the same application to be protected with different authentication levels. However, the mod_osso plug-in does not support two resources on the same application with a different trust level.

Note: mod_osso delegates authentication to OAM. Oracle recommends that mod_osso-protected resources be protected with OAM authentication levels.

In such cases, the application must enforce the Level and send the Dynamic Directive to mod_osso for re-authentication. On receiving the Dynamic Directive, mod_osso will redirect to Oracle Access Manager for re-authentication at the appropriate level.

For more information, see:

- About Agents and Multi-Level Authentication
- Detection of Insufficient Authentication Level by OAM Agent

- Request Flow for Multi-Level Authentication with OSSO Agent (mod_osso 10g)

8.6.1.4.1 About Agents and Multi-Level Authentication Registered agents detect the authentication level as follows:

- mod_osso detects the authentication level from dynamic directives, as described in "Request Flow for Multi-Level Authentication with OSSO Agent (mod_osso 10g)" on page 8-27
- OAM Agents receive an insufficient level error message from the OAM Server, as described in "Detection of Insufficient Authentication Level by OAM Agent" on page 8-27

Both agent types redirect the user the OAM server to authenticate again. The challenge is presented according to the level of the authentication scheme configured in the policy for the resource.

8.6.1.4.2 Detection of Insufficient Authentication Level by OAM Agent When the user requests a resource that is protected with a higher level authentication scheme, the following process occurs.

Process overview: OAM Agent detects insufficient user session level

1. The OAM Agent sends the request to the OAM Proxy to obtain the scheme details for the protected resource.
2. The OAM Agent sends the request for session information to the OAM Proxy.
3. The OAM Proxy returns details of the ObSSOCookie, including the authenticated level of the ObSSOCookie.
4. The OAM Agent compares the level of ObSSOCookie with that of the authentication scheme.
   - If insufficient, the agent invokes the authentication process again.
   - If sufficient, the access is granted access.

No check of the authentication level is made on the server side.

8.6.1.4.3 Request Flow for Multi-Level Authentication with OSSO Agent (mod_osso 10g) In contrast to OAM Agents, all the resources protected by mod_osso on a host (or virtual host) are protected at the same level.

With mod_osso, multi-level authentication applies when user is already authenticated using one mod_osso host (or virtual host) at Level 2 and then tries to access another mod_osso protected host (or virtual host) at level 3.

Process overview: OSSO Agent multi-level authentication flow

1. The user tries to access a resource protected by mod_osso on host1 at level 2.
2. The OSSO Agent sends the request to the OAM Proxy to obtain the authentication scheme details for the protected resource.
3. The OAM_ID cookie for SSO Server and a host based cookie "HOST_port" for host1 are set and contain authentication level information.
4. After authentication, the user tries to access a resource on host2 that is protected with a higher level of authentication.
5. The user is redirected to the OAM Server for authentication because this is the first time accessing host2.

6. The OAM Server (OSSO Proxy) receives the OAM_ID cookie which has an insufficient level to access the resource on host2.
   - If the level is insufficient, the OAM Server (OSSO Proxy) triggers re-authentication.
   - If the level is sufficient, the access is granted access.

8.6.2 Creating an Authentication Scheme

Users with valid OAM Administrator credentials can use the following procedure to add a new authentication scheme for use in an application domain.

Prerequisites

The authentication module must be defined and ready to use.

> **See Also:** "About the Authentication Schemes Page" on page 8-18

To create an authentication scheme

1. From the Policy Configuration tab, navigation tree, expand the Shared Components node.
2. Click the Authentication Schemes node, then click the Create button in the tool bar.
3. Fill in the fresh Authentication Scheme page, as described in Table 8–6:
   a. Name
   b. Description
   c. Authentication Level
   d. Default
   e. Challenge Method
   f. Challenge Redirect
   g. Authentication Module
4. Click Apply to submit the new scheme (or close the page without applying changes).
5. Dismiss the Confirmation window.
6. Optional: Click the Set as Default button to automatically use this with new application domains, then close the Confirmation window.
7. In the navigation tree, confirm the new scheme is listed, and then close the page

8.6.3 Searching for a Authentication Scheme

Users with valid OAM Administrator credentials can perform the following task to search for a specific authentication scheme.

> **See Also:** "About Search Controls" on page 2-24

To search for an authentication scheme

1. Activate the Policy Configuration tab.
2. From the search type list, choose Authentication Schemes to define your search.

3. In the text field, enter the exact name of the instance you want to find. For example:

   ```
   my_AuthnScheme
   ```

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.
6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

8.6.4 Viewing or Editing a Authentication Scheme

Users with valid OAM Administrator credentials can use the following procedure to view or modify an existing authentication scheme.

Prerequisites

Review any application domain using this authentication scheme and specify a different scheme.

See Also: "About the Authentication Schemes Page" on page 8-18

To view or modify an authentication scheme

1. From the Policy Configuration tab, navigation tree, expand the:
   a. Shared Components node
   b. Authentication Schemes node
2. Double-click the name of the scheme to change.
3. On the Authentication Scheme page, modify values as needed (Table 8–6).
4. Click Apply to submit the changes (or close the page without applying changes).
5. Dismiss the Confirmation window.
6. Optional: Click the Set as Default button to automatically use this with new application domains, then close the Confirmation window.
7. Close the page when you finish.

8.6.5 Deleting an Authentication Scheme

Users with valid OAM Administrator credentials can use the following procedure to delete an existing authentication scheme.

Prerequisites

Review any application domain using this authentication scheme and specify a different scheme.

See Also: "About the Authentication Schemes Page" on page 8-18

To delete an authentication scheme

1. From the Policy Configuration tab, navigation tree, expand the:
   a. Shared Components node
   b. Authentication Schemes node
2. Optional: Double-click the name of the scheme to confirm it is the one to delete, then close the page.
3. In the navigation tree, click the name of the scheme and then click the Delete button in the tool bar.
4. Confirm removal (or dismiss the Confirmation window).
5. In the navigation tree, confirm the instance has been removed.

9

Managing Policies to Protect Resources and Enable SSO

This chapter describes how to create and manage policies, and identify the resources to be governed by these policies. This chapter focuses on using the Oracle Access Manager 11g Administration Console for tasks and includes the following topics:

- Prerequisites
- Introduction to Application Domain Creation
- Anatomy of an Application Domain and Policies
- Managing Application Domains using the Administration Console
- Adding and Managing Resource Definitions for Use in Policies
- Defining Authentication Policies for Specific Resources
- Defining Authorization Policies for Specific Resources
- Introduction to Policy Responses for SSO
- Adding and Managing Policy Responses for SSO
- Introduction to Authorization Constraints
- Defining Authorization Policy Constraints
- Managing the Common SSO Engine
- Validating Authentication and Authorization in an Application Domain

9.1 Prerequisites

Review Chapter 7 for an introduction to the OAM policy model and single sign-on.

System level requirements for tasks in this chapter include the following:

- Oracle Access Manager 11g should be operational
- Users and groups who can access a protected resource should already be created in the User Identity Store associated with OAM 11g.
- Policy-enforcement Agents should be registered as described in Chapter 5.
- Shared components for use in any application domain should be defined, as described in Chapter 8.

9.2 Introduction to Application Domain Creation

This section provides the following topics:

- About Automatic Application Domain Creation
- About Manually Creating Application Domains

9.2.1 About Automatic Application Domain Creation

When you register a policy-enforcement Agent with OAM 11g, you can choose to have policies created automatically. An application domain, and host identifier, are created automatically based on details specified for the Agent. The domain is seeded with a default resource, and with default authentication and authorization policies.

Each application domain is a collection of resources that represents a singular application on a particular host. Configurable authentication and authorization policies allow or deny access to the resources in the domain.

During Agent registration, it is presumed that the Agent is on the same Web Server as the application it protects. However, the Agent can be on a proxy Web server and the application can be on a different host.

> **Note:** IDMDomainAgent is a pre-registered Java Agent filter that provides an application domain of the same name to protect Oracle Fusion Middleware console and other

For more information, see "Anatomy of an Application Domain and Policies" on page 9-3.

9.2.2 About Manually Creating Application Domains

When a new application is placed behind an existing agent, the administrator must decide if it should be protected by a separate application domain and policies or an existing application domain and policies.

Administrators can define different application domains for resources even when these reside on the same Web server and are closely tied to each other in one way or another. For example, an administrator can create a single application domain for a financial application and an accounts receivable application or have a different application domain for each one.

> **Note:** Before adding a new resource to an application domain, consider whether the resource belongs to an existing application and its policies or if it should be defined as a new application with specific policies of its own.

The following task overview outlines the procedures that must be performed to manually create a fresh application domain.

> **Note:** Tasks 3 through 6 enable administrators to manage an existing application domain.

Task overview: Creating an application domain manually

1. Review "Anatomy of an Application Domain and Policies" on page 9-3.
2. Start the application domain registration as described in "Creating a Fresh Application Domain" on page 9-9.
3. Add one or more resources to the domain, as described in "Adding and Managing Resource Definitions for Use in Policies" on page 9-11.
4. Configure one or more authentication policies to the domain, as described in "Defining Authentication Policies for Specific Resources" on page 9-19.
5. Configure one or more authorization policies to the domain, as described in "Defining Authorization Policies for Specific Resources" on page 9-24.
6. Define one or more SSO Responses to your policies, as described in "Adding and Managing Policy Responses for SSO" on page 9-28.
7. Define one or more authorization constraints as described in "Defining Authorization Policy Constraints" on page 9-42.
8. Configure SSO Engine settings as described in "Managing the Common SSO Engine" on page 9-45.
9. Validate the domain's operation, as described in "Validating Authentication and Authorization in an Application Domain" on page 9-47.

9.3 Anatomy of an Application Domain and Policies

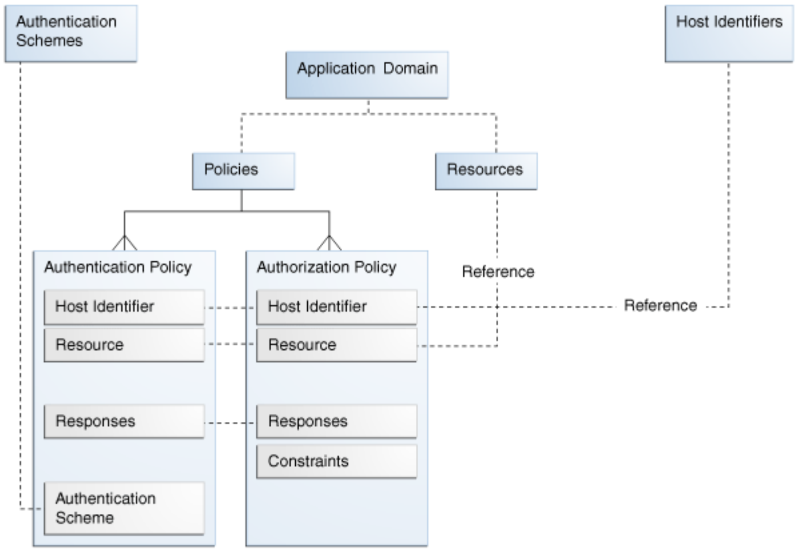
OAM 11g default behavior is to deny access when a resource is not protected by a policy that explicitly allows access.

> **Note:** OAM 10g default behavior allowed access when a resource was not protected by a rule or policy that explicitly denied access to limit the number of WebGate queries to the Access Server.

With OAM 11g, administrators control who can access resources by defining policies that discriminate between authenticated users who are authorized to access a particular resource and those who are not authorized. Each application domain can be made to contain policy elements related to an entire application deployment, a particular tier of the deployment, or a single host. Application domains do not have any hierarchical relationship to one another.

Figure 9–1 illustrates Application Domain-specific components of the Oracle Access Manager 11g Policy Model.

Figure 9–1 Application-Specific Components of the OAM Policy Model



The application domain that is automatically created during Agent registration, is named after the Agent and is seeded with default resources and default policies (authentication and authorization), based on the Agent name. Initially, all resources are protected by the default authentication and authorization policies.

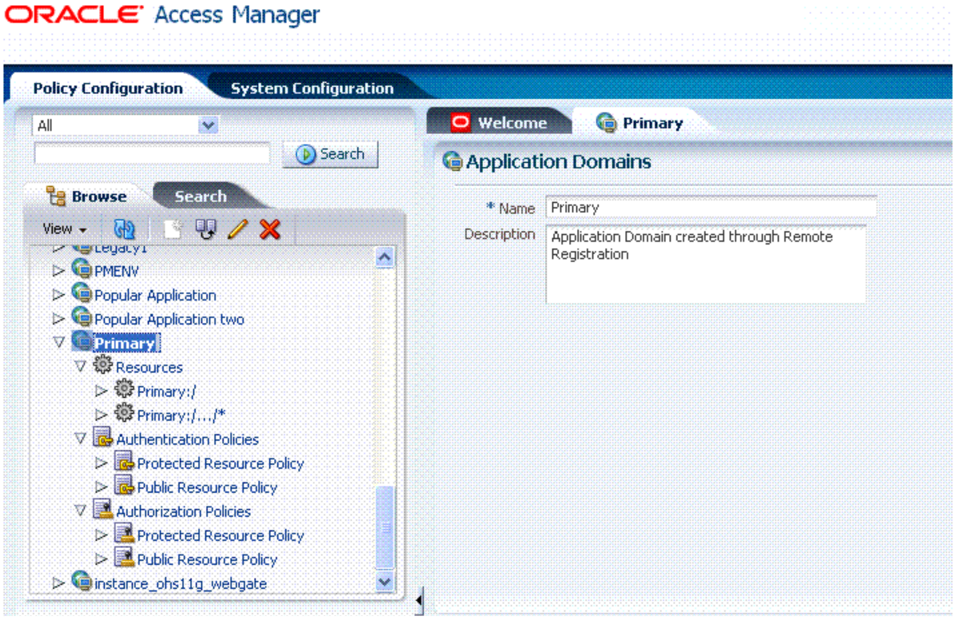
Whether you register an Agent using the Administration Console or the remote registration tool, the elements of an application domain are the same, as described in following topics:

- Application Domain General Details
- Default Resource Definition in a Generated Application Domain
- Default Authentication Policies in a Generated Application Domain
- Default Authorization Policies in a Generated Application Domain

9.3.1 Application Domain General Details

Figure 9–2 shows an application domain name highlighted in the navigation tree under the Application Domains node, and the Application Domain page with a name and description.

Figure 9–2 Application Domain Generated using the Administration Console



* Name: Primary

Description: Application Domain created through Remote Registration

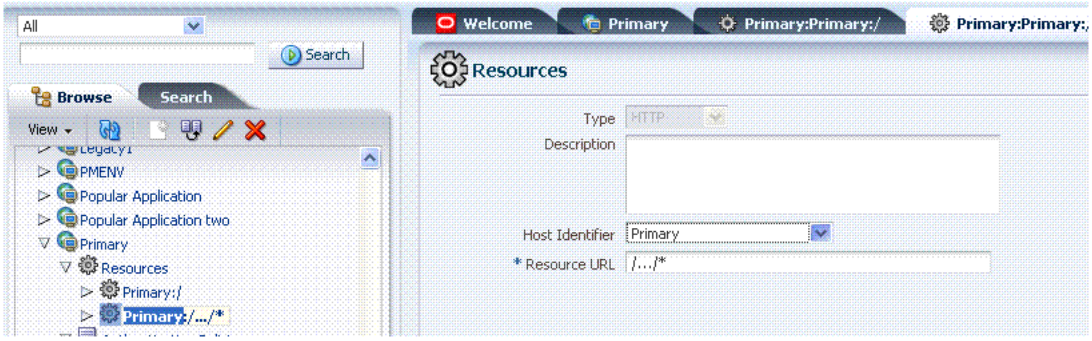
Beneath the application domain name in the navigation tree are the default resource definitions, and policies.

Administrators can modify the application domain to add more resources, and define policies, responses, and authorization constraints.

9.3.2 Default Resource Definition in a Generated Application Domain

Figure 9–3 illustrates the default resource definition in the generated application domain. The Host Identifier matches the Agent name that was specified during registration. The default Resource Type is HTTP; the default Resource URL is /.../*.

Figure 9–3 Default Resource Definition in the Application Domain



Type: HTTP

Description:

Host Identifier: Primary

* Resource URL: /.../*

See Also: "Adding and Managing Resource Definitions for Use in Policies" on page 9-11

9.3.3 Default Authentication Policies in a Generated Application Domain

Each resource can be protected by only a single authentication policy. When an administrator creates an application domain manually she must also manually create all policies. However, when the application is generated automatically the following authentication policies are automatically generated:

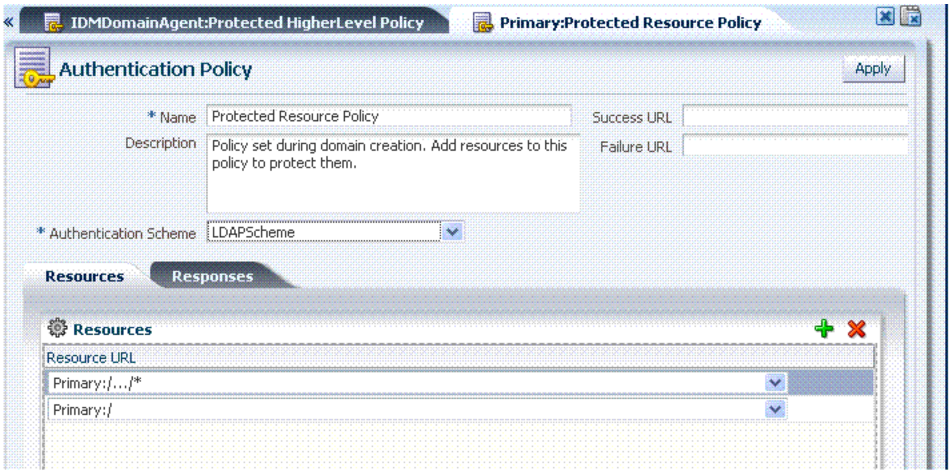
- Protected Resource Policy
- Public Resource Policy

The **Protected Resource Policy** is shown in Figure 9–4. The description explains "`Policy set during domain creation. Add resources to this policy to protect them.`" This default policy uses the default authentication scheme (in this case, it is the LDAPScheme authentication scheme). Protected Resources are identified on the Resources tab as *`HostIdentifier`*`/.../*`. Administrators can change the authentication scheme, specify Success and Failure URLs, add other resources, and define authentication Responses.

Authentication policies are local, which means that each policy applies only to the resources specified for the policy. A policy cannot be derived or applied to any other resource.

> **Note:** Initially, all resources are protected. Success and Failure URLs and Responses must be added manually; no default values are supplied.

Figure 9–4 Default Authentication Policy for Protected Resources


IDMDomainAgent:Protected HigherLevel Policy
Primary:Protected Resource Policy
Authentication Policy
Apply
Resources
Responses

* Name: Protected Resource Policy

Description: Policy set during domain creation. Add resources to this policy to protect them.

Success URL:

Failure URL:

* Authentication Scheme: LDAPScheme

Resources

Resource URL
Primary:/.../*
Primary:/

Authentication, **Public Resource Policy**: A second authentication policy is also created, which uses AnonymousScheme as its default authentication scheme. The description tells administrators "`Policy set during domain creation. Add resources to this policy to allow anyone access.`" This Public Resource Policy does not initially protect any Resources.

See Also: "Introduction to Policy Responses for SSO" on page 9-28

9.3.4 Default Authorization Policies in a Generated Application Domain

Each resource can be protected by only a single authorization policy. Administrators who manually create an application domain must manually create all policies. However, when the application is generated automatically the following authorization policies are automatically generated:

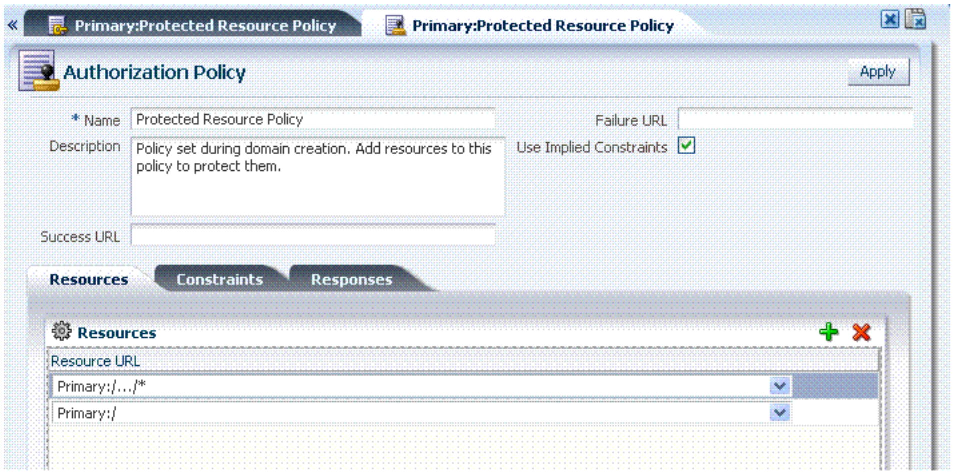
- Protected Resource Policy

- Public Resource Policy

> **Note:** Initially, all resources are protected and access is denied. Success and Failure URLs and Constraints and Responses must be added manually; no default values are supplied.

The Authorization, Protected Resource Policy is shown Figure 9–5. The description explains "`Policy set during domain creation. Add resources to this policy to protect them.`" Protected Resources are identified on the Resources tab as *`HostIdentifier`*`/.../*`. Administrators can specify Success and Failure URLs, add other resources, and define authorization Constraints and Responses.

Figure 9–5 Default Authorization Policy for Protecting Resources


Primary:Protected Resource Policy
Primary:Protected Resource Policy
Authorization Policy
Apply
Resources
Constraints
Responses

* Name: Protected Resource Policy

Description: Policy set during domain creation. Add resources to this policy to protect them.

Success URL:

Failure URL:

Use Implied Constraints ☑

Resources

Resource URL
Primary:/.../*
Primary:/

Authorization, **Public Resource Policy**: A second authorization policy is also created. The description explains "`Policy set during domain creation. Add resources to this policy to allow anyone access.`" This Public Resource Policy does not initially protect any Resources.

See Also:

- "Introduction to Policy Responses for SSO" on page 9-28
- "Introduction to Authorization Constraints" on page 9-35

9.4 Managing Application Domains using the Administration Console

This section provides the following topics:

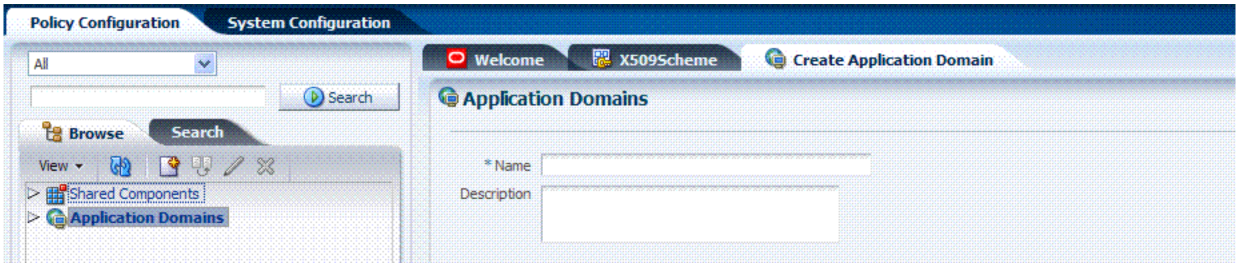
- About the Application Domains Page
- Creating a Fresh Application Domain
- Searching for an Application Domain
- Viewing or Editing an Application Domain
- Deleting an Application Domain and Its Content

9.4.1 About the Application Domains Page

Managing an application domain involves adding, modifying, copying, or deleting general and resource-related settings as well authentication and authorization policies. The copy uses a default name that is based on the original. For example, if you copy an Application Domain named *AppDom3*, the copy is named *copy of AppDom3*. All other settings in the copied domain are retained in the copy.

When creating or editing an application domain using the Administration Console, several pages are involved. Initially, you add general details (name and optional description) on the form show in Figure 9–6.

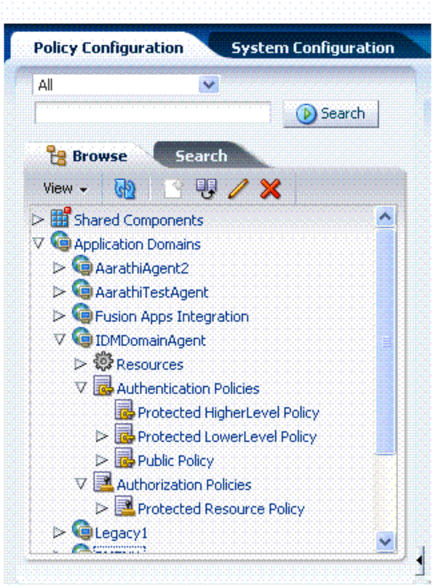
Figure 9–6 Fresh Application Domains General Page



Each application domain must have a unique name that matches the agent name. After applying the name and optional description for the new Application Domain, it is created and the name is added to the Application Domains node in the navigation tree under the Policy Configuration tab. The list includes all application domain names as a flat list of containers.

You can expand the Application Domains node to reveal all domains, including the new one. Figure 9–7 illustrates the Application Domains navigation tree and the IDMDomainAgent-related containers for resources and authentication and authorization policies.

Figure 9–7 Application Domains Navigation Tree



When you select a domain name in the navigation tree, you can add, modify, or delete individual elements, as described in topics elsewhere in this chapter.

9.4.2 Creating a Fresh Application Domain

Users with valid OAM Administrator credentials can perform the following task to manually create an application domain using the OAM Administration Console.

You can protect multiple applications using the same Agent by manually creating the application domain and manually adding resources and policies.

Prerequisites

Decide whether you need a fresh application domain or if you can add resources to an existing application domain.

> **Note:** You can duplicate an existing domain to use as a template and edit the copy to define unique identifiers (resource name and resource URLs), as described in "Viewing or Editing an Application Domain" on page 9-10.

See Also: "Introduction to Application Domain Creation" on page 9-2

To create a fresh application domain

1. From the Policy Configuration tab navigation tree, click the Application Domains node, and then click the Create command button in the tool bar.

   Alternatively: From the Welcome page, Policies panel, click the Add Application Domain link to open a fresh page.

2. On the fresh Application Domains page, add a unique name and an optional description for this domain, then click Apply and close the Confirmation window.
3. Click the Refresh button in the tool bar and confirm that the new application domain appears in the navigation tree.
4. In the navigation tree, expand the new Application Domain to view and manage the following nodes:
   - **Resources:** See "Adding and Managing Resource Definitions for Use in Policies" on page 9-11.
   - **Authentication Policies**: See "Defining Authentication Policies for Specific Resources" on page 9-19.
   - **Authorization Policies**: See "Defining Authorization Policies for Specific Resources" on page 9-24.

9.4.3 Searching for an Application Domain

Users with valid OAM Administrator credentials can use the following procedure to search for a specific application domain.

See Also: "About Search Controls" on page 2-24

To search for an application domain

1. Activate the Policy Configuration tab.

2. From the search type list, choose Application Domain to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   my_App_Domain_Name

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.
6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

9.4.4 Viewing or Editing an Application Domain

Users with valid OAM Administrator credentials can perform the following task to view or modify an application domain (including its resources, policies, constraints, and responses) using the OAM Administration Console.

> **Note:** You can duplicate an existing domain to use as a template and edit the copy to define unique identifiers (resource name and resource URLs).

Oracle recommends that you consider grouping similar applications into the same application domain. While editing the application domain, be aware that different applications are using the same domain. Editing the description and domain name are supported.

To view or modify an application domain and its content

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
   Desired Domain

2. Expand each of the following nodes to add, view, modify, or delete specific:
   - **Resources:** See "Adding and Managing Resource Definitions for Use in Policies" on page 9-11.
   - **Authentication Policies**: See "Defining Authentication Policies for Specific Resources" on page 9-19.
   - **Authorization Policies**: See "Defining Authorization Policies for Specific Resources" on page 9-24.
3. Click Apply to submit the changes (or close the page without applying changes).

9.4.5 Deleting an Application Domain and Its Content

Users with valid OAM Administrator credentials can perform the following task to delete an application domain (including its resources, policies, constraints, and responses) using the OAM Administration Console.

WARNING: The application domain must be empty before you delete it. You must delete all authentication and authorization policies and resources before you can delete the domain itself. Deleting the domain does not automatically delete the host identifier used in the domain.

Prerequisites

Ensure that resources in the domain to be deleted are placed in another application domain for protection.

To delete an application domain

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
       Domain Name

2. Delete all Authentication Policies:
   a. Expand the Authentication Policies node to reveal existing policies.
   b. Click the policy name, click the Delete button in the tool bar, and confirm removal in the Confirmation window.
   c. Repeat Steps a through c to remove all Authentication Policies.
3. Delete all Authorization Policies:
   a. Expand the Authorization Policies node to reveal existing policies.
   b. Click the policy name, click the Delete button in the tool bar, and confirm removal in the Confirmation window.
   c. Repeat Steps a through c to remove all Authorization Policies.
4. Delete all Resources in the Domain:
   a. Expand the Resources node to reveal existing resource names.
   b. Click the resource name, click the Delete button in the tool bar, and confirm removal in the Confirmation window.
   c. Repeat Steps a through c to remove all resources from the application domain.
5. In the navigation tree, click the application domain name and then click the Delete button in the tool bar.
6. In the Confirmation window, click Delete (or click Cancel to dismiss the window).
7. Check the navigation tree to confirm the application domain has been removed.

9.5 Adding and Managing Resource Definitions for Use in Policies

Protecting resources requires an application domain containing specific resource definitions. With OAM, you can protect different types of resources, including:

- An entire external Web site

- Specific pages in a Web site
- Partner portals
- A parts order application
- Invoice applications
- A benefits enrollment application on Web servers of an enterprise in many countries

This section provides the following topics:

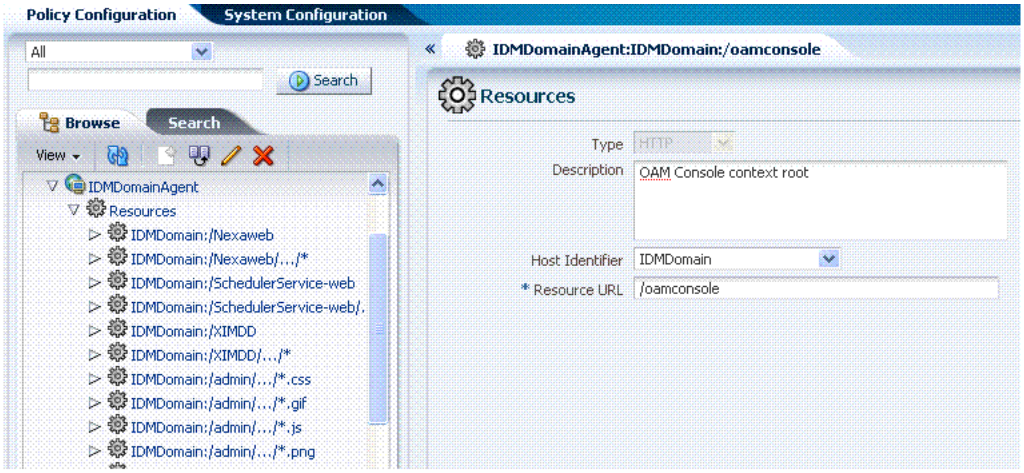
- About the Resource Definition Page in an Application Domain
- Searching for a Resource URL Definition
- Adding Resource Definitions to an Application Domain
- Viewing or Editing a Resource Definition in an Application Domain
- Deleting a Resource Definition from an Application Domain

9.5.1 About the Resource Definition Page in an Application Domain

Within an application domain, resource definitions exist as a flat collection of objects. Each resource is defined as a specific resource type, and the URL prefix that identifies a document or entity stored on a server and available for access by a large audience. The location is specified using an existing shared Host Identifier.

Each resource is defined separately in an application domain. Figure 9–8 illustrates a typical Resources definition page, which illustrates details of one resource in the pre-configured IDMDomainAgent application domain.

Figure 9–8 Resources Page in an Application Domain



IDMDomainAgent:IDMDomain:/oamconsole

Type: HTTP

Description: OAM Console context root

Host Identifier: IDMDomain

* Resource URL: /oamconsole

Table 9–1 describes elements that must be specified on the Resources page.

Table 9–1 Resource Definition Elements

Elements	Description
Type	The HTTP type is the default; it covers resources that are accessed using either the HTTP or HTTPS protocol. Policies that govern a particular resource apply to all operations. See Also "Managing Resource Types" on page 8-2.

Table 9–1 (Cont.) Resource Definition Elements

Elements	Description
Description	An optional unique description for this resource.
Host Identifier	A list of host identifiers is available, which contains all identifiers that were defined as a shared component. You must choose a host identifier to assign this resource. **Note**: The combination of the host identifier and URL string that make up a resource definition must be unique across all application domains. See Also: "Managing Host Identifiers" on page 8-5.
Resource URL	The URL value must be expressed as a single relative URL string that represents a path component of a full URL composed of a series of hierarchical levels separated by the '/' character. The URL value of a resource must begin with / and must match a resource value for the chosen host identifier. Other components, such as the query string or fragment, are not used. Based on its contents, a URL is matched in response to an incoming request as a literal or a wild card pattern. The special characters available to define a pattern, if included, are: ■ The asterisk (*) is allowed only at the lowest, terminating level of the path. The asterisk matches zero or more characters. ■ An ellipses (…) is allowed at any level of the path except the terminating level. The ellipses represents a sequence of zero or more intermediate levels. See Also Table 9–2.

Table 9–2 shows sample URL values for resources. For more information, see "About the Resource URL" on page 9-14.

Table 9–2 HTTP Resources Sample URL Values

Resource	Sample URL Values
Directories	■ /mydirectory ■ /mydirectory/projects ■ /mydirectory/*
Pages	■ /mydirectory/projects/index.html ■ /mydirectory/projects/*.html ■ /mydirectory/.../*.html ■ /.../*.html
Web applications	■ /mydirectory/projects/myexe.exe ■ /mydirectory/projects/*.exe ■ /mydirectory/.../*.exe ■ /.../*.exe

After adding the resource, it is grouped under the Resources node of the named application domain. When you create authentication and authorization policies all defined resources for the domain appear on a list so that you can choose one or more for inclusion in the policy.

For more information on resource definitions, see the following topics:

- About the Resource Type in a Resource Definition
- About the Host Identifier in a Resource Definition
- About the Resource URL
- About Run Time Resource Evaluation

9.5.1.1 About the Resource Type in a Resource Definition

When adding a resource definition to an application domain, administrators must choose from a list of defined Resource Types.

When adding an HTTP type resource to an application domain, administrators choose from a list of existing host identifiers and then add the resource URL. Operations associated with the HTTP resource type need not be defined by an administrator. Instead, policies apply to all HTTP operations.

Non-HTTP resource types are named and are not associated with a URL. When adding a non-HTTP type resource definition, administrators must enter the type's name into the Resource URL field.

9.5.1.2 About the Host Identifier in a Resource Definition

Administrations identify resources in an application domain by the host where the resources reside and the resource URL.

Note: Non-HTTP resource types are not associated with a host identifier. Instead, administrators must enter the type's name into the Resource URL field of the resource definition page.

Host identifiers create a context for each resource, which is useful when adding resources that have the same URL paths on different computers. Administrations can protect all of these resources in the same way within the same application domain. The only variable that distinguishes one set of resources from another is identification of its host computer.

All defined host identifiers appear on the Host Identifiers list on the Resources page. When adding a resource to an application domain, administrations must choose one host identifier for the computer hosting the resource.

To ensure that OAM recognizes the URL for a resource, OAM must know the various ways used to refer to that resource's host computer.

9.5.1.3 About the Resource URL

During automated application domain generation, a URL prefix is defined under which all resources are protected. Resources are linear, not hierarchical. Resource definitions are treated as complete URLs.

Note: There is no host identifier or URL associated with a non-HTTP resource type.

Administrations identify individual resources in the application domain using a specific resource URL. Individual resource URLs need not be unique across domains. However, the combination of a resource URL and a host identifier must be unique across domains.

An HTTP type resource is expressed as a single relative URL string representing a path. The string is composed of a series of hierarchical levels separated by the '/' character. Based on its content, a URL is matched in response to an incoming request as a literal or a wild card pattern.

URL Prefixes

The policy model does not support a resource prefix. If a policy is defined for `/mydirectory/projects/`, it only applies to this URL (and does not apply to `/mydirectory/projects/index.html`, for example). In other words, there is no policy inheritance. If you need a policy for all resources with the same prefix string, you can define the resource using special characters (three periods ... (ellipsis) or * (asterisk) for instance: `/mydirectory/projects/.../*`.

URL Patterns

Administrators can create granular URL patterns to specify the fine-grained portion of a resource's namespace.

Pattern matching is provided for only the following two patterns (with limited features):

… *

The three periods (called an ellipsis) matches any sequence of one or more characters that starts and ends with the forward slash character (/). It represents a sequence of zero or more intermediate levels and enables spanning multiple directories. The ellipsis (...) can be used at any level of the path except the terminating one, and only one time in any URL. The ellipsis serves to protect every resource within the applicable domain.

The * (asterisk) can be used only at the lowest, terminating level of the path. It matches zero or more characters. Every character in a URL pattern must match the corresponding character in the URL path exactly, with two exceptions:

- At the end of a pattern, /* matches any sequence of characters from that point forward.
- The pattern *.extension matches any file name ending with the named extension.

> **Note:** No other wild cards are supported. An asterisk at any other position in the pattern is not a wild card.

For example the following URL pattern:

```
/.../update.html
```

matches these resources:

```
/humanresources/benefits/update.html
/corporate/news/update.html
update.html
```

Table 9–3 illustrates a number of resource definitions within a single application domain. These are organized alphabetically according to the Host Identifier and Resource URLs.

Table 9–3 Resource URLs for.jsp

Resource URL	Correct
/bank/accounts/*	Yes
/bank/accounts/*.jsp	Yes
/bank/accounts/checking	Yes
/bank/.../checking.jsp	Yes

Table 9–3 (Cont.) Resource URLs for.jsp

Resource URL	Correct
/.../*.jsp	Yes
/bank/accounts/checking*.jsp	No
/bank/accounts/c*.jsp	No
/bank/.../accounts/def.gif	No

9.5.1.4 About Run Time Resource Evaluation

While processing requests for resources, an evaluation is made to ensure that the proper policy is invoked for the resource.

> **See Also:** "Managing Run Time Policy Evaluation Caches" on page 4-16

Process overview: Resource evaluation

1. A user specifies the URL for a requested resource.
2. Based on the host identifier and URL, OAM creates a fully qualified URL that includes the URL pattern.
3. OAM compares the incoming URL for the requested resource to the fully qualified URL constructed from the application domain information and the policy's URL pattern:
   - If there is a match, the various policies are evaluated to determine whether the requester should be allowed or denied access to the resource.
   - If requester is allowed access, the resource is served to the user.

Table 9–4 describes the possible outcomes.

Table 9–4 Resource Evaluation Outcomes

Outcome	Description
Best Match	The best match is when a resource definition has the least resource scope compared to other possible matches to the run time resource. The term resource scope represents all possible resources that could be matched using a particular resource definition
No Match1	If no match is found, the default evaluation outcome is FAILURE. Depending on what kind of policy was being evaluated, this could mean no authentication is attempted, or no resource access is granted.

Look Up Mechanism Examples

- The default resource URL in a generated application domain defines the broadest scope of content possible (all directories and below):

  ```
  /.../*
  ```

- The pattern /.../index.html matches:

  /index.html

  /oracle/index.html

  /oracle/sales/index.html

  index.html

  It does not match, for example, `xyzindex.html`.

- /oracle/.../*.html matches:

  /oracle/index.html

  /oracle/sales/order.html, and so on.

Resource Scope Examples

- Resource scope of the following resource definition (includes the asterisk):

  /mybank/…/*

  includes all URLs prefixed with "/mybank/"

- Resource scope of the following resource definition (no special characters in the definition):

  /mybank/account.html

  includes only one URL: "/mybank/account.html"

9.5.2 Adding Resource Definitions to an Application Domain

Users with valid OAM Administrator credentials can use the following procedure to add the resource definitions to protect to the corresponding application domain.

> **Note:** Failure can occur if you specify a host identifier value that is invalid. An error informs you that the challenge URL is invalid.

Prerequisites

The resource type must be defined as a Shared Component. For details, see "Managing Resource Types" on page 8-2.

> **See Also:** "About the Resource Definition Page in an Application Domain" on page 9-12

To add resource definitions to an application domain

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
   Desired Domain

2. Click the Resources node, then click the Create button in the tool bar.
3. On the Resource Definition page:
   a. Add details for a single resource:

      Type
      Description (optional)
      Host Identifier
      Resource URL (see Table 9–2)

   b. Click Apply to add this resource to the application domain.
   c. In the navigation tree, expand Resources to confirm that the addition is successful (or click the Refresh button in the tool bar).
   d. Repeat this procedure to add other resources to this application domain.
4. Proceed to adding a resource to specific policies as described in:

- Defining Authentication Policies for Specific Resources
- Defining Authorization Policies for Specific Resources

9.5.3 Searching for a Resource URL Definition

Users with valid OAM Administrator credentials can use the following procedure to search for a specific resource definition.

> **See Also:** "About Search Controls" on page 2-24

To search for a specific resource definition

1. Activate the Policy Configuration tab.
2. From the search type list, choose Resources to define your search.
3. In the text field, enter the exact name of the instance you want to find. For example:

   ```
   my_Resource_Name
   ```

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.
6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

9.5.4 Viewing or Editing a Resource Definition in an Application Domain

Users with valid OAM Administrator credentials can use the following procedure to modify resource definitions within a specific application domain.

Prerequisites

You must have the desired resource type defined as a shared component. For details, see "Managing Resource Types" on page 8-2.

> **See Also:** "About the Resource Definition Page in an Application Domain" on page 9-12

To view or modify resource definitions in an application domain

1. From the Policy Configuration tab, navigation tree, expand the:

   Application Domains
   Desired Domain
   Resources

2. Double-click the desired resource definition to open the page and proceed as follows:
   - View Only: Close the page when you finish.

- Modify: Alter the definition as desired and then click Apply to submit changes (or close the page without applying changes).

9.5.5 Deleting a Resource Definition from an Application Domain

Users with valid OAM Administrator credentials can use the following procedure to delete a resource from an application domain.

Prerequisites

Ensure that the resource definition you will delete is not used in any policy.

To delete resource definitions from an application domain

1. From the Policy Configuration tab navigation tree, expand the following nodes:

   Application Domains
   Desired Domain
   Resources

2. Optional: Double-click the desired resource definition and confirm this is the one to be deleted, then close the page.
3. Click the name of the desired resource definition and then click the Delete button in the tool bar.
4. In the Confirmation window, click Delete (or click Cancel to dismiss the window).
5. Repeat this procedure as often as needed to delete other resources in an application domain.

9.6 Defining Authentication Policies for Specific Resources

Each resource assigned to an application domain can be protected by only one authentication policy. After adding a resource definition to the application domain, the administrator can begin refining a default authentication policy, adding a new policy, and assigning resources to the authentication policy.

In an automatically generated application domain, the following authentication policies are seeded as defaults to help streamline the administrator's tasks:

- Protected Resource
- Public Resource

> **See Also:** "Anatomy of an Application Domain and Policies" on page 9-3

This section provides the following topics:

- About the Authentication Policy Page
- Adding an Authentication Policy and Resources
- Searching for an Authentication Policy
- Viewing or Editing an Authentication Policy
- Deleting an Authentication Policy

9.6.1 About the Authentication Policy Page

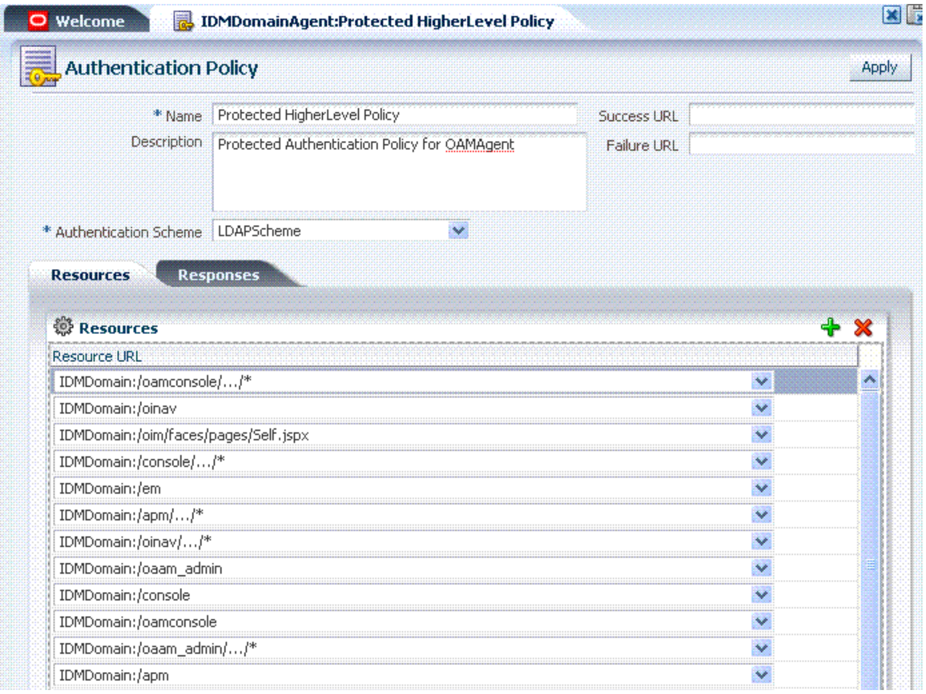
Administrators use authentication policies to protect specific resources. The authentication policy provides the sole authentication method for resources governed by the policy.

Each authentication policy defines the type of verification that must be performed to provide a sufficient level of trust for Oracle Access Manager to grant access to the user making the request.

Authentication policies are local. A single policy can be defined to protect one or more resources in the application domain. However, each resource can be protected by only one authentication policy. There is no policy inheritance as there is with Oracle Entitlement Server. The policy cannot be applied to any other resource.

Figure 9–9 shows the Authentication Policy page within an application domain. The resources assigned to this policy are displayed on the Resources tab of the policy. This example is from the IDMDomainAgent application domain.

Figure 9–9 Authentication Policy Page: IDMDomainAgent



* Name: Protected HigherLevel Policy
Description: Protected Authentication Policy for OAMAgent
Success URL:
Failure URL:
* Authentication Scheme: LDAPScheme

Resource URL
IDMDomain:/oamconsole/.../*
IDMDomain:/oinav
IDMDomain:/oim/faces/pages/Self.jspx
IDMDomain:/console/.../*
IDMDomain:/em
IDMDomain:/apm/.../*
IDMDomain:/oinav/.../*
IDMDomain:/oaam_admin
IDMDomain:/console
IDMDomain:/oamconsole
IDMDomain:/oaam_admin/.../*
IDMDomain:/apm

Table 9–5 describes authentication policy elements. The IDMDomainAgent application domain is shown simply as an example.

Table 9–5 Authentication Policy Elements and Descriptions

Element	Description
Name	A unique name used as an identifier in the navigation tree.
Description	Optional unique text that describes this authentication policy.
Authentication Scheme	A single, previously-defined authentication scheme to be used by this policy for user authentication. See Also: "Managing Authentication Schemes" on page 8-18 for details.
Success URL	The redirect URL to be used upon successful authentication.
Failure URL	The redirect URL to be used if authentication fails.

Table 9–5 (Cont.) Authentication Policy Elements and Descriptions

Element	Description
Resources	The URL of a resource chosen from those listed. The listed URLs were added to this application domain earlier. You can add one or more resources to protect with this authentication policy. The resource definition must exist within the application domain before you can include it in a policy. See Also: "About Resources in an Authentication Policy" on page 9-21.
Responses	The obligations (post authentication actions) to be carried out by the Web agent. After a successful authentication, the application server hosting the protected application should be able to assert the User Identity based on these responses.After a failed authentication, the browser redirects the request to a pre-configured URL See Also: "Introduction to Policy Responses for SSO" on page 9-28.

9.6.1.1 About Resources in an Authentication Policy

You can choose to add one or more resources to be protected by the authentication policy. The Resources tab on the Authentication Policy page provides a table where you can enter resource URLs. A list is also provided from which you can choose from defined resources within the application domain.

To add a resource, click the + button and select from the list. To delete a resource, select the name from the Resources table and click the Delete button in the table.

9.6.2 Adding an Authentication Policy and Resources

Users with valid OAM Administrator credentials can use the following procedure to add an authentication policy and resources to an application domain. You can use a pre-configured authentication scheme or a custom authentication scheme in the authentication policy.

See Also:

- "About the Authentication Policy Page" on page 9-20
- "Managing Authentication Schemes" on page 8-18

Prerequisites

Any resource to be added to a policy must be defined within the same Application Domain as the policy.

To add an authentication policy for specific resources

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain
       Authentication Policies

2. Click the Create button in the tool bar to open a fresh page.
3. **Add General Policy Details**:
   - Name
   - Authentication Scheme
4. **Add Global Policy Elements** (Table 9–5):
   - Description (optional)

- Success URL
- Failure URL

5. **Add Resources**:
   - Click the Resources tab on the Authentication Policy page.
   - Click the Add button on the tab.
   - Click a URL from the list.
   - Repeat these steps as needed to add more resources.
6. Click Apply to save changes and close the Confirmation window.
7. Add policy Responses, as described in "Adding and Managing Policy Responses for SSO" on page 9-33.
8. Close the page when you finish.

9.6.3 Searching for an Authentication Policy

Users with valid OAM Administrator credentials can use the following procedure to search for a specific authentication policy.

> **See Also:** "About Search Controls" on page 2-24

To search for an authentication policy in an application domain

1. Activate the Policy Configuration tab.
2. From the search type list, choose Authentication Policies to define your search.
3. In the text field, enter the exact name of the policy you want to find. For example:

   my_AuthNPolicy

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit command button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.
6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

9.6.4 Viewing or Editing an Authentication Policy

Users with valid OAM Administrator credentials can use the following procedure to modify an authentication policy in an application domain. This includes changing the authentication scheme, adding or removing resources or responses, and altering the Success or Failure URLs.

> **See Also:** "About the Authentication Policy Page" on page 9-20

To view or modify an authentication policy

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
   Desired Domain
   Authentication Policies

2. Double-click the desired authentication policy name.

   The Authentication Policy is opened and its Resource tab is available.

3. General Policy Elements:

   - Name
   - Authentication Scheme

4. Global Policy Elements: Edit as desired, (Table 9–5):

   - Description
   - Success URL
   - Failure URL

5. Resource: Click the Resources tab and perform the following tasks as needed:

   - Add: Click the Add button on the Resources table, click a URL in the list, click Apply.
   - Delete: Click a URL in the Resources table, click the Delete button on the table.

6. Click Apply to submit changes and close the Confirmation window (or close the page without applying changes)
7. Responses: View or edit responses as described in "Adding and Managing Policy Responses for SSO" on page 9-33.
8. Close the page when you finish.

9.6.5 Deleting an Authentication Policy

Users with valid OAM Administrator credentials can use the following procedure to delete an authentication policy from an application domain.

When you remove the policy, all resource definitions remain within the application domain. However, the policy and all responses are eliminated.

> **See Also:** "About the Authentication Policy Page" on page 9-20

The following procedure describes how to delete the entire policy. To simply alter an element in the policy, see "Viewing or Editing an Authentication Policy".

To delete an authentication policy

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
   Desired Domain
   Authentication Policies

2. Optional: Double-click the desired policy name to review its content, and then close the page.

3. Delete all responses, as described in "Adding and Managing Policy Responses for SSO" on page 9-33.
4. In the navigation tree, click the name of the authentication policy, then click the Delete button in the tool bar.
5. In the Confirmation window, click Delete to confirm (or click Cancel to dismiss the window).
6. Ensure that resources governed by this policy are added to a different policy.

9.7 Defining Authorization Policies for Specific Resources

Each resource assigned to an application domain can be protected by only one authorization policy.

In an automatically generated application domain, the following authorization policies are seeded as defaults:

- Protected Resource
- Public Resource

> **See Also:** "Anatomy of an Application Domain and Policies" on page 9-3

After adding resource definitions to the application domain, administrators can begin refining a default authorization policy, adding a new policy, and adding resources to authorization policies. This section provides the following topics:

- About Authorization Policies for Specific Resources
- Adding an Authorization Policy and Specific Resources
- Searching for an Authorization Policy
- Viewing or Editing an Authorization Policy and Resources
- Deleting an Authorization Policy

9.7.1 About Authorization Policies for Specific Resources

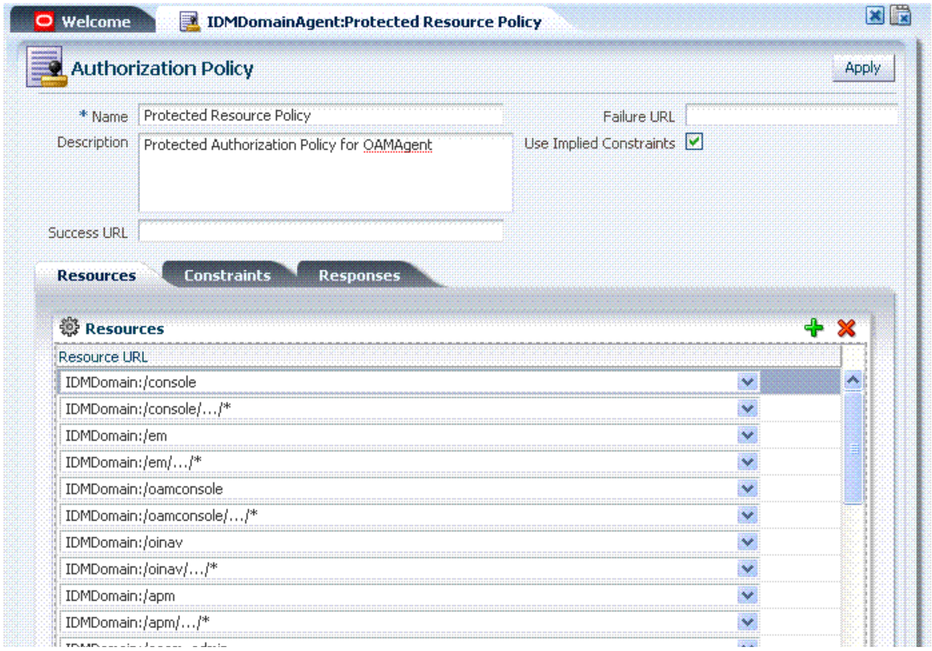
Administrators can create an authorization policy to protect access to one or more resources based on attributes of an authenticated user or the environment. The authorization policy provides the sole authorization protection for resources included in the policy.

Authorization policies are local, which means that each policy applies only to the resources specified for the policy. A policy cannot be derived or applied to any other resource.

A single policy can be defined to protect one or more resources in the application domain. However, each resource can be protected by only one authorization policy.

Figure 9–10 shows the Authorization Policy page within an application domain. The resources assigned to this policy are displayed on the Resources tab of the policy. The IDMDomainAgent application domain is shown simply as an example.

Figure 9–10 Authorization Policy Page: IDMDomainAgent



* Name: Protected Resource Policy

Description: Protected Authorization Policy for OAMAgent

Success URL:

Failure URL:

Use Implied Constraints ☑

Resources

Resource URL
IDMDomain:/console
IDMDomain:/console/.../*
IDMDomain:/em
IDMDomain:/em/.../*
IDMDomain:/oamconsole
IDMDomain:/oamconsole/.../*
IDMDomain:/oinav
IDMDomain:/oinav/.../*
IDMDomain:/apm
IDMDomain:/apm/.../*

Table 9–6 describes authorization policy elements. The elements are the same regardless of the domain; only the details will differ. The IDMDomainAgent application domain is shown simply as an example.

Table 9–6 Authorization Policy Elements and Descriptions

Element	Description
Name	A unique name used as an identifier in the navigation tree.
Description	Optional unique text that describes this authorization policy.
Success URL	The redirect URL to be used upon successful authorization.
Failure URL	The redirect URL to be used if authorization fails.
Use Implied Constraints	Allows (or denies) access. Set on the authorization policy to allow access in the absence of any authorization constraints of a particular class. Default: Checked (enabled) See Also "Introduction to Authorization Constraints" on page 9-35.
Resources Tab	One or more previously-defined resource URLs to be protected by this authorization policy.
Constraints Tab	See Also "Introduction to Authorization Constraints" on page 9-35.
Responses Tab	See Also "Introduction to Policy Responses for SSO" on page 9-28.

9.7.2 Adding an Authorization Policy and Specific Resources

Users with valid OAM Administrator credentials can use the following procedure to add an authorization policy to an application domain.

Prerequisites

Any resource to be added to a policy must be defined within the same Application Domain as the policy.

> **See Also:** "About Authorization Policies for Specific Resources" on page 9-24

To add an authorization policy and resources

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain
       Authorization Policies

2. Click the Create button in the tool bar.
3. Add a unique name for this authorization policy.
4. Global Policy Elements: Enter your own details (Table 9–6):
   - Description (optional)
   - Success URL
   - Failure URL
   - Use Implied Constraints
5. **Resources**:
   - On the Resource tab, click the Add button.
   - From the list provided, click a resource URL.
   - Repeat these steps to add more resources to this policy.
6. Click Apply to save changes and close the Confirmation window.
7. **Constraints**: Add authorization constraints, as described in "Defining Authorization Policy Constraints" on page 9-42.
8. **Responses**: Add or edit responses for SSO, as described in "Adding and Managing Policy Responses for SSO" on page 9-33.
9. Close the page when you finish.

9.7.3 Searching for an Authorization Policy

Users with valid OAM Administrator credentials can use the following procedure to locate a specific authorization policy.

> **See Also:** "About Search Controls" on page 2-24

To search for an authorization policy

1. Activate the Policy Configuration tab.
2. From the search type list, choose Authentication Policies to define your search.
3. In the text field, enter the exact name of the policy you want to find. For example:

   ```
   my_AuthZPolicy
   ```

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit:** Click the Edit button in the tool bar to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the instance; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **View**: Select a View menu item to alter the appearance of the results table.

6. Click the Browse tab to return to the navigation tree when you finish with the Search results.

9.7.4 Viewing or Editing an Authorization Policy and Resources

Users with valid OAM Administrator credentials can use the following procedure to view or modify an authorization policy within an application domain.

> **See Also:** "About Authorization Policies for Specific Resources" on page 9-24

To view or edit an authorization policy

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
   Desired Domain
   Authorization Policies

2. Double-click the desired policy name to display details.
3. **Global Elements**: Edit as needed (Table 9–6):
   - Description (optional)
   - Success URL
   - Failure URL
   - Use Implied Constraints
4. **Resource**: Click the Resources tab and perform the following tasks as needed:
   - Add: Click the Add button on the Resources table, click a URL in the list, click Apply.
   - Delete: Click a URL in the Resources table, click the Delete button on the table.
5. Click Apply to submit changes and close the Confirmation window (or close the page without applying changes).
6. **Constraints**: View or edit these as described in "Viewing, Editing, or Deleting Authorization Policy Constraints" on page 9-45.
7. **Responses**: View or edit these as described in "Viewing, Editing, or Deleting a Policy Response for SSO" on page 9-34.
8. Close the page when you finish.

9.7.5 Deleting an Authorization Policy

Users with valid OAM Administrator credentials can use the following procedure to delete an authorization policy or simply delete resources within the policy.

When you remove the entire policy, all resource definitions remain within the application domain. However, the authorization policy and the constraints and responses governing access are eliminated.

> **Note:** To simply alter an element in the policy see "Viewing or Editing an Authentication Policy".

> **See Also:** "About Authorization Policies for Specific Resources" on page 9-24

Prerequisites

Assign resources governed by this policy to another authorization policy, either before or after deleting the policy.

To delete an authorization policy

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain
       Authorization Policies

2. Optional: Double-click the policy name to review its content, and then close the page when finished.
3. Click the policy name, and then click the Delete button in the tool bar.
4. In the Confirmation window, click Delete (or click Cancel to dismiss the window).
5. Confirm that the policy is no longer listed in the navigation tree.

9.8 Introduction to Policy Responses for SSO

Each policy can optionally contain one or more authentication or authorization responses, or both. Responses are post-processing actions (obligations) to be carried out by the web agent.

This section provides the following information:

- About Authentication and Authorization Policy Responses for SSO
- About the Policy Response Language
- About the Namespace and Variable Names for Policy Responses
- About Constructing a Policy Response for SSO
- About Policy Response Processing

9.8.1 About Authentication and Authorization Policy Responses for SSO

Administrators can define responses that declare the actions that must be fulfilled after successful authentication or authorization. Authentication and authorization data is returned to the client (typically a Web Agent).

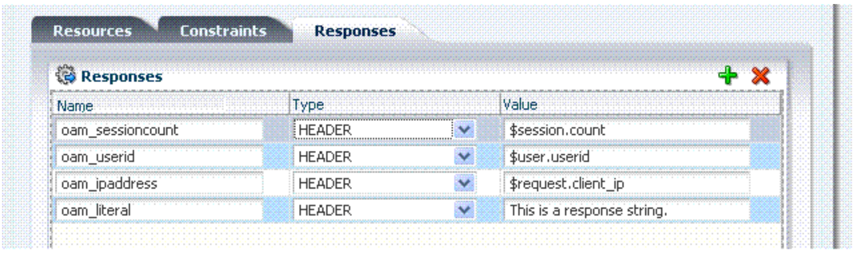
Policy responses enable the insertion of information into a session or application and the ability to withdraw the information at a later time to enable SSO. For instance, identity mappings can be inserted into the customer's application or actions can be carried out by the Agent or the application.

Depending on the responses specified for authentication or authorization success and failure, the user might be redirected to a specific URL, or user information might be passed on to other applications through a header variable or a cookie value.

> **Note:** OAM 10g provided data passage to (and between) applications only by redirecting to URLs in a specific sequence.

There are no default response provided. Figure 9–11 illustrates an Authorization Policy Response defined by an administrator in the OAM Administration Console. Authorization responses can operate in conjunction with authorization constraints.

Figure 9–11 Authorization Policy Response in the Administration Console



Name	Type	Value
oam_sessioncount	HEADER	$session.count
oam_userid	HEADER	$user.userid
oam_ipaddress	HEADER	$request.client_ip
oam_literal	HEADER	This is a response string.

Each response consists of two inputs (a type and an expression) and a single output (the value of the evaluated expression). The expression declares how the value should be constructed when the expression is processed. The response type defines the form of action to be taken with the value string.

- The authentication policy determines how the identity of the user. Each authentication policy requires an authentication scheme and responses (expressions).
- The authorization policy determines whether the user has the right to access the resource. Each authorization policy requires authorization constraints and responses (expressions).

Administrators set Responses in the Administration Console, as described Table 9–7.

Table 9–7 Response Elements

Element	Description
Name	A unique name to distinguish this response from other responses that use the same mechanism (type).
Type	The mechanism used to convey the response. form of the action to be taken with the value string: ■ **HEADER (Header variables)**: Sets an HTTP request header for downstream applications using the defined value to dictate the action to be taken (such as the assertion of a User ID using a pre-defined HTTP header name). ■ **SESSION**: Sets an attribute inside the user session by the client (to enable single sign-on) based on the defined session variable name and value. ■ **COOKIE**: Sets a variable name and value (typically set by Web agents) inside the authentication session cookie to enable single sign-on. In cookie-less mode, Web-cache is currently used to store cookies from WebGate. However, in cookie-less mode, the end application does not have access to cookies and cannot use them.
Value	The response expression, set as a variable. For more information, see "About the Policy Response Language".

9.8.2 About the Policy Response Language

OAM 11g authentication and authorization responses are defined using a very small, domain-specific language (DSL) with two main constructs:

- Literal strings: For example: `This is a valid expression`
- Variable references:

- Declared using a dollar sign prefix `$`
- Scoped to a namespace: `$namespace.var_name`

Note: Certain variables include an attribute: `$ns.name.attribute`

9.8.3 About the Namespace and Variable Names for Policy Responses

With the namespace mechanism, the following variable types are supported to enable single sign-on:

- Request: Information on the requested resource, the client making the request, and the policy matched during evaluation
- Session: User session details
- User: User details (user ID, group, and attribute information)

For details of each, see:

- Table 9–8, " Namespace Request Variables for Single Sign-On"
- Table 9–9, " Namespace Session Variables for Single Sign-On"
- Table 9–10, " Namespace User Variables"

Table 9–8 Namespace Request Variables for Single Sign-On

Namespace	Description
agent_id	Name of the requesting agent
client_ip	IP address of the user browser
policy_appdomain	Name of the application domain holding the policy matched for the request
policy_res	Resource host ID and URL pattern matched for the request
res_policy	Name of the specific policy matched for the request
res_host	Requested resource's hostname
res_port	Requested resource's port number
res_type	Requested resource's type
res_url	Requested resource URL

Table 9–9 Namespace Session Variables for Single Sign-On

Namespace	Description
attr	Reference to an arbitrary session attribute, the name of which is passed to us as a variable attribute. Its value has been bound to the session by executing a session response during a previous request
authn_level	Current authentication level for the session
authn_scheme	Name of the authentication scheme executed to achieve the current authentication level
count	Session count for the user bound to this session
creation	Session creation time
expiration	Session expiration time

Table 9–10 Namespace User Variables

Namespace	Description
attr	Reference to an arbitrary LDAP attribute, the name of which is passed to us as a variable attribute
groups	Comma-separated list of the user's group membership
userid	The user ID

9.8.4 About Constructing a Policy Response for SSO

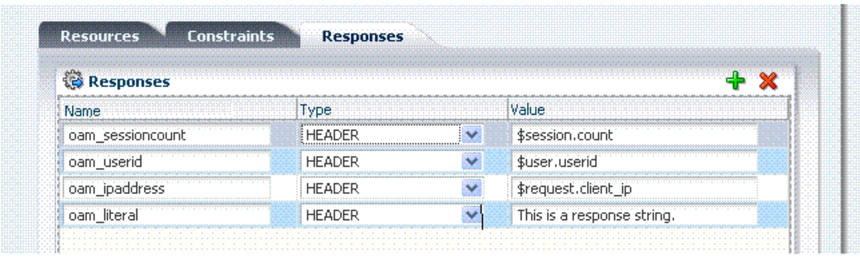
This section is divided as follows:

- Simple Responses
- Compound and Complex Responses

9.8.4.1 Simple Responses

After deciding on the response type and determining which namespace and variable, you simply enter the response attributes in the Administration Console. A simple response might look like one of the several authorization responses shown in Figure 9–12.

Figure 9–12 Simple Response Samples



Name	Type	Value
oam_sessioncount	HEADER	$session.count
oam_userid	HEADER	$user.userid
oam_ipaddress	HEADER	$request.client_ip
oam_literal	HEADER	This is a response string.

Simple responses stand alone. Each is preceded with the dollar sign ($), followed by the namespace, which is separated from the variable Value by a dot (.). For example:

```
$namespace1.var1
```

Table 9–11 illustrates several simple responses and a description of what each one returns.

Table 9–11 Simple Responses and Descriptions

Name	Type	Value (Simple $Namespace.Variable)	Returned Environment Variables and Values
oam_sessioncount	Header	$session.count	HTTP_OAM_SESSIONCOUNT *integer*
oam_userid	Header	$user.userid	HTTP_OAM_USERID *name*
oam_ipaddress	Header	$request.client_ip	HTTP_OAM_IPADDRESS *nnn.nn.nn.nnn*
oam_literal	Header	This is a response string.	HTTP_OAM_LITERAL *This is a response string*

9.8.4.2 Compound and Complex Responses

When crafting a compound or complex policy response, administrators can combine literals and variables arbitrarily using braces { } to construct an expression. A colon (:) is used as a separator. For example:

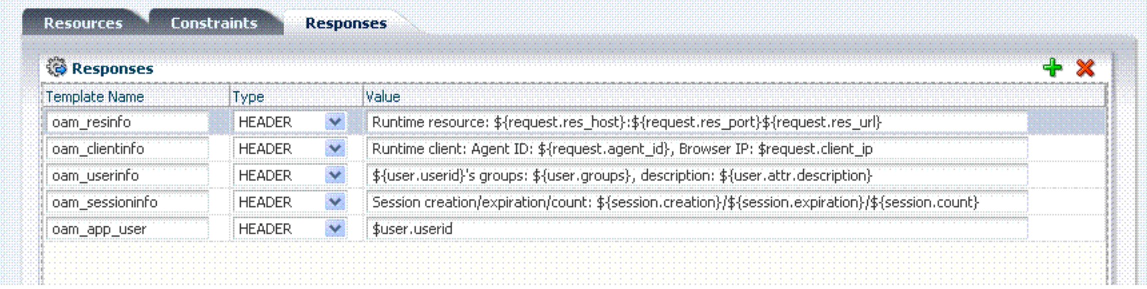
```
${namespace1.var1}:${namespace2.var2}

Literal String (LS): ${namespace1.var1}:${namespace2.var2}

LS: ${namespace1.var1}, LS:${namespace2.var2}
```

Figure 9–13 illustrates several complex responses defined by an administrator. All are Header type responses, which set values in a header variable of an HTTP request for consumption by a downstream application.

Figure 9–13 Sample Complex Responses



Template Name	Type	Value
oam_resinfo	HEADER	Runtime resource: ${request.res_host}:${request.res_port}${request.res_url}
oam_clientinfo	HEADER	Runtime client: Agent ID: ${request.agent_id}, Browser IP: $request.client_ip
oam_userinfo	HEADER	${user.userid}'s groups: ${user.groups}, description: ${user.attr.description}
oam_sessioninfo	HEADER	Session creation/expiration/count: ${session.creation}/${session.expiration}/${session.count}
oam_app_user	HEADER	$user.userid

Table 9–12 describes the complex responses shown in Figure 9–13.

Table 9–12 Complex Responses

Name	Value	Returned Environment Variables and Values
oam_resinfo	Runtime resource: ${request.res_host}:${request.res_port}${request.res_url}	HTTP_OAM_RESINFO Runtime resource: myhost.domain.com:1234/cgi-bin/myres3
oam_clientinfo	Runtime client: Agent ID: ${request.agent_id}, Browser IP: $request.client_ip	HTTP_OAM_CLIENTINFO Runtime client: Agent ID: *RREG_OAM*, Browser IP: 123.45.67.891
oam_userinfo	${user.userid}'s groups: ${user.groups}, description: ${user.attr.description}	HTTP_OAM_USERINFO *WebLogic's groups: Administrators, description: This user is the default administrator*
oam_sessioninfo	Session creation/expiration/count: ${session.creation}/${session.expiration}/${session.count}	HTTP_OAM_SESSIONINFO Session creation/expiration/count: *Tue Feb 23 17:47:42 PST 2010/Wed Feb 24 01:47:42 PST 2010/7*
oam_app_user	$user.userid	HTTP_OAM_USERID *name*

For more information, see "About Policy Response Processing".

9.8.5 About Policy Response Processing

Policy response processing occurs during the authorization request for which the authentication responses are replayed. Variable references are filled with appropriate values to ensure that all variables have a value set, and can be set consistently with authorization values.

Processing a response expression is done through a series of steps:

- Scanner/tokenizer
- Parser
- Interpreter

During interpretation, variable references are resolved to values. The result after processing is a simple String value, which is propagated to the Agent or saved within the session for future use.

Authentication success responses are saved and then "replayed" along with any authorization responses on the first applicable authorization request.

Authorization response expressions create the actions to be taken, depending on the evaluation of the expression: success, failure, or inconclusive.

Note: OAM 10g exhibits the same behavior in the "authenticating WebGate" configuration. This is also employed by OAM 11g with 10g WebGates: The 10g WebGate always redirects to the OAM 11g credential collector which acts like the authenticating WebGate

---

When referencing a variable, either the value is returned, or the following is returned:

- NOT FOUND is returned if the variable is not set
- NULL is returned if the variable is set to a null value

Note: Verify the Responses as follows:

- Header:
- Session:
- Cookie: Use a browser plug-in tool or turn on the browser "show cookies" settings.

Pass Through Without Processing

A value that must be passed through without processing, can be identified using a \. For example:

```
\$1000
```

results in the value `$1000` appearing in the returned value.

9.9 Adding and Managing Policy Responses for SSO

Policies and responses enable single sign-on and can override other directives. Before starting activities in this section, be sure to review the "Introduction to Policy Responses for SSO" on page 9-28.

Unless explicitly stated, information in this section applies equally to authentication and authorization responses.

- Adding a Policy Response for SSO
- Viewing, Editing, or Deleting a Policy Response for SSO

9.9.1 Adding a Policy Response for SSO

Users with valid OAM Administrator credentials can use the following procedure to add a policy response for authentication or authorization.

Prerequisites

Analyze authorization constraints before crafting authorization responses to ensure the appropriate actions are taken by the response. You need an application domain with an existing authentication or authorization policy.

> **See Also:** "Introduction to Policy Responses for SSO" on page 9-28

To add a policy response

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain
       Authorization Policies (or Authentication Policies)

2. Double-click the desired policy name to open the page.
3. Click to activate the Responses tab, then click its Add button and:
   - In the Name field, enter a unique name for this response.
   - From the Type list, choose a response type (Session or Header or Cookie).
   - In the Value field, enter a value for this response. For example: $*namespace1.var1*

     > **See Also:** "About the Namespace and Variable Names for Policy Responses" on page 9-30

   - Repeat as needed.
4. Click Apply, then close the Confirmation window.
5. Close the page when you finish.
6. Verify the Responses based on your definitions for:
   - Header
   - Session:
   - Cookie: Use a browser plug-in tool or turn on the browser "show cookies" settings.

9.9.2 Viewing, Editing, or Deleting a Policy Response for SSO

Users with valid OAM Administrator credentials can use the following procedure to view or edit a policy response for authentication or authorization.

Prerequisites

You must have an application domain with an existing authentication or authorization policy.

> **See Also:** "Introduction to Policy Responses for SSO" on page 9-28

To view, modify, or delete a policy response

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain
       Authorization Policies (or Authentication Policies)

2. Double-click the desired policy name to open the page.
3. Responses: Click the Responses tab and proceed as needed:
   - Add a response as described in "Adding a Policy Response for SSO"
   - Edit: Click the desired response Name, Type, or Value, edit as needed, and click Apply.
   - Delete: Click the desired response, then click the Delete button for the Response table.
4. Responses: Edit a response as follows.
   a. Click the Responses tab.
   b. Click the desired response Name, Type, or Value.
   c. Make the desired change.
   d. Repeat as needed and click Apply when finished.
5. Click Apply, then close the Confirmation window.
6. Close the page when you finish.
7. Verify Responses based on your definitions for:
   - Header
   - Session
   - Cookie: Use a browser plug-in tool or turn on the browser "show cookies" settings.

9.10 Introduction to Authorization Constraints

Authorization constraints must be specified by an administrator to apply to all resources within a specific authorization policy in an application domain.

An authorization constraint is a rule that grants or denies access to a particular resource based on the context of the request for that resource. Authorization Constraints define the obligations (requirements) that must be fulfilled before responding to a client's request.

Evaluation of constraints determines if the authorization policy applies to the incoming request. The appropriate obligations take affect after successful authentication and work in concert with defined authorization responses. For each incoming request, the authorization policy determines if there are any constraints. During authorization, constraints are evaluated.

Allow versus Deny Type Constraints

Each authorization policy can contain one or more authorization constraints that determine whether access to the requested resource should be granted or denied.

Authorization constraints contain:

- An Allow type condition that specifies who is authorized to access a protected resource.
- A Deny type condition that specifies explicitly who is denied access to the protected resource.

> **Note:** When defining constraints within a particular policy, only a single outcome (allow or deny) is allowed.

Constraint Classes

Using different constraint classes, you can configure different constraints within the same authorization policy. For instance, you can configure constraints to:

- Identify the users or groups of users who are who are either allowed or denied access to protected resources.
- Stipulate the range of IP addresses who are either allowed or denied access to protected resources.

> **Note:** If the user's IP address falls outside the range of denied addresses, this by itself is not enough for authorization to be successful. For authorization to be successful, the user must specifically be granted access based on an Allow rule.

- Set a time period defining when the constraint applies.

This section provides the following topics:

- About Allow or Deny Type Constraints
- About Classifying Users and Groups for Constraints
- About Constraints and General Authorization Policy Details
- About the Add Constraint Window
- About Identity Class Constraints
- About IP4Range Class Constraints
- About Temporal Class Constraints

9.10.1 About Allow or Deny Type Constraints

Each authorization constraint enables you to include either a Allow or Deny type outcome. The Use Implied Constraints option can be set on the authorization policy to allow access in the absence of any authorization constraints.

If Allow Access conditions do not apply to a user, the user is not qualified by the policy and, by default, the user is denied access to the requested resource.

> **Note:** Access is denied when no constraints are defined and when Use Implied Constraints is not enabled.

9.10.2 About Classifying Users and Groups for Constraints

Oracle recommends that you consider the same information for the policies and constraints when analyzing users and groups to determine who is explicitly allowed or denied access. For example, one authorization policy might be constrained to a particular time of day (temporal class) while another might be constrained to a specific group of users (Identity class).

> **Note:** Do not be concerned about users who are denied access under any conditions. They are denied access by default if none of the constraints qualify them.

When classifying users Oracle recommends that you divide the users, and groups of users, into groups for whom different conditions apply. For example, constraints can determine when the users can access the resources, the computers from which they must make their requests, and so on.

If some users fall into multiple categories, for example, a user in the marketing group belongs to a certain project group, or a user in the human resources group also belongs to the project group, put the user in both categories. You can require that the user meet the conditions of two constraints.

9.10.3 Guidelines for Authorization Responses Based on Constraints

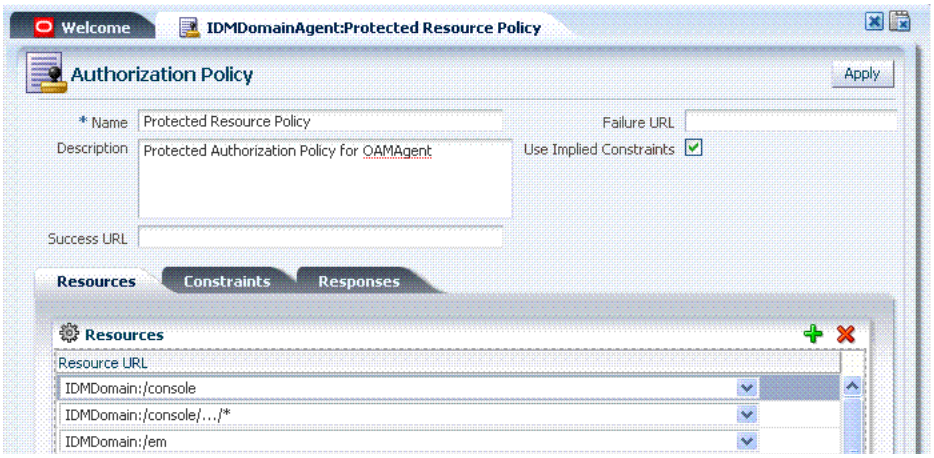
For each constraint class, consider the response actions that you want to occur for authorized users. For example, you may want the system to return user profile information and pass that information to a downstream application, as follows:

- If the user is authorized, you may want to pass the user's cn (common name) to another application so that the application can present a customized greeting to the user.
- If the user is not authorized, you may also want to return information about the user for security purposes.

9.10.4 About Constraints and General Authorization Policy Details

All constraint definitions apply along with the general authorization policy details shown in Figure 9–14.

Figure 9–14 Authorization Policy Page, General Details



* Name: Protected Resource Policy

Failure URL:

Description: Protected Authorization Policy for OAMAgent

Use Implied Constraints ☑

Success URL:

Resource URL
IDMDomain:/console
IDMDomain:/console/…/*
IDMDomain:/em

Table 9–13 describes the common authorization policy general details.

Table 9–13 Authorization Policy General Details

Element	Description
Name	A unique name used as an identifier in the navigation tree.
Description	Optional unique text that describes this authorization policy.
Success URL	The redirect URL to be used upon successful authorization.
Failure URL	The redirect URL to be used if authorization fails.
Use Implied Constraints	Allows (or denies) access. Set on the authorization policy to allow access in the absence of any authorization constraints of a particular class. Default: Checked (enabled)

9.10.5 About the Add Constraint Window

When an administrator adds a constraint to an authorization policy, the window shown in Figure 9–15 appears to capture the name, class, and outcome type of the constraint. When submitted, this information is used to create a container for the constraint details that must be specified by the administrator.

Figure 9–15 Add Constraint Window

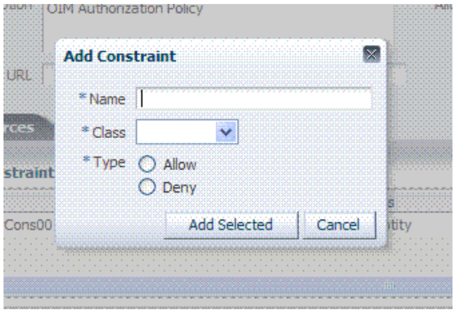


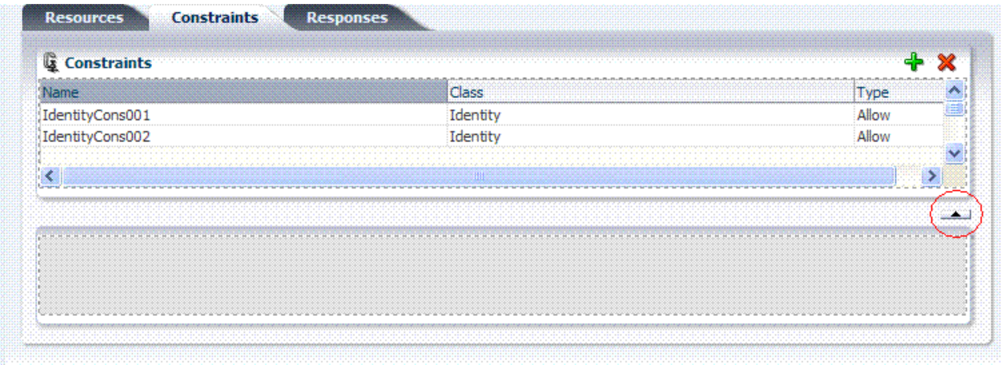
Table 9–14 describes the Add Constraint window elements.

Table 9–14 Add Constraint Window Elements

Element	Description
Name	A unique name for this constraint.
Class	Only one class can be specified (Identity, IP4 Range, or Temporal).
Type	Outcome type: Allow or Deny access.
Add Selected	Click this button to initiate creation of the constraint container.

After specifying the general details, the constraint container is added and displayed on the policy page as shown in Figure 9–16. Here, only the Name, Class, and Type are displayed. However, a window control (red circle in the figure) allows administrators to open a window where they can specify details for the selected constraint.

Figure 9–16 Constraint Containers on the Authorization Policy Page



Name	Class	Type
IdentityCons001	Identity	Allow
IdentityCons002	Identity	Allow

Constraint details are specific to the chosen constraint class, as described in following topics:

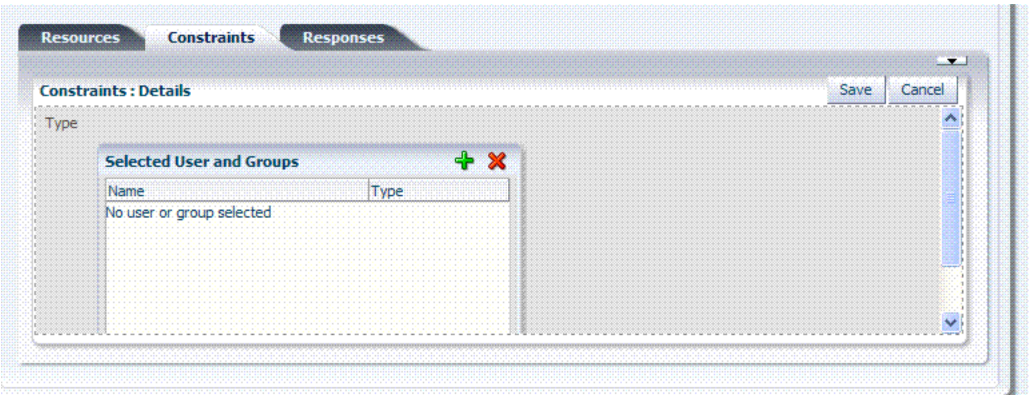
- About Identity Class Constraints
- About IP4Range Class Constraints
- About Temporal Class Constraints

9.10.6 About Identity Class Constraints

With the Identity constraint class, administrators must add a user population to the constraint. After opening the constraint container, any defined user population is displayed. Like the other constraints, this one can be used in conjunction with identity and temporal constraints.

When no user population is specified you see the screen as it appears in Figure 9–18.

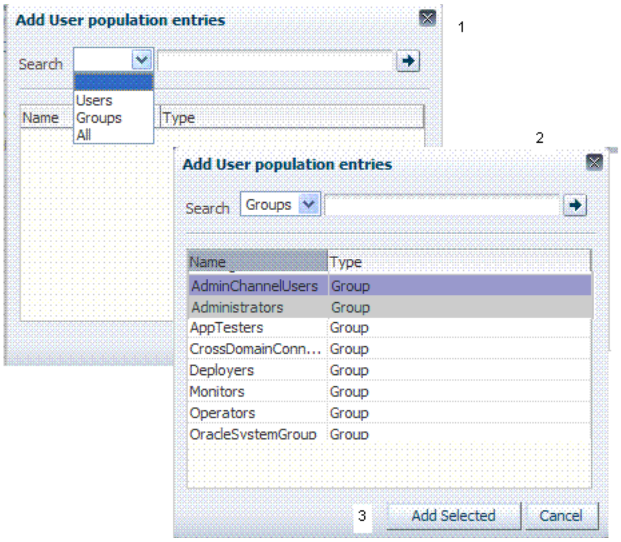
Figure 9–17 Identity Class Constraint Details: Selected User and Groups Table



Name	Type
No user or group selected	

Figure 9–18 shows the Add User Population Entries window, which appears when you click the Add button on the Selected User and Groups table. From this table, you can choose from several populations: Users, Groups, or All.

Figure 9–18 Identity Class Add User Population Entries Window



Name	Type
AdminChannelUsers	Group
Administrators	Group
AppTesters	Group
CrossDomainConn...	Group
Deployers	Group
Monitors	Group
Operators	Group
OracleSystemGroup	Group

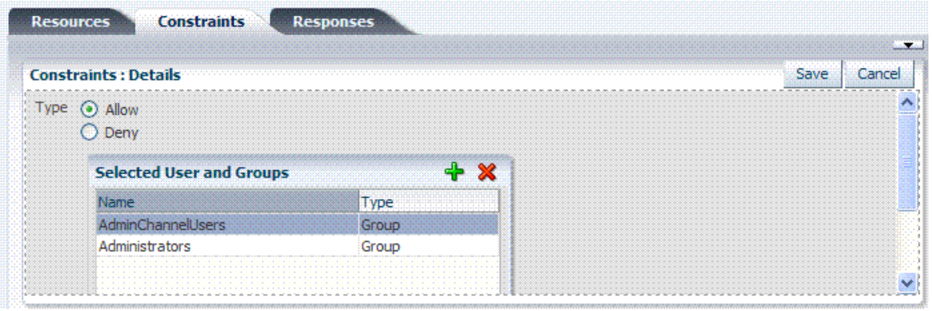
Table 9–15 describes the Add User Population Entries elements.

Table 9–15 Identity Class Constraint Details

Element	Description
Search	List of possible search types: Users, Groups, or All
Text Field	Enter the name of a specific user or group and click the arrow button.
Results table	Displays the results of your search for selection.
Add Selected	Adds the selected users or groups from the results table to the Constraints Details page.

After selecting one or more populations and clicking the Add Selected button, your screen might look something like Figure 9–19.

Figure 9–19 Selected User and Groups Window



Type ● Allow
○ Deny

Name	Type
AdminChannelUsers	Group
Administrators	Group

To save these details as a constraint, click the Save button in the upper-right corner of the details page.

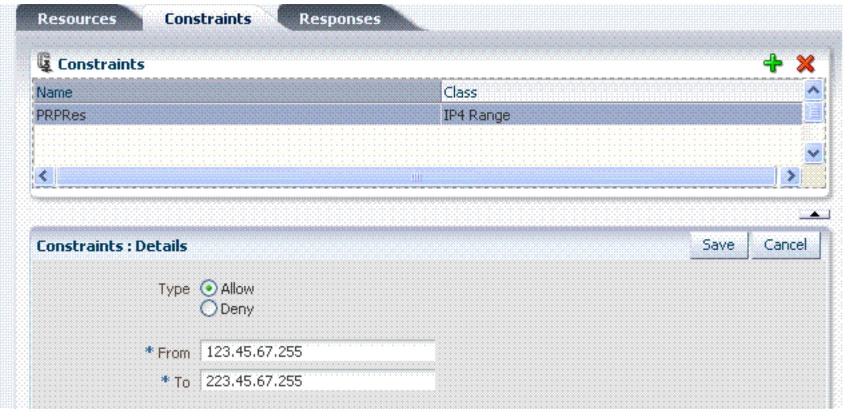
> **See Also:** "Defining Identity Class Constraints" on page 9-42

9.10.7 About IP4Range Class Constraints

With the IP4Range constraint class, administrators must add the range of IP addresses who are either allowed or denied access. Like the other constraints, this one can be used in conjunction with identity and temporal constraints.

Each IP address should be of the format 999.999.999.999 as shown in Figure 9–20.

Figure 9–20 IP4Range Class Constraints



Name	Class
PRPRes	IP4 Range

Type ● Allow ○ Deny

* From 123.45.67.255

* To 223.45.67.255

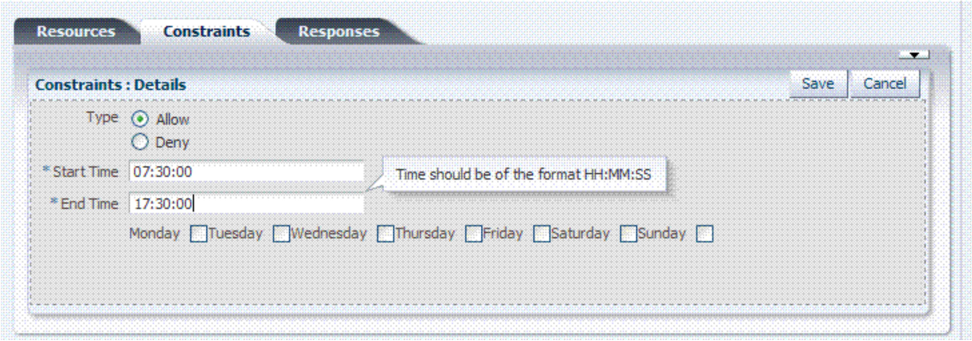
See Also: "Defining IP4Range Class Constraints" on page 9-43

9.10.8 About Temporal Class Constraints

With the Temporal constraint class, administrators must add the start and end time and the range of days. Like the other constraints, this one can be used in conjunction with identity and IP4Range constraints.

By default, all days in the range are enabled (though none are checked in the form as shown in Figure 9–18.

Figure 9–21 Temporal Constraint Class Details Page



Type ● Allow ○ Deny

* Start Time 07:30:00

* End Time 17:30:00

Time should be of the format HH:MM:SS

Monday □ Tuesday □ Wednesday □ Thursday □ Friday □ Saturday □ Sunday □

Time periods must be specified in the HH:MM:SS (hour, minute, and second) format based on a 24-hour clock based on Greenwich Mean Time (GMT). Midnight is specified as 00:00:00 (start). The day ends at 24:59:59.

Table 9–16 Temporal Constraint Class Details

Elements	Description
Type	Outcome type: Allow or Deny access.
Start Time **Notes**: Time is specified using a full 24-hour range. For instance, midnight is specified as 00:00:00 and 11:00 PM is specified as 23:00:00.	Specifies the hour, minute, and second that this constraint begins. **Notes**: Time is based on Greenwich Mean Time (GMT). GMT is the same all year with no adjustments for daylight savings time or summer time.
End Time	Specifies the hour, minute, and second that this constraint concludes.
Days	Specifies the days where this policy is active. Default: AlL Days (even though these are not checked).

Save the details before closing this page.

> **See Also:** "Defining Temporal Class Constraints" on page 9-44

9.11 Defining Authorization Policy Constraints

This section is divided as follows:

- Defining Identity Class Constraints
- Defining IP4Range Class Constraints
- Defining Temporal Class Constraints
- Viewing, Editing, or Deleting Authorization Policy Constraints

9.11.1 Defining Identity Class Constraints

Users with valid OAM Administrator credentials can use the following procedure to add identity class constraints to an application domain.

> **Note:** You must save each constraint definition individually, before adding or selecting another constraint.

Prerequisites

The application domain must already exist.

> **See Also:** "About Identity Class Constraints" on page 9-39

To add identity class constraints to an authorization policy

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain
       Authorization Policies

2. Double-click the desired policy name to open the page (or click the Edit command button in the tool bar).
3. Click the Constraints tab.

4. Click the Add button on the Constraints tab to display the Add Constraint window (Table 9–14) and:

   - In the Name field, enter a unique name for this Constraint.
   - From the Class list, choose Identity as the Constraint type.
   - Click the option button beside the Type (Allow or Deny).
   - Click Add Selected in the Add Constraint window.
   - Highlight the new constraint in the table and then click the Add button in the details table.
   - Add User Population Entries: Choose Users, Groups or All; enter desired search criteria, click the arrow, select desired results, and then click Add Selected.
   - Scroll in the details window to confirm your definition and click Save.
   - Repeat as needed.

5. Click Apply and then close the Confirmation window.
6. Close the page when you finish.
7. Verify the Constraints.

9.11.2 Defining IP4Range Class Constraints

Users with valid OAM Administrator credentials can use the following procedure to add IP-4 Range class constraints to an application domain.

> **Note:** If the user's IP address falls outside the range of denied addresses, this by itself is not enough for authorization to be successful. For authorization to be successful, the user must specifically be granted access based on an Allow rule.

Prerequisites

The application domain must exist.

> **Note:** You must save each constraint definition individually, before adding or selecting another constraint.

See Also: "About IP4Range Class Constraints" on page 9-41

To add IP-4 Range class constraints to an authorization policy

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain
       Authorization Policies

2. Double-click the desired policy name to open the page.
3. Click the Constraints tab.
4. Click the Add button on the Constraints tab to display the Add Constraint window (Table 9–14) and:

- In the Name field, enter a unique name for this Constraint.
- From the Class list, choose IP4Range.
- Click the option button beside the Type (Allow or Deny).
- Click Add Selected in the Add Constraint window.

5. Add the desired IP address range (Table 9–20):
   - Enter the start of the range in the From: field.
   - Enter the end of the range in the To: field.
   - Click the option button beside the Type (Allow or Deny).
   - Click Save.
6. Click Apply and then close the Confirmation window.
7. Verify the IP-4 Range Constraints.

9.11.3 Defining Temporal Class Constraints

Users with valid OAM Administrator credentials can use the following procedure to add temporal class constraints to an application domain.

> **Note:** You must save each constraint definition individually, before adding or selecting another constraint.

Prerequisites

The application domain must exist.

> **See Also:** "About Temporal Class Constraints" on page 9-41

To add temporal class constraints to an authorization policy

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
   Desired Domain
   Authorization Policies

2. Double-click the desired policy name to open the page.
3. Click the Constraints tab.
4. Click the Add button on the Constraints tab to display the Add Constraint window and:
   - In the Name field, enter a unique name for this Constraint.
   - From the Class list, choose Temporal (Table 9–14).
   - Click the option button beside the Type (Allow or Deny).
   - Click Add Selected in the Add Constraint window.
5. Add Temporal Details (Table 9–16): Double-click the name of the constraint, and expand the details panel.
   - Click the option button beside the constraint type (Allow or Deny).
   - Enter the Start time.

- Enter the End time.
- Click the days of the week to which this constraint applies (or leave all blank to specify every day of the week).
- Click Save.
- Repeat as needed.

6. Click Apply and then close the Confirmation window.
7. Verify the Temporal Constraints.

9.11.4 Viewing, Editing, or Deleting Authorization Policy Constraints

Users with valid OAM Administrator credentials can use the following procedure to add identity class constraints to an application domain.

Prerequisites

The application domain and authorization policy already exist.

> **See Also:** "Introduction to Authorization Constraints" on page 9-35

To view, edit, or delete authorization policy constraints

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain
       Authorization Policies

2. Double-click the desired policy name to open the page.
3. Click the Constraints tab.
4. View Constraint Details: Click the row containing the constraint name and view the details in the lower panel.
5. Edit Constraints: Click the row containing the constraint name, change and immediately save the details as described in:
   - "Defining Identity Class Constraints" on page 9-42
   - "Defining IP4Range Class Constraints" on page 9-43
   - "Defining Temporal Class Constraints" on page 9-44
6. Remove constraints: Click the constraint to remove and click the Delete button on the Constraint tab.
7. Click Apply and then close the Confirmation window.
8. Close the page when you finish.
9. Verify the Constraints by accessing the resource and evaluating the results.

9.12 Managing the Common SSO Engine

The SSO Engine is the controller for user sessions. SSO Engine settings common to all OAM Servers in the administration domain.

This section provides the following details:

- About Common SSO Engine Settings

- Viewing or Editing Common SSO Engine Details

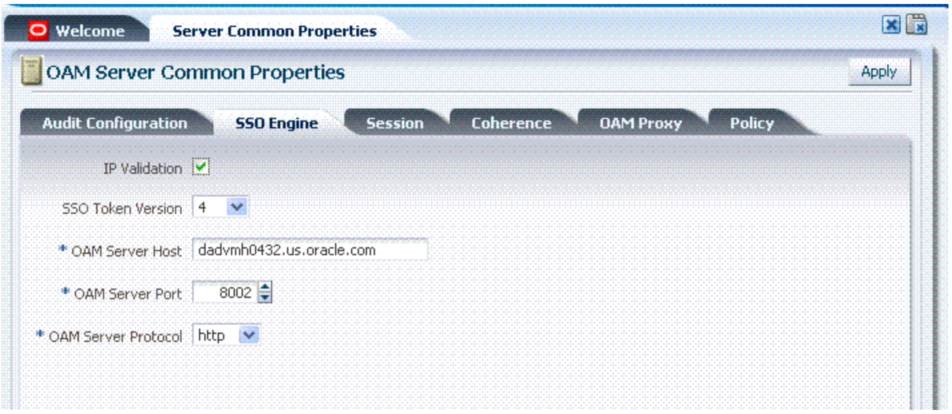
> **See Also:** "Introduction to OAM Server Registration and Management" on page 4-1

9.12.1 About Common SSO Engine Settings

The SSO Engine is the controller for user sessions. SSO Engine settings are global and common to all OAM Servers in the WebLogic administration domain.

Figure 9–22 shows the common SSO Engine settings that are shared by all Oracle Access Manager server instances.

Figure 9–22 Common SSO Engine Settings



IP Validation ☑

SSO Token Version 4

* OAM Server Host dadvmh0432.us.oracle.com

* OAM Server Port 8002

* OAM Server Protocol http

Table 9–17 describes common SSO Engine settings, which include a type (text string) and a value.

Table 9–17 Common SSO Engine Settings

Setting	Description
IP Validation	Specific to WebGates and is used to determine whether a client's IP address is the same as the IP address stored in the ObSSOCookie generated for single sign-on.
SSO Token Version	Select your SSO token version from the list.
OAM Server Host	The port on which the host is listening.
OAM Server Port	The port on which the host is listening.
OAM Server Host Protocol	Either HTTP or HTTPS. See Also: "About Security Modes and X509Scheme Authentication" on page E-3

9.12.2 Viewing or Editing Common SSO Engine Details

Users with valid OAM Administrator credentials can perform the following task to modify common SSO Engine details using the OAM Administration Console.

> **See Also:** "About Common SSO Engine Settings" on page 9-46

To view or edit common SSO Engine details

1. From the System Configuration tab, navigation tree, double-click Server Instances to display the Server Common Properties page.
2. On the Server Common Properties page, click the SSO Engine tab:

- View Only: Close the page when you finish.
- Modify: Perform remaining steps to edit the common SSO Engine configuration.

3. Edit settings as needed for your deployment, based in details in Table 9–17.
4. Click Apply to submit the changes (or close the page without applying changes).
5. Dismiss the Confirmation window.

9.13 Validating Authentication and Authorization in an Application Domain

The procedure here provides several methods for confirming that Agent registration and authentication and authorization policies are operational. The procedures are nearly identical for both OAM Agents and OSSO Agents (mod_osso). However, OSSO Agents use only the authentication policy and not the authorization policy.

Prerequisites

- Users and groups who are granted access must exist in the primary LDAP User Identity Store that is registered with OAM 11g
- Agents must be registered to operate with OAM 11g. After registration, protected resources should be accessible with proper authentication without restarting the Administration or Managed Server.
- Application domain, authentication policies, and authorization policies must be configured.

> **See Also:** Chapter 10, "Validating Connectivity and Policies Using the Access Tester"

To verify authentication and access

1. Using a Web browser, enter the URL for an application protected by the registered Agent to confirm that the login page appears (proving that the authentication redirect URL was specified appropriately). For example:

   ```
   http://myWebserverHost.us.abc.com:8100/resource1.html
   ```

2. Confirm that you are redirected to the login page.
3. On the Sign In page, enter a valid username and password when asked, and click Sign In.
4. Confirm that you are redirected to the resource and proceed as follows:
   - **Success**: If you authenticated successfully and were granted access to the resource; the configuration is working properly.
   - **Failure**: If you received an error during login or were denied access to the resource, check the following:
     - Authentication Failed: Sign in again using valid credentials.
     - **Access to URL ... denied**: This userID is not authorized to access this resource.
     - **Resource not Available**: Confirm that the resource is available.
     - **Wrong Redirect URL**: Verify the redirect URL in the Administration Console.

10
Validating Connectivity and Policies Using the Access Tester

The Oracle Access Manager Access Tester enables IT professionals and administrators to simulate interactions between registered OAM Agents and OAM 11g Servers to help troubleshoot issues involving agent connections and to test policy definitions. This chapter introduces the Oracle Access Manager Access Tester and how to use it. The following topics are provided:

10.1 Prerequisites

Before you can perform tasks in this chapter:

- Ensure that the OAM Administration Console, OAM run-time Server, and registered OAM Agent are running
- Confirm the application domain and policies for one or more resources, as described in Chapter 9.

10.2 Introduction to the OAM 11g Access Tester

The Access Tester is a portable, stand-alone Java application that ships with Oracle Access Manager 11g. The Access Tester provides a functional interface between an individual IT professional or administrator and the OAM Server.

IT professionals can use the Access Tester to verify connectivity and troubleshoot problems with the physical deployment. Application administrators can use the Access Tester to perform a quick validation of policies. In this chapter, the term "administrator" represents any individual who is using the Access Tester.

The Access Tester can be used from any computer, either within or outside the WebLogic Server domain. Both a graphical user interface (known as the Console in this chapter) and a command-line interface are provided. Command line mode enables complete automation of test script execution in single or multi-client mode environments.

By appearing to be a real agent, the Access Tester helps with policy configuration design and troubleshooting, and sometimes with troubleshooting OAM Server responsiveness. When using the Access Tester, you must appear to be the real end user; the Access Tester does not actually communicate with a real end user.

To use the Access Tester, you must understand and administer authentication and authorization policies for an application or resource that is protected by Oracle Access Manager 11g.

The Access Tester enables you to:

- Configure a request to be sent to the OAM Server that emulates what a real agent would send to the OAM Server in a real environment.
- Send your request to the OAM Server and receives a response that is the same as the response that would received by a real Agent. The Access Tester uses the OAM Access Protocol (OAP) API to send requests over the OAP channel to the OAM Proxy running as part of the OAM Server. The OAM Server processes the request and returns a response.
- Process and display the server response.
- Proceed in the manner a real agent would to handle the response. For example, if a WebGate determines that a resource is protected by a certificate authentication scheme, then it must obtain the end user's certificate from the http SSL connection.

  In the case of a certificate authentication scheme, you must point the Access Tester to a certificate to be used as the end user's credentials.

In addition to simulating the Agent while performing functions in the previous list, the Access Tester enables you to:

- Review performance characteristics of intended policy changes
- Track the latency of authentication and authorization requests
- Stress test the OAM Server to establish low- and high-performance watermarks relative to desired user loads, and to size back-end hardware
- Establish performance metrics and measuring on an ongoing basis to prove desired outcomes

During basic operations, the Access Tester does not make any determination about the Server response and whether it is a right or wrong response (for instance, whether or not resource X is protected, or user Y is authorized to access resource X). When operating the Access Tester, you must be aware of the policy configuration to determine if a specific response is appropriate.

The Access Tester offers advanced functionality that enables you to group a number of individual requests into a test script that can be sent to the OAM Server for processing. The output of such a test run can be captured by the Access Tester and used to compare against a similar document containing "known good" responses. In this way, the Access Tester can be used for automated testing of policy configuration against errant changes.

For more information, see the following topics in this chapter:

- About OAM Agent and Server Interoperability
- About Access Tester Security and Processing
- About Access Tester Modes and Administrator Interactions

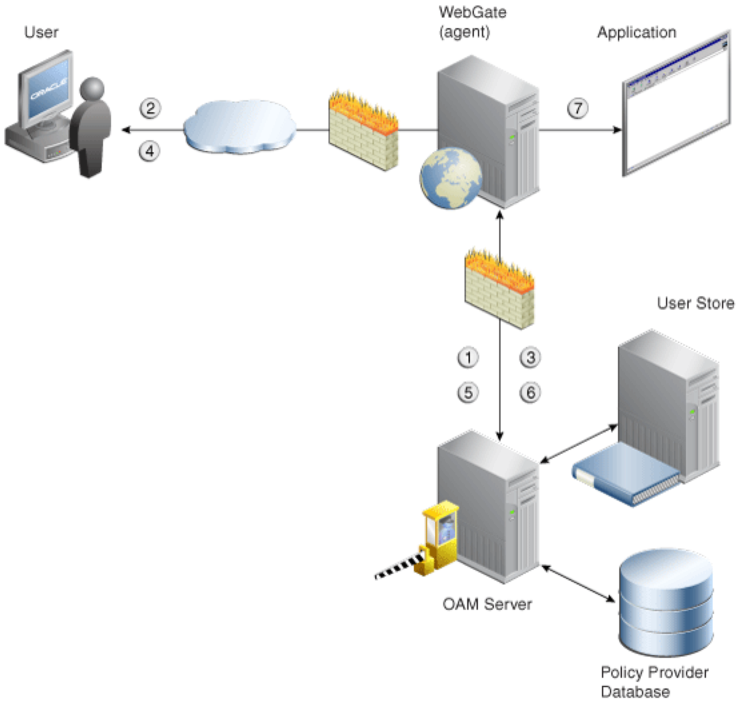
10.2.1 About OAM Agent and Server Interoperability

The two primary types of actors in the OAM architecture are the policy servers (OAM Servers) and OAM policy enforcement agents (WebGates or AccessGates). In the security world, Agents represent the policy enforcement point (PEP), while OAM Servers represent the policy decision point (PDP):

- The Agent plays the role of a gatekeeper to secure resources such as http-based applications and manage all interactions with the user who is trying to access that resource. This is accomplished according to access control policies maintained on the policy server (OAM Server).
- The role of the OAM Server is to provide policy, identity, and session services to the Agent to properly secure application resources, authenticate and authorize users, and manage user sessions.

This core OAM product architecture revolves around the following exchanges, which drive the interaction between the Agent and OAM Server. To expose interoperability and the key decision points, Figure 10–1 illustrates a typical OAM Agent and OAM Server interaction during a user's request for a resource.

Figure 10–1 OAM Agent (PEP) and OAM Server (PDP) Interoperability



The following overview outlines the processing that occurs between OAM Agents and OAM Servers. During testing, the Access Tester emulates the Agent and communicates with the OAM Server while the administrator emulates the end user.

Process overview: Interoperability between OAM Agents and OAM Servers

1. Establish server connectivity: The registered OAM Agent connects to the OAM Server.
2. The user requests accesses to a resource.

3. Validate resource protection: The Agent forwards the request to the OAM Server to determine if the resource is protected.

   Protected: The OAM Server responds with the type of credentials required.

4. User credentials: Establishing the user identity enables tracking for Audit and SSO purposes, and conveyance to the application. For this, the Agent prompts the user for his credentials.

5. Authenticate user credentials: The Agent forwards the supplied user credentials to the OAM Server for validation.

   Authentication Success: The Agent forwards the resource request to the OAM Server.

6. Authorize user access to a resource: The Agents must first determine if the user is allowed to access the resource by forwarding the request for access to the OAM Server for authorization policy evaluation.

7. The Agent grants or denies access based on the policy response.

10.2.2 About Access Tester Security and Processing

The Access Tester supports only Open and Simple connection modes for communication with the OAM Server.

> **Note:** The Access Tester does not currently support OAM Servers and Agents configured for Cert mode transport security.

The Access Tester encrypts all password-type values that it saves to configuration files and test cases. All network connectivity inherits the NetPoint Access Protocol (NAP) limit of a single connection pool (one primary or secondary connection pool).

Persistence: The Access Tester manages a number of data structures that require persistent storage between Access Tester invocations. XML-file-based storage is provided for the following types of information:

- Configuration data to minimize data entry between invocations of the application (OamTestConfiguration)
- Test scripts consisting of captured test cases (OamTestScriptCase)
- Statistical data representing execution metric from a test run (OamTestStats)

XML Files for Input, Logging, and Analysis: The following XML files are produced when you run the Access Tester to process test scripts:

- Configuration Script: config.xml is the output file generated using the Save Configuration command within the Access Tester. The name of this document is used within the input script to provide proper connection information to the Access Tester running in command line mode. For details, see "About the Saved Connection Configuration File" on page 10-30.
- Input Script: script.xml represents a script that is generated by the Access Tester after capturing one or more test cases. For details, see "About the Generated Input Test Script" on page 10-31.
- Target Output Script: oamtest_target.xml is generated by running the Access Tester in command line mode and specifying the input script. For details, see "About the Target Output File Containing Test Run Results" on page 10-32. For example: **`-Dscript.scriptfile="script.xml" -jar oamtest.jar`**

- Statistics: oamtest_stats.xml is generated together with the output script. For details, see "About the Statistics Document" on page 10-34.
- Execution Log: lamtest_log.log is generated together with the output script. For details, see "About the Execution Log" on page 10-36.

For more information, see "About Access Tester Modes and Administrator Interactions".

10.2.3 About Access Tester Modes and Administrator Interactions

In Console mode, the Access Tester provides a single window for interactions with the user. All Access Tester operations are available in the main window, which performs as a central dashboard where users can submit specific details for the test case and view responses.

Alternatively, you can use the Access Tester in command line mode and develop test scripts, which you can run interactively or in batches for computerized execution to maximize productivity and minimize costs and resources.

Run-Time: The Access Tester requires nap-api.jar in the same directory as the main jar oamtest.jar. Starting the application requires oamtest.jar.

Regardless of the mode you choose for running the Access Tester, your primary interactions with the Access Tester include:

- Issuing Requests and Reviewing Results

  You use the Access Tester to issue requests to the OAM Server to validate resource protection, policy configuration, user authentication, and user authorization. You can immediately analyze test case results and also retain the data for longer-term analysis, if needed.

- Managing Test Scripts

  You can build test scripts by capturing the data generated by test execution, which is available as stand-alone documents. You can run the test script for manual or automated analysis. The Access Tester provides for some automated analysis after each test run, while collecting full set of statistics to enable analysis after the fact.

- Managing OAM Server Connectivity

  You can manage application settings that include server connection information.

Figure 10–2 depicts the flow of information during operations in both Console and command-line modes. Details follow the figure. Advanced operations include building and executing test scripts.

Note:

Figure 10–2 User Interactions with the Access Tester

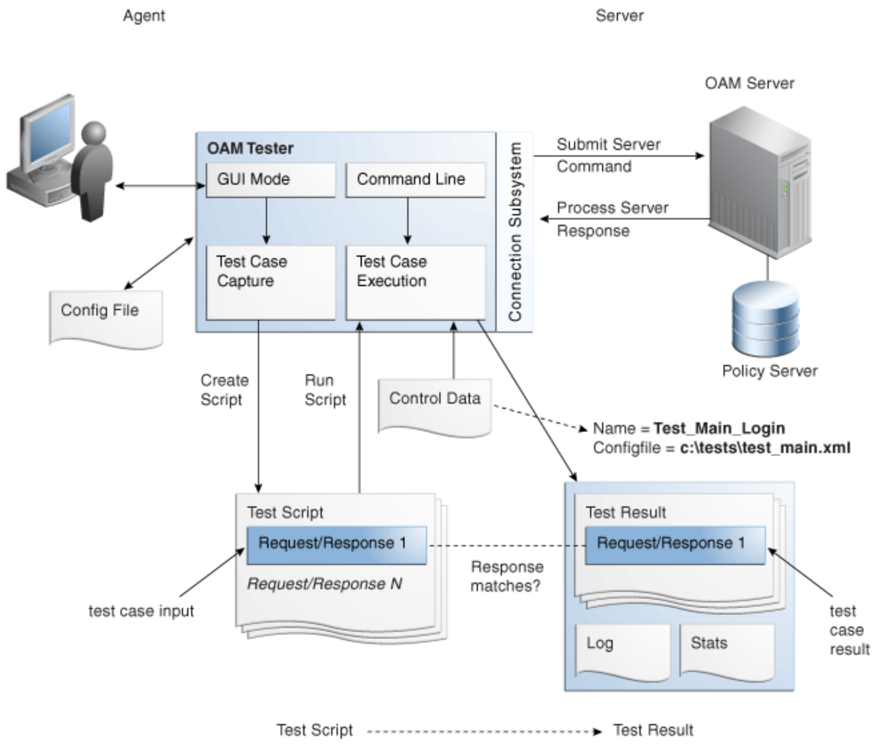


Table 10–1 describes the process flow of information during both Console mode operations and command-line mode operations.

Table 10–1 User Interactions Using Console Mode versus Command Line Mode Operations

Console mode	**Command Line Mode**
The user starts the Access Tester from the command line.	The user or a shell script starts the Access Tester in command line mode.
The user opens a previously saved configuration file to populate the application fields and minimize data entry, including server connection fields. **Alternatively**, the user can use the Console and enter data manually	The Access Tester starts processing test cases based on the input script.
The user clicks the Connect button to open the connection with the OAM Server.	The Access Tester opens a connection with the OAM Server based on details in the input script.
Resource Protection: The user performs steps in a sequence to validate resource protection, authenticate user credentials, and authorize user access.	Resource Protection: The Access Tester starts processing test cases based on the input script.
When the test completes, the Access Tester generates: ▪ A script with results ▪ A file with execution statistics including information about mismatched responses ▪ A log file detailing processing flow	Once the script completes, the Access Tester generates: ▪ A script with results ▪ A file with execution statistics including information about mismatched responses ▪ A log file detailing processing flow
The user repeats steps as needed to complete validation	The user repeats steps as needed to complete validation.

The following overview outlines the tasks involved with using the Access Tester, and the topics where more information can be found in this chapter.

Task overview: Testing OAM 11g connections and policies includes

1. Review the following topics:
   - Installing and Starting the Access Tester
   - Introduction to the Access Tester Console and Navigation
2. Perform and capture tests using the Access Tester Console as described in "Testing Connectivity and Policies from the Access Tester Console":
3. Proceed to "Creating and Managing Test Cases and Scripts"

10.3 Installing and Starting the Access Tester

The Access Tester consists of two jar files that can be used from any computer, either within or outside the WebLogic Server domain. This section describes how to install the Access Tester, which involves copying the Access Tester jar files to a computer from which you want to run tests. The Access Tester must be started from a command line regardless of the mode you choose for test input: Console mode or command line mode. This section is divided into the following topics:

- Installing the Access Tester
- About Access Tester Supported System Properties
- Starting the Access Tester Without System Properties For Use in Console Mode
- Starting the Access Tester with System Properties For Use in Command Line Mode

10.3.1 Installing the Access Tester

This topic describes how to install the Access Tester for use on any computer. Following installation, the Access Tester is ready to use. No additional setup is required.

To install the Access Tester

1. Ensure that the computer from which the tester will be run includes JDK/JRE 6. For example, you can test for Java as follows:

   ```
   java -version
   ```

   The previous command returns the following information:

   ```
   java version "1.6.0_18"
   Java(TM) SE Runtime Environment (build 1.6.0_18-b07)
   Java HotSpot(TM) Client VM (build 16.0-b13, mixed mode)
   ```

2. On a computer hosting the OAM Server, locate and copy the Access Tester Jar files. For example:

   ```
   Oracle_HOME/oam/server/tester/oamtest.jar
   Oracle_HOME/oam/server/tester/nap-api.jar
   ```

3. Store the jar file copies together in the same directory on any computer from which you want to run the Access Tester.
4. Proceed as follows, depending on your environment and requirements:
   - Starting the Access Tester Without System Properties For Use in Console Mode enables you to manually drive requests.

- Starting the Access Tester with System Properties For Use in Command Line Mode
- Executing a Test Script enables you to use a test script that has been created against a "Known Good" policy configuration and marked as "Known Good"

10.3.2 About Access Tester Supported System Properties

The Access Tester supports a number of configuration options that are used for presentation or during certain aspects of testing. These options are specified at startup using the Java-D mechanism, as shown in Table 10–2, which describes all supported system properties.

Table 10–2 Access Tester Supported System Properties

Property	Access Tester Mode	Description and Command Syntax
log.traceconnfile	Console and Command Line modes	Logs connection details to the specified file name. -Dlog.traceconnfile="<file-name>"
display.fontname	Console mode	Starts the Access Tester with the specified font. This could be useful in compensating for differences in display resolution. - Ddisplay.fontname ="<font-name>"
display.fontsize	Console mode	Starts the Access Tester with the specified font size. This could be useful in compensating for differences in display resolution. - Ddisplay.fontsize ="<font-size>"
display.usesystem	Console mode	Starts the Access Tester with the default font name and size (Dialog font, size 10). - Ddisplay.usesystem
script.scriptfile	Command Line mode	Runs the script <file-name> in command line mode. -Dscript.scriptfile="<file-name>"
control.configfile	Command Line mode	Overwrites script's "configfile" attribute containing the absolute path to the configuration XML file with the connection information. The Access Tester uses the configuration file to establish a connection to the Policy Server indicated by Connection element. -Dcontrol.config="<file-name>"
control.testname	Command Line mode	Overwrites script's "testname" attribute of the Control element containing a string representing a name of the test series to be used in naming output script, stats, and log files. Output log files begin with <testname>_ <testnumber>. -Dcontrol.testname="<String>"

Table 10–2 (Cont.) Access Tester Supported System Properties

Property	Access Tester Mode	Description and Command Syntax
control.testnumber	Command Line mode	Specifies the control number to be used in naming output script, stats, and log files. Output log files begin with <testname>_ <testnumber>. -Dcontrol.testnumber="<String>". Although the auto generated string is a 7 digit number based on current local time (2 character minutes + 2 character seconds + 3 character hundredths), any string can be used to denote the control number as long as it can be used in a filename.
control.ignorecontent	Command Line mode	Overwrites script's "ignorecontent" attribute of the Control element indicating the Access Tester should ignore differences in Content between the original test case and current results. -Dcontrol.testname="true | false"
control.loopback	Command Line mode	Runs the Access Tester in loopback mode to test the Access Tester for internal regressions against a known good script. Used for unit testing the Access Tester. -Dcontrol.loopback="true"

10.3.3 Starting the Access Tester Without System Properties For Use in Console Mode

To manually drive (and capture) requests and view real-time response through the graphical user interface, start the tester in Console mode. This procedure omits all system properties, even though several can be used with Console mode.

The jar file defines the class to be started by default; no class name need be specified. Ensure that the nap-api.jar is present in the same directory as oamtest.jar.

See Also:

- "About Access Tester Supported System Properties"
- "Starting the Access Tester with System Properties For Use in Command Line Mode"

To start the Access Tester in console mode without system properties

1. From the directory containing the Access Tester jar files, enter the following command:

```
java -jar oamtest.jar
```

2. Proceed to one of the following topics for more information:
   - Introduction to the Access Tester Console and Navigation
   - Testing Connectivity and Policies from the Access Tester Console
   - Creating and Managing Test Cases and Scripts

10.3.4 Starting the Access Tester with System Properties For Use in Command Line Mode

This section is divided into the following topics:

- About the Access Tester Command Line Mode
- Starting the Access Tester Without System Properties For Use in Console Mode

10.3.4.1 About the Access Tester Command Line Mode

To run a test script, or to customize Access Tester operations, you must start the tester in command line mode and include system properties using the Java -D option.

> **See Also:** "About Access Tester Supported System Properties" on page 10-8

When running in command line mode, the Access Tester returns completion codes that can be used by shell scripts to manage test runs. When you run the Access Tester in Console mode, you do not need to act upon codes that might be returned by the Access Tester.

Shell scripts that wrap the Access Tester to execute specific test cases must be able to recognize and act upon exit codes communicated by the Access Tester. In command line mode, the Access Tester exits using System.Exit (N), where N can be one of the following codes:

- 0 indicates successful completion of all test cases with no mismatches. This also includes a situation where no test cases are defined in the input script.
- 3 indicates successful completion of all test cases with at least one mismatch.
- 1 indicates that an error prevented the Access Tester from running or completing test cases. This includes conditions such as No input script specified, Unable to read the input script, Unable to establish server connection, Unable to generate the target script.

These exit codes can be picked up by shell scripts ($? In Bourne shell) designed to drive the Access Tester to execute specific test cases.

10.3.4.2 Starting the Access Tester with System Properties

Use the following procedure to start the Access Tester in command line mode and specify any number of configuration options using the Java-D mechanism.

> **See Also:** "About Access Tester Supported System Properties" on page 10-8

To start the Access Tester with system properties or for use in command line mode

1. From the directory containing the Access Tester jar files, enter the command with the appropriate system properties for your environment. For example:

   ```
   java -Dscript.scriptfile="\tests\script.xml" -Dcontrol.ignorecontent="true"
   -jar oamtest.jar
   ```

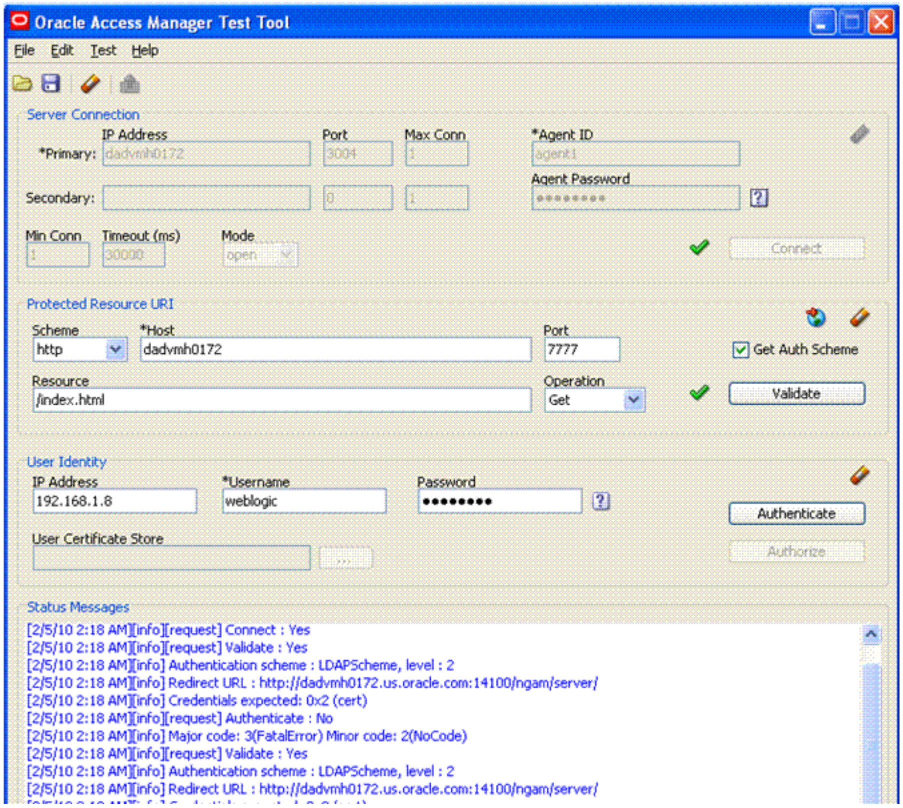
2. After startup, proceed to one of the following topics for more information:
   - Testing Connectivity and Policies from the Access Tester Console
   - Creating and Managing Test Cases and Scripts

10.4 Introduction to the Access Tester Console and Navigation

This section introduces the Access Tester Console, navigation, and controls.

Figure 10–3 shows the fixed-size Access Tester Console. This is the window through which users can interact with the application if the Access Tester is started in Console mode. The window can not be resized. Details follow the screen.

Figure 10–3 Access Tester Console



Server Connection

	IP Address	Port	Max Conn
*Primary:	dadvmh0172	3004	1
Secondary:		0	1

*Agent ID: agent1

Agent Password: ••••••••

Min Conn: 1

Timeout (ms): 30000

Mode: open

Protected Resource URI

Scheme: http

*Host: dadvmh0172

Port: 7777

☑ Get Auth Scheme

Resource: /index.html

Operation: Get

User Identity

IP Address: 192.168.1.8

*Username: weblogic

Password: •••••••••

User Certificate Store:

Status Messages

```
[2/5/10 2:18 AM][info][request] Connect : Yes
[2/5/10 2:18 AM][info][request] Validate : Yes
[2/5/10 2:18 AM][info] Authentication scheme : LDAPScheme, level : 2
[2/5/10 2:18 AM][info] Redirect URL : http://dadvmh0172.us.oracle.com:14100/ngam/server/
[2/5/10 2:18 AM][info] Credentials expected: 0x2 (cert)
[2/5/10 2:18 AM][info][request] Authenticate : No
[2/5/10 2:18 AM][info] Major code: 3(FatalError) Minor code: 2(NoCode)
[2/5/10 2:18 AM][info][request] Validate : Yes
[2/5/10 2:18 AM][info] Authentication scheme : LDAPScheme, level : 2
[2/5/10 2:18 AM][info] Redirect URL : http://dadvmh0172.us.oracle.com:14100/ngam/server/
```

At the top of the main window are the menu names within a menu bar. Under the menu bar is the tool bar. All of the commands represented by buttons in the tool bar are also available as menu commands.The Access Tester Console is divided into four panels, described in Table 10–3.

Table 10–3 Access Tester Console Panels

Panel Name	Description
Server Connection	Provides fields for the information required to establish a connection to the OAM Server (a single primary server and a single secondary server), and the Connect button: See also: "Establishing a Connection Between the Access Tester and the OAM Server" on page 10-14.
Protected Resource URI	Provides information about a resource whose protected status needs to be validated. The Validate button is used to submit the Validate Resource server request. See also: "Validating Resource Protection from the Access Tester Console" on page 10-16.

Table 10–3 (Cont.) Access Tester Console Panels

Panel Name	Description
User Identity	Provides information about a user whose credentials need to be authenticated. The Authenticate button is used to submit the Authenticate User server request. See also: "Testing User Authentication from the Access Tester Console" on page 10-19.
Status Messages	Provides a scrollable status message area containing messages displayed by the application in response to user gestures. The Authorize button is used to submit the Authorize User server request. See also: "Observing Request Latency" on page 10-22.

Text fields support right-clicking to display the Edit menu and drag-and-drop operations using the mouse and cursor.

There are four primary buttons through which you submit test requests to the OAM Server. Each button acts as a trigger to initiate the named action described in Table 10–4.

Table 10–4 Command Buttons in Access Tester Panels

Panel Button	Description
Connect	Submits connection information and initiates connecting.
Validate	Submits information provided in the Protected Resource URI panel and initiates validation of protection.
Authenticate	Submits information provided in the User Identity panel and initiates authentication confirmation.
Authorize	Submits information provided in the User Identity panel and initiates authorization confirmation.

See Also: "Access Tester Menus and Command Buttons"

10.4.1 Access Tester Menus and Command Buttons

Table 10–5 identifies additional Access Tester Console buttons and their use. All command buttons provide a tip when the cursor is on the button.

Table 10–5 Additional Access Tester Buttons

Command Buttons	Description
📂	Loads connection configuration details that were saved to an XML file (config.xml, by default). You can refresh the information in the Console by clicking this button.
💾	Saves connection configuration details to a file (default name, config.xml). You can add the name of this document to the input script to provide proper connection information to the Access Tester running in command line mode. The Save command button at the bottom of the Console saves the content of the Status Message panel to a log file.
✎	Clears fields on a panel containing the icon. Tool bar action clears all fields except connection fields if the connection has already been established.

Table 10–5 (Cont.) Additional Access Tester Buttons

Command Buttons	Description
⇧ ⇧	Captures the last named request to the capture queue with the corresponding response received from the OAM Server. Together, the request and response create a test case. The capture queue status at the bottom of the Console is updated to reflect the number of test cases in the queue. You can save the contents of the capture queue to create a test script containing multiple test cases using the Generate Script command on the Test menu or a command button.
	Generates a test script that includes every test case currently in the capture queue, and asks if the queue should be cleared. Do not clear the queue until all your test cases have been captured and saved to a test script.
	Runs a test script against the current OAM Server. The Status message window is populated with the execution status as the script progresses through each test case.
	Imports a copied URI from the clipboard after parsing it to populate fields in the URI panel.
?	Displays a dialog showing the password in clear text

The Access Tester provides the menus described in Table 10–6. All menu items have mnemonics that are exposed by holding down the ALT key (on Windows systems). There are also command accelerators (keyboard activation) available using the CTRL-<KEY> combination defined for each menu command.

Table 10–6 Access Tester Menus

Menu Title	Menu Commands
File	▪ Open Configuration ▪ Save Configuration ▪ Exit **Note**: To minimize the amount of data entry the Save Configuration and Open Configuration menu (and tool bar command buttons) allow for specific Connection, URI, and Identity information to be saved to (and read from) a file. Thus, it becomes fairly simple to manage multiple configurations. Also, the configuration file can be used as input to the Access Tester when you run it in command line mode and execute a test script.
Edit	Provides standard editing commands, which act on fields: ▪ Cut ▪ Copy ▪ Paste ▪ Clear all fields ▪ Import URI fields from a saved URL
Test	▪ Capture last "..." request (for example, Capture last "authorize" request) ▪ Save test script ▪ Run test script **Note**: You can use functions here to capture the last request and response to create a test case that you can save to a test script to be run at a later time.
Help	

10.5 Testing Connectivity and Policies from the Access Tester Console

This section describes how to perform quick spot checks using the Access Tester in Console mode with OAM Servers.

Spot checks or troubleshooting connections between the Agent and OAM Server can help you assess whether the Agent can communicate with the OAM Server, which is especially helpful after an upgrade or product migration. Spot checks or troubleshooting resource protection that can be exercised by Agents and OAM Servers can help you develop end-to-end tests of policy configuration during the application lifecycle.

The following overview identifies the tasks and sequence to be performed and where to locate additional information about each task.

> **Note:** You can capture each request and response pair to create a test case, and save the test cases to a script file that can be run later. For details, see "Creating and Managing Test Cases and Scripts" on page 10-22.

Task overview: Performing spot checks from the Access Tester Console

1. Start the Access Tester, as described in "Installing and Starting the Access Tester" on page 10-7.
2. Add relevant details to the Server Connection panel and click Connect, as described in "Establishing a Connection Between the Access Tester and the OAM Server" on page 10-14.
3. Enter or import details into the Protected Resource URI pane and click Validate, as described in "Validating Resource Protection from the Access Tester Console" on page 10-16.
4. Add relevant details to the User Identity panel and click Authenticate, as described in "Testing User Authentication from the Access Tester Console" on page 10-19.
5. After successful authentication, click Authorize in the User Identity panel, as described in "Testing User Authorization from the Access Tester Console" on page 10-21.
6. Check the latency of requests, as described in "Observing Request Latency" on page 10-22.

10.5.1 Establishing a Connection Between the Access Tester and the OAM Server

Before you can send a request to the OAM Server you must establish a connection between the Access Tester and the server. This section describes how to establish that connectivity.

- About the Connection Panel
- Connecting the Access Tester with the OAM Server

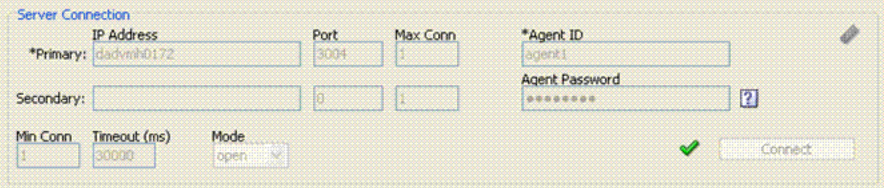
10.5.1.1 About the Connection Panel

You enter required information for the OAM Server and the Agent you are emulating in the Access Tester Connection panel and then click the Connect button. The Tester initiates the connection, and displays the status in the Status Messages panel. Once the connection is established, it is used for all further operations.

Caution: Once the connection is established, it cannot be changed until you restart the Access Tester Console.

Figure 10–4 illustrates the Server Connection panel and controls.

Figure 10–4 Server Connection Panel in the Access Tester



Server Connection

IP Address / Port / Max Conn

*Primary: dadvmh0172 / 3004 / 1

Secondary: / 0 / 1

*Agent ID: agent1

Agent Password: ********

Min Conn: 1

Timeout (ms): 30000

Mode: open

Table 10–7 describes the information needed to establish the connection. The source of your values is the OAM Administration Console, System Configuration tab.

Table 10–7 Connection Panel Information

Fields	Description
IP Address	The IP Address of the Primary and Secondary OAM Proxy listens on for this set of tests. **Note**: Oracle recommends that you enter values for only the Primary OAM Proxy. The Secondary OAM Proxy is needed only if you want to test failover between the primary and secondary OAM Server. However, a more practical use of the Secondary Server is reserved for later use, when the OAP API supports load balancing between Primary and Secondary OAM Server.
Port	Enter the port number of the Primary and Secondary OAM Server.
Max Conn	The maximum number of physical connection (TCP) sockets the Access Tester will use. Access Tester emulates a single threaded Agent. **Note**: Oracle recommends that you accept the default value, 1.
Min Conn	The minimum number of physical connection (TCP) sockets the Access Tester will use. The Access Tester emulates a single threaded Agent. **Note**: Oracle recommends that you accept the default value, 1.
Timeout	The number of milliseconds the Access Tester should wait for the connection to be established or to receive a response from the OAM Server. **Note**: Oracle recommends that you accept the default value.
Mode	The level of communication security that is designated for the Agent to be emulated. ▪ Open--No communication security set between the Agent and OAM Server. ▪ Simple--The physical connection is encrypted using built-in certificates. Choosing Simple presents a field for the global pass phrase set the OAM Server. ▪ Cert--The physical connection is encrypted using customer-supplied certificates. Choosing Cert presents a button that opens a dialog for the certificate information.
Agent ID	Enter the identity of the OAM Agent the Tester is simulating.
Agent Password	Enter the password for the OAM Agent the Tester is simulating, if there is one configured.
✓	The green check mark beside the Connect button indicates a "Yes" response; the connection is made. The Status Messages panel also indicates a "Yes" response for the connection.

Table 10–7 (Cont.) Connection Panel Information

Fields	Description
⊗	The red circle beside the Connect button indicates a "No" response; no connection exists. The Status Messages panel also indicates a "No" response for the connection.

After entering information and establishing a connection, you can save details to a configuration file that can be re-used later.

> **See Also:** "Establishing a Connection Between the Access Tester and the OAM Server"

10.5.1.2 Connecting the Access Tester with the OAM Server

Use the following procedure to submit your connection details for the OAM Server.

Prerequisites

Installing and Starting the Access Tester

> **See Also:** "About the Connection Panel"

To test connectivity between the Access Tester and the OAM Server

1. In the Server Connection Panel (Table 10–7), enter:
   - Primary and secondary OAM Proxy details
   - Timeout period
   - Communication encryption mode
   - Agent details
2. Click the Connect button.
3. Beside the Connect button, look for the green check mark indicating the connection is established.
4. In the Status Messages panel, verify a Yes response.

   If the connection still cannot be made, start the Access Tester Console using the Trace Connection command mode and look for additional details in the connection log. Also, ask the OAM administrator of the OAM Server to review the policy server log.
5. Save Good Connection Details: From the Test menu, click the Generate a Script command and enter a name for this configuration file (or use the default name, config.xml).

10.5.2 Validating Resource Protection from the Access Tester Console

Before a user can access a resource, the Agent must first validate that the resource is protected. Using the Access Tester, you can act as the Agent to have the OAM Server validate whether or not the given URI is protected and communicate the response to the Access Tester, as described here.

- About the Protected Resource URI Panel
- Validating Resource Protection

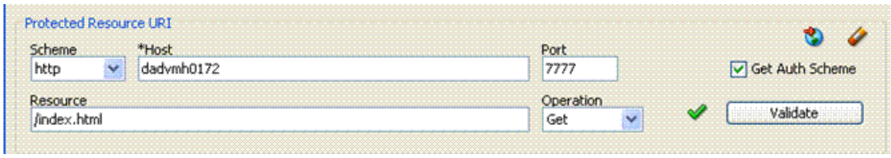
10.5.2.1 About the Protected Resource URI Panel

You must enter required information for the resource you want to validate in the Access Tester Protected Resource URI panel, and then click the Validate button.

To minimize data entry, you can import long URIs that you have copied from a browser and then click the Import URI command button. The Tester parses the URI saved to the clipboard and populates the URI fields in the Access Tester.

Figure 10–5 illustrates the panel where you enter the URI details to validate that the resource is protected. When combined, the URI fields follow RFC notation. For example: *`http://oam_server1:7777/index.html`*.

Figure 10–5 Protected Resource URI Panel in the Access Tester



Protected Resource URI

Scheme: http

*Host: dadvmh0172

Port: 7777

☑ Get Auth Scheme

Resource: /index.html

Operation: Get

Table 10–8 describes the information needed to perform this validation.

Table 10–8 Protected Resource URI Panel Fields and Controls

Field or Control	**Description**
Scheme	Enter http or https, depending on the communication security specified for the resource. **Note**: The Access Tester supports only http or https resources. You cannot use the Access Tester to test policies that protect custom non-http resources.
Host	Enter a valid host name for the resource. **Note**: Your <*host:port*> combination specified in the Access Tester must match one of the Host Identifiers defined in the OAM Administration Console. If the host identifier is not recognized, OAM cannot validate resource protection.
Port	Enter a valid port for the URI. **Note**: The <host:port> combination specified in the Access Tester must match one of the Host Identifiers as defined in the OAM Server. If the host identifier is not recognized, OAM cannot validate resource protection.
Resource	Enter the Resource component of the URI (/index.htm in the example). This resource should match a resource defined for an authentication and authorization policy in the OAM Administration Console. **Note**: If protected, the resource identifier that you provide here must match the one specified in an authorization policy in the OAM Administration Console.
🌐	Click this button to parse and import a URI that is saved on a clipboard.
Operation	Select the operational component of the URI from the list provided in the Access Tester. The OAM Server does not distinguish between different actions, however. Therefore, leaving this set to Get should suffice.

Table 10–8 (Cont.) Protected Resource URI Panel Fields and Controls

Field or Control	Description
Get Auth Scheme	Check this box to request the OAM Server to return details about the Authentication Scheme that is used to secure the protected resource. If the URI is protected, this information is displayed in the Status Messages panel.
Validate	Click the Validate button to submit the request to the OAM Server. When the response is received, the Access Tester displays it in the Status Messages panel.
✓	A green check mark appearing beside the Validate button indicates a "Yes" response; the resource is protected. The Status Messages panel provides the redirect URL for the resource and that credentials are expected. **Note**: If you checked the Get Auth Scheme box, the name and level of the Authentication Scheme that protects this resource are also provided in the Status Messages panel.
⊗	A red circle appearing beside the Validate button indicates that the resource is not protected. A No response will also appear in the Status Messages.

You can capture each request and response pair to create a test case, and save multiple test cases to a script file that can be run later.

See Also:

- "Validating Resource Protection from the Access Tester Console"
- "Creating and Managing Test Cases and Scripts" on page 10-22

10.5.2.2 Validating Resource Protection

Use the following procedure to submit your resource information to the OAM Server and verify responses in the Status Messages panel.

Prerequisites

Establishing a Connection Between the Access Tester and the OAM Server

See Also: "About the Protected Resource URI Panel"

To confirm that a resource is protected

1. In the Access Tester Protected Resource URI panel, enter or import your own resource information (Table 10–8).
2. Click the Validate button to submit the request.
3. Review Access Tester output, including the relevant data about the resource such as how the resource is protected, level of protection, and so on.
4. Beside the Validate button, look for the green check mark indicating the resource is protected.
5. In the Status Messages panel, verify the redirect URL, authentication scheme, and that credentials are expected.
6. Capture the request and response to create a test case for use later, as described in "Creating and Managing Test Cases and Scripts" on page 10-22.
7. Retain the URI to minimize data entry and server processing using one of the following methods.

8. Proceed to "Testing User Authentication from the Access Tester Console"

10.5.3 Testing User Authentication from the Access Tester Console

This topic provides the following information:

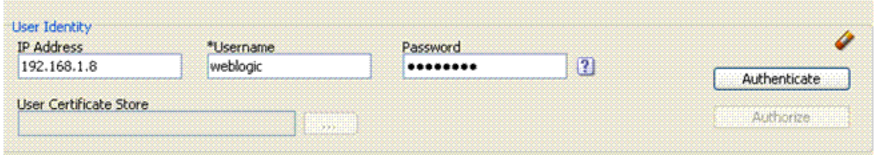
- About the User Identity Panel
- Testing User Credential Authentication

10.5.3.1 About the User Identity Panel

Before a user can access a resource, the Agent must validate the user's identity based on the defined authentication policy on the OAM Server. Using the Access Tester, you can act as the Agent to have the OAM Server authenticate a specific userID for the protected resource. All relevant authentication responses are considered during this policy evaluation.

Figure 10–6 illustrates the Access Tester panel where you enter the information needed to test authentication.

Figure 10–6 Access Tester User Identity Panel



User Identity

IP Address: 192.168.1.8

*Username: weblogic

Password: ••••••••

User Certificate Store

Table 10–9 describes the information you must provide.

Table 10–9 Access Tester User Identity Panel Fields and Controls

Field or Control	Description
IP Address	Enter the IP Address of the user whose credentials are being validated. All Agents communicating with the OAM Server send the IP address of the end user. Default: The IP address that is filled in belongs to the computer from which the Access Tester is run. To test a policy that requires a real user IP address, replace the default IP address with the real IP address.
User Name	Enter the userID of the individual whose credentials are being validated. Note: The Access Tester enables (or disables) the user name and password fields if the resource is protected by an authentication scheme that requires those credentials. Otherwise, this field is disabled.
Password	Enter the password of the individual whose credentials are being validated.
?	Click this button to display the password in clear text within a popup window.
User Certificate Store	The file (in PEM format) containing the X.509 certificate of the user whose credentials should be authenticated. If the URI is protected by the X.509 Authentication Scheme, the Access Tester uses the PEM-formatted X.509 certificate as a credential instead of (or in addition to) the username/password. If the the Authentication Scheme does not require an X.509 certificate, this field is disabled. Note: For certificate-based authentication to work, the OAM Server must be properly configured with root CA certificates and SSL keystore certificates. See Appendix E for details about securing communication between OAM 11g Servers and WebGates.

Table 10–9 (Cont.) Access Tester User Identity Panel Fields and Controls

Field or Control	Description
...	Click this button to browse the file system for the user certificate store path.
Authenticate	Click the Authenticate button to submit the request to the OAM Server and look for a response in the Status Messages panel. Note: The type of credentials supplied (username/password or X.509 certificate) must match the requirements of the authentication scheme that protects the URI.
Authorize	After the user's credentials are validated, you can click the Authorize button to submit the request for the resource to the OAM Server. Check the Status Messages panel for a response.
✓	A green check mark appearing beside the Authenticate button indicates authentication success; The Status Messages panel also indicates "yes" authentication was successful, and provides the user DN and session id. A green check mark appearing beside the Authorize button indicates authorization success; The Status Messages panel also indicates "yes" authorization was successful, and provides application domain details.
⊗	A red circle appearing beside the Authenticate button indicates authentication failure; The Status Messages panel also indicates "no" authentication was not successful. A red circle appearing beside the Authorize button indicates authorization failure; The Status Messages panel also indicates "no" authorization was not successful.

You can capture each request and response pair to create a test case, and save multiple test cases to a script file that can be run later.

See Also:

- "Testing User Authentication from the Access Tester Console"
- "Creating and Managing Test Cases and Scripts" on page 10-22

10.5.3.2 Testing User Credential Authentication

Use the following procedure to submit the end user credentials to the OAM Server and verify authentication. All relevant authentication responses are considered during this policy evaluation.

Prerequisites

Validating Resource Protection from the Access Tester Console with URI information retained in the Console

See Also: "About the User Identity Panel"

To test user credential authentication

1. In the Access Tester User Identity panel, enter information for the user to be authenticated (Table 10–9).
2. Click the Authenticate button to submit the request.
3. Beside the Authenticate button, look for the green check mark indicating the user is authenticated.

   Not Successful: Confirm that you entered the correct userID and password and try again. Also, check the OAM Administration Console for an active user session that you might need to end, as described in Chapter 12.

4. Capture the request and response to create a test case for use later, as described in "Creating and Managing Test Cases and Scripts" on page 10-22.
5. Retain the URI and user identity information and proceed to "Testing User Authorization from the Access Tester Console".

10.5.4 Testing User Authorization from the Access Tester Console

Before a user can access a resource, the Agent must validate the user's permissions based on defined policies on the OAM Server. Using the Access Tester, you can act as the Agent to have the OAM Server validate whether or not the authenticated user identity can be authorized to access the resource.

Use the following procedure to verify the authenticated end user’s authorization for the resource. All relevant authorization constraints and responses are considered during this policy evaluation.

Prerequisites

Testing User Authentication from the Access Tester Console with all information retained in the Console

See Also: "About the User Identity Panel"

Note: Once the protected resource URI is confirmed and the user’s identity is authenticated from the Access Tester, no further information is needed. You simply click the Authorize button to submit the request. However, if the resource is changed to another you must start the sequence anew and validate, then authenticate, and then authorize.

To test user authorization

1. In the Access Tester User Identity panel, confirm the user is authenticated (Table 10–9).
2. In the Access Tester User Identity panel, click the Authorization button.
3. Beside the Authorization button, look for the green check mark indicating the user is authorized.

   Not Successful: Confirm the authorization policy using the OAM Administration Console.
4. In the Status Messages panel (or execution log file), verify details about the test run.
5. Capture the request and response to create a test case for use later, as described in "Creating and Managing Test Cases and Scripts" on page 10-22.
6. Proceed to:
   - Observing Request Latency
   - Creating and Managing Test Cases and Scripts
   - Evaluating Scripts, Log File, and Statistics

10.5.5 Observing Request Latency

To understand OAM Server performance you must know how well the OAM Server handles requests passed by the Agent. While there are many ways to expose a server's metrics, it is sometimes useful to expose server performance from the standpoint of the Agent. Using the Access Tester, you can do just that as described here.

Prerequisites

"Installing and Starting the Access Tester" on page 10-7

Task overview: Observing request latency includes

1. "Validating Resource Protection" on page 10-18
2. "Testing User Authentication from the Access Tester Console" on page 10-19
3. "Testing User Authorization from the Access Tester Console" on page 10-21
4. Check latency information in the execution logfile as shown here, as well as in other files generated during a test run. For example:

```
...
[2/3/10 11:03 PM][info] Summary statistics
[2/3/10 11:03 PM][info] Matched 4 of 4, avg latency 232ms vs 238ms
[2/3/10 11:03 PM][info] Validate: matched 2 of 2, avg latency 570ms vs 578ms
[2/3/10 11:03 PM][info] Authenticate: matched 1 of 1, avg latency 187ms vs
187ms
[2/3/10 11:03 PM][info] Authorize: matched 1 of 1, avg latency 172ms vs 188ms
...
```

5. Proceed to:
   - Creating and Managing Test Cases and Scripts
   - Evaluating Scripts, Log File, and Statistics

10.6 Creating and Managing Test Cases and Scripts

Test management refers to the creation of repeatable tests that can be executed at any time by an individual administrator or system. Quick spot checks are very useful and effective in troubleshooting current issues. However, a more predictable and repeatable approach to validating server and policy configuration is often necessary. This approach can include testing OAM Server configuration for regressions after a product revision, or during a policy development and QA cycle.

To be useful such tests must allow for multiple use cases to be executed as group. Once the test scripts have been designed and validated as correct, replaying the tests against the OAM Server helps identify regressions in a policy configuration.

This section provides the information you need to perform test management in the following topics:

- About Test Cases and Test Scripts
- Generating an Input Test Script
- Personalizing an Input Test Script
- Executing a Test Script

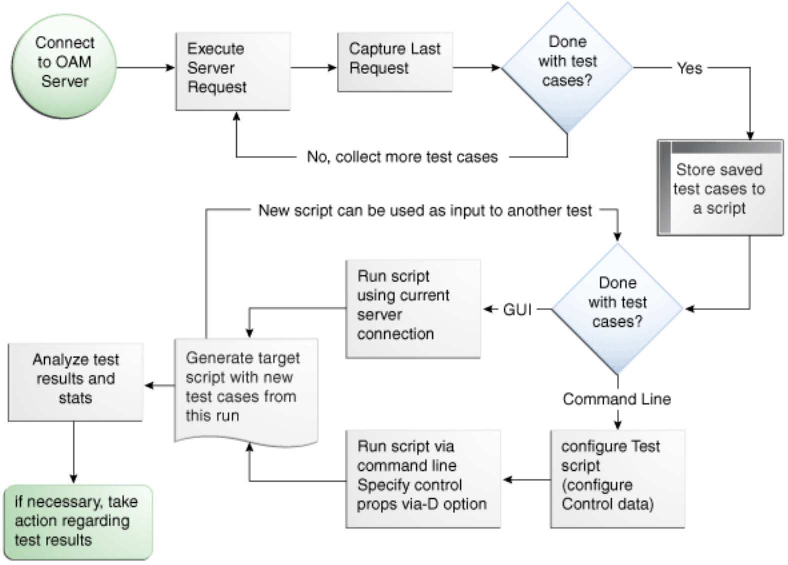
10.6.1 About Test Cases and Test Scripts

A test case is created from the request sent to, and response data received from, the OAM Server using the Access Tester. Among other data elements, a test case includes request latency and other identifying information that enables analysis and comparison of old and new test cases.

Once captured, the test case can be replayed without new input, and then new results can be compared with old results. If the old results are marked as "known good" then deviations from those results constitute failed test cases.

The test case workflow is illustrated by Figure 10–7.

Figure 10–7 Test Case Workflow


Connect to OAM Server
Execute Server Request
Capture Last Request
Done with test cases?
Yes
No, collect more test cases
Store saved test cases to a script
New script can be used as input to another test
Run script using current server connection
GUI
Done with test cases?
Command Line
Analyze test results and stats
Generate target script with new test cases from this run
Run script via command line Specify control props via-D option
configure Test script (configure Control data)
if necessary, take action regarding test results

Task overview: Creating and managing a test case

From the Access Tester Console, you can connect to the OAM Server and manually conduct individual tests. You can save the request to the capture queue after a request is sent and the response is received from the OAM Server. You can continue capturing additional test cases before generating a test script and clearing the capture queue. If you exit the Access Tester before saving the capture queue, you are asked if the test cases should be saved to a script before exiting. Oracle recommends that you do not clear the queue until all your test cases have been captured.

Once you have the test script, you can run it from either the Access Tester Console or from the command line.

10.6.2 Capturing Test Cases

You can save each test case to a capture queue after sending the request from the Access Tester to the OAM Server and receiving the response. You can capture as many individual test cases as you need before generating a test script that will automate running the group of test cases. For instance, the following outlines three test cases that must be captured individually:

- A validation request and response

- An authentication request and response
- An authorization request and response

Table 10–10 describes the location of the capture options.

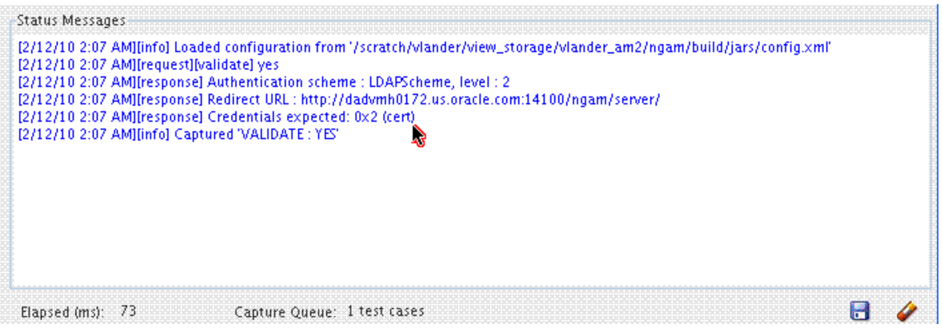
Table 10–10 Access Tester Capture Request Options

Location	Description
Test menu Capture last "..." request	Select this command from the Test menu to add the last request issued and results received to the capture queue (for inclusion in a test script later).
⇪	Select this command button from the tool bar to add the last request issued and results received to the capture queue (for inclusion in a test script later).

If you exit the Access Tester before saving the capture queue, you are asked if the test cases should be saved to a script before exiting. Do not clear the Access Tester capture queue until all your test cases have been captured.

To capture one or more test cases

1. Initiate a request from the Access Tester Console, as described in "Testing Connectivity and Policies from the Access Tester Console" on page 10-14.
2. After receiving the response, click the Capture last "..." request command button in the tool bar (or choose it from the Test menu).
3. Confirm the capture in the Status Messages panel and note the Capture Queue test case count at the bottom of the Console, as shown here.



```
[2/12/10 2:07 AM][info] Loaded configuration from '/scratch/vlander/view_storage/vlander_am2/ngam/build/jars/config.xml'
[2/12/10 2:07 AM][request][validate] yes
[2/12/10 2:07 AM][response] Authentication scheme : LDAPScheme, level : 2
[2/12/10 2:07 AM][response] Redirect URL : http://dadvmh0172.us.oracle.com:14100/ngam/server/
[2/12/10 2:07 AM][response] Credentials expected: 0x2 (cert)
[2/12/10 2:07 AM][info] Captured 'VALIDATE : YES'
```

Elapsed (ms): 73   Capture Queue: 1 test cases

4. Repeat steps 1, 2, and 3 to capture in the queue each test case that you need for your test script.
5. Proceed to "Generating an Input Test Script".

10.6.3 Generating an Input Test Script

A test script is a collection of individual test cases that were captured using the Access Tester Console. When individual test cases are grouped together, it becomes possible to automate test coverage to validate policy configuration for a specific application or site.

You can create a test script to be used as input to the Access Tester and drive automated processing of multiple test cases. The Generate Script option enables you to create an XML file test script and clear the capture queue. If you exit the Access Tester before saving the capture queue, you are asked if the test cases should be saved to a script before exiting.

> **Note:** Do not clear the capture queue until you have captured all the test cases you want to include in the script.

10.6.3.1 About Generating an Input Test Script

You can create a test script to be used as input to the Access Tester and drive automated processing of multiple test cases. Such a script must follow these rules:

- Allows possible replay by a person or system
- Allows possible replay against different policy servers w/o changing the script, to enable sharing of test scripts to drive different Policy Servers
- Allows comparison of test execution results against "Known Good" results

Following are the locations of the Generate Script command.

Table 10–11 Generate Script Command

Location of the Command	Description
Test menu Generate Script	Select Generate Script from the Test menu to initiate creation of the script containing your captured test cases.
📜	Select the Generate Script command button from the tool bar to initiate creation of the script containing your captured test cases. After you specify or select a name for your script, you are asked if the capture queue should be cleared. Do not clear the capture queue until all your test cases are saved to a script.

10.6.3.2 Generating an Input Test Script

Prerequisites

Capturing Test Cases

To record a test script containing captured test cases

1. Perform and capture each request that you want in the script, as described in "Capturing Test Cases" on page 10-23.
2. Click the Generate Script command button in the tool bar (or choose it from the Test menu to include all captured test cases.
3. In the new dialog box, select or enter the name of your new XML script file and then click Save.
4. Click Yes to overwrite an existing file (or No to dismiss the window and give the file a new name).
5. In the Save Waning dialog box, click No to retain the capture queue and continue adding test cases to your script (or click Yes to clear the queue of all test cases).
6. Confirm the location of the test script before you exit the Access Tester.
7. Personalize the test script to include details such as who, when, and why the script was developed, as described next.

10.6.4 Personalizing an Input Test Script

This section describes how to personalize and customize a test script.

- About Customizing a Test Script
- Customizing a Test Script

10.6.4.1 About Customizing a Test Script

The control block of a test script is used to tag the script and specify information to be used during the execution of a test. You might want to include details about who created the script and when and why the script was created. You might also want to customize the script using one or more control parameters.

The Access Tester provides command line "control" parameters to change processing of the script without changing the script. (test name, test number, and so on). This enables you to configure test runs without having to change "known good" input test scripts. Table 10–12 describes the control elements and how to customize these.

Table 10–12 Test Script Control Parameters

Control Parameter	Description
`ignorecontent=true`	Ignores differences in the Content section of the use case when comparing the original OAM Server response to the current response. The default is to compare the Content sections. This parameter can be overwritten by a command line property when running in the command line mode. Default: false (Compare Content sections). Values: true or false In command line mode, use Ignorecontent=true to over ride the specified value in the Control section of the input script.
testname="oamtest"	Specifies a prefix to add to file names in the "results bundle" as described in the previous section. In command line mode, use Testname=name to over ride the specified value in the Control section.
Configfile="config.xml"	Specifies the absolute path to a configuration XML file that was previously created by the Access Tester. In command line mode, this file is used by the Access Tester to locate connection details to establish a server connection.
Numthreads Reserved for future use	indicates the number of threads to be started by the Access Tester to run multiple copies of the test script. This supports stress testing of the OAM Server. Default: 1
Numiterations Reserved for future use	indicates the number of iterations of the test should be performed by the Access Tester. This provides for longevity testing of the OAM Server. Default: 1

10.6.4.2 Customizing a Test Script

Prerequisites

Generating an Input Test Script

To customize a test script

1. Locate and open the test script that was generated by the Access Tester.
2. Add any details that you need to customize or personalize the script.
3. Save the file and proceed to "Executing a Test Script".

10.6.5 Executing a Test Script

Once a test script has been created against a "Known Good" policy configuration and marked as "Known Good", it is important to drive the Access Tester using the script rather than specifying each test manually using the Console. This section provides the following topics:

- About Test Script Execution
- Running a Test Script

10.6.5.1 About Test Script Execution

You can interactively execute tests scripts from within the Access Tester Console, or use automated test runs performed by command scripts. Automated test runs can be scheduled by the operating system or a harness such as Apache JMeter, and executed without manual intervention. Other than lack of human input in command line mode, the two execution modes are identical.

> **Note:** A script such as .bat (Windows) or .sh (Unix) executes a test script in command line mode. Once a test script is created, it can be executed using either the Run Script menu command or the Access Tester command line.

Table 10–13 describes the commands to execute a test script.

Table 10–13 Run Test Script Commands

Location	Description
Test menu Run Script	Select the Run Script command from the Test menu to begin running a saved test script against the current policy server. The Status message panel is populated with the execution status as the script progresses.
	Select the Run Script command button from the tool bar to begin running a saved test script against the current policy server. The Status message panel is populated with the execution status as the script progresses.
Command line mode	A script such as .bat (Windows) or .sh (Unix) executes a test script in command line mode. Once a test script is created, it can be executed using either the Run Script menu command or the Access Tester command line.

The following overview describes how the Access Tester operates when running a test. Other than lack of human input in command line mode, the two execution modes are identical.

Process overview: Access Tester behavior when running a test script

1. The Access Tester loads the input xml file.

   In command line mode, the Access Tester opens the configuration XML file defined within the input test script's Control element.

2. The Access Tester connects to the primary and secondary OAM Proxy using information in the Server Connection panel of the Console.

   In command line mode, the Access Tester uses information in the Connection element of the configuration XML file.

3. In command line mode, the Access Tester checks the Control elements in the input script XML file to ensure none have been overwritten on the command line (command line values take precedence).

4. For each original test case defined in the script, the Access Tester:

   a. Creates a new target test case.

   b. Sends the original request to the OAM Server and collects the response.

   c. Makes the following comparisons:

      Compares the new response to the original response.

      Compares response codes and marks as "mismatched" any new target test case where response codes differ from the original test case. For instance, if the original Validate returned "Yes", and now returns "No", a mismatch is marked.

      When response codes are identical, and "the ignorecontent" control parameter is "false", the Access Tester compares Content (the name of the Authentication scheme or post authorization actions that are logged after each request). If Content sections differ, the new target test case is marked "mismatched".

   d. Collect new elapsed time and store it in the target use case.

   e. Build a new target test case containing the full state of the last server request and the same unique ID (UUID) as the original test case.

   f. Update the internal statistics table with statistics for the target test case (request type, elapsed time, mismatched, and so on).

5. After completing all the input test cases, the Access Tester:

   a. Displays summary results.

   b. Obtains and combines the *testname* and *testnumber*, and generates a name for the "results bundle" (three files whose names start with <*testname*>_<*testnumber*>.

      Note: Shell scripts can automate generating the bundle by providing testname and testnumber command line parameters.

      Obtain *testname* from the command line parameter. If not specified in the command line, use the *testname* element of the input script's Control block.

      Obtain *testnumber* from the command line parameter. If not specified, *testnumber* defaults to a 7-character numeric string based on the current local time: 2 character minutes, 2 character seconds, 3 character hundredths.

   c. Generates the "results bundle": three files whose names start with <*testname*>_<*testnumber*>:

      The target XML script contains the new test cases: <*testname*>_<*testnumber*_results.xml.

      The statistics XML file contains a summary and detailed statistics of the entire test run, plus those test cases marked as "mismatched": <*testname*>_<*testnumber*_stats.xml

      The execution log file contains information from the Status Message panel: <*testname*>_<*testnumber*_log.log.

d. In command line mode, the Access Tester exits with the exit code as described in "About the Access Tester Command Line Mode" on page 10-10.

10.6.5.2 Running a Test Script

Prerequisites

Generating an Input Test Script

To run a test script

1. Confirm the location of the saved test script before exiting the Access Tester., as described in "Generating an Input Test Script" on page 10-24.
2. Submit the test script for processing using one of the following methods:
   - From the Access Tester Console, click the Run Script command button in the tool bar (or select Run Script from the Test menu), then follow the prompts and observe messages in the Status Message panel as the script executes.
   - From the command line, specify your test script with the desired system properties, as described in "Starting the Access Tester with System Properties For Use in Command Line Mode" on page 10-10.

     ```
     java -Dscript.scriptfile="\tests\script.xml" -Dcontrol.ignorecontent="true"
     -jar oamtest.jar
     ```
3. Review the log and output files and perform additional analysis after the Access Tester compares newly generated results with results captured in the input script, as described in "Evaluating Scripts, Log File, and Statistics".

10.7 Evaluating Scripts, Log File, and Statistics

This section provides the following information:

- About Evaluating Test Results
- About the Saved Connection Configuration File
- About the Generated Input Test Script
- About the Target Output File Containing Test Run Results
- About the Statistics Document
- About the Execution Log

10.7.1 About Evaluating Test Results

At the end of a test run a "results bundle" gets generated containing three documents:

- Target script: An XML document containing new test cases
- Execution log: A text file containing the messages displayed during script execution
- Execution statistics: An XML document containing test metrics and a list of mismatched elements

The matching pair of test cases in the original and target scripts shares the test case ID. This ID is represented by a UUID value, which makes it possible to compare individual test cases in the original script with those in the target script. For more information, see "About the Generated Input Test Script" on page 10-31.

The statistics document contains the summary and detail statistics, as well as a list of test cases that did not match. The detailed statistics can be used for further analysis or to keep a historical trail of results. The summary statistics are the same statistics displayed at the end of the test run and can be used to quickly assess the state of a test run. The list of mismatched test cases as created in the statistics document contains test case IDs that have triggered mismatch and includes the reason for the mismatch, as seen in Table 10–14.

Table 10–14 Mismatched Results Reasons in the Statistics Document

Reason for a MisMatch	Description
Result	The test cases did not match because of the difference in OAM Server response codes (Yes versus No).
Content	The test cases did not match because of the differences in the specific data values that were returned by the OAM Server. The specific values from the last test run that have triggered the mismatch are included.

10.7.2 About the Saved Connection Configuration File

This is the output files that is saved using the Save Configuration command on the File menu; the default file name is config.xml. This connection configuration file includes details that were specified in the Access Tester Console, Server Connection panel.

> **Note:** An input test script file is also generated as described in the following topic. The name of the configuration file is used in the input test script to ensure that running the Access Tester in command line mode picks up connection information defined in the connection file.

Example 10–1 Connection Configuration File

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<oamtestconfig xmlns="http://xmlns.oracle.com/idm/oam/oamtest/schema"
version="1.0">
    <connection timeout="30000" minnconn="1" mode="open">
        <agent password="00030d05101b050c42" name="agent1"/>
        <keystore rootstore="" keystore_password="" keystore=""
global_passphrase=""/>
        <primary>
            <server maxconn="1" port="2100" addr="oam_server1"/>
        </primary>
        <secondary>
            <server maxconn="1" port="0" addr=""/>
        </secondary>
    </connection>
    <uri getauthscheme="true">
        <scheme>http</scheme>
        <host>oam_server1</host>
        <port>7777</port>
        <resource>/index.html</resource>
        <operation>Get</operation>
    </uri>
    <identity>
        <id>admin1</id>
        <password>00030d05101b050c42</password>
        <ipaddr>111.222.3.4</ipaddr>
    </identity>
</oamtestconfig>
```

10.7.3 About the Generated Input Test Script

The input test script is generated by using the Access Tester and capturing your own test cases. The "configfile" attribute of the "Control" element is updated after creation to specify the connection configuration file to be used in command line mode for establishing a connection to the OAM Server.

Example 10–2 Generated Input Test Script

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<oamtestscript xmlns="http://xmlns.oracle.com/idm/oam/oamtest/schema"
version="1.0">
    <history description="Manually generated using agent 'agent1'"
createdon="2010-02-03T22:28:00.468-05:00" createdby="test_user"/>
    <control numthreads="1" numiterations="1" ignorecontent="false"
testname="samplerun1" configfile="config.xml"/>
    <cases numcases="4">
        <case uuid="465a4fda-d814-4ab7-b81b-f3f1cd72bbc0">
            <request code="Validate">
                <uri getauthscheme="true">
                    <scheme>http</scheme>
                    <host>oam_server1</host>
                    <port>7777</port>
                    <resource>/index.html</resource>
                    <operation>Get</operation>
                </uri>
            </request>
            <response elapsed="984" code="Yes">
                <comment></comment>
                <status>Major code: 4(ResrcOpProtected) Minor code:
2(NoCode)</status>
                <content>
                    <line type="auth.scheme.id">LDAPScheme</line>
                    <line type="auth.scheme.level">2</line>
                    <line type="auth.scheme.required.creds">2</line>
                    <line
type="auth.scheme.redirect.url">http://dadvmh0172.us.oracle.com:14100/oam/server/<
/line>
                </content>
            </response>
        </case>
        <case uuid="009b44e3-1a94-4bfc-a0c3-84a38a9e0f2a">
            <request code="Authenticate">
                <uri getauthscheme="true">
                    <scheme>http</scheme>
                    <host>oam_server1</host>
                    <port>7777</port>
                    <resource>/index.html</resource>
                    <operation>Get</operation>
                </uri>
                <identity>
                    <id>weblogic</id>
                    <password>00030d05101b050c42</password>
                    <ipaddr>192.168.1.8</ipaddr>
                </identity>
            </request>
            <response elapsed="187" code="Yes">
                <comment></comment>
                <status>Major code: 10(CredentialsAccepted) Minor code:
```

```
2(NoCode)</status>
                <content>
                    <line type="user.dn">cn=weblogic,dc=us,dc=oracle,dc=com</line>
                </content>
            </response>
        </case>
        <case uuid="84fe9b06-86d1-47df-a399-6311990743c3">
            <request code="Authorize">
                <uri getauthscheme="true">
                    <scheme>http</scheme>
                    <host>oam_server1</host>
                    <port>7777</port>
                    <resource>/index.html</resource>
                    <operation>Get</operation>
                </uri>
                <identity>
                    <id>weblogic</id>
                    <password>00030d05101b050c42</password>
                    <ipaddr>192.168.1.8</ipaddr>
                </identity>
            </request>
            <response elapsed="188" code="Yes">
                <comment></comment>
                <status>Major code: 8(Allow) Minor code: 2(NoCode)</status>
                <content/>
            </response>
        </case>
        <case uuid="61579e47-5532-42c3-bbc7-a00828256bf4">
            <request code="Validate">
                <uri getauthscheme="false">
                    <scheme>http</scheme>
                    <host>oam_server1</host>
                    <port>7777</port>
                    <resource>/index.html</resource>
                    <operation>Get</operation>
                </uri>
            </request>
            <response elapsed="172" code="Yes">
                <comment></comment>
                <status>Major code: 4(ResrcOpProtected) Minor code:
2(NoCode)</status>
                <content/>
            </response>
        </case>
    </cases>
</oamtestscript>
```

10.7.4 About the Target Output File Containing Test Run Results

This example was generated by running the Access Tester in command line mode and specifying the script.xml file as input to execute the 4 captured test cases:

```
Dscript.scriptfile="script.xml" -jar oamtest.jar
```

Notice the various sections in Example 10–3. As shown in the execution log, this test run found no mismatches, and shows that 4 out of 4 requests matched.

Example 10–3 Output File Generated During a Test Run

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<oamtestscript xmlns="http://xmlns.oracle.com/idm/oam/oamtest/schema"
version="1.0">
    <history description="Generated from script 'script.xml' using agent 'agent1'"
createdon="2010-02-03T23:03:02.171-05:00" createdby="test_user"/>
    <control numthreads="1" numiterations="1" ignorecontent="false"
testname="oamtest" configfile=""/>
    <cases numcases="4">
        <case uuid="465a4fda-d814-4ab7-b81b-f3f1cd72bbc0">
            <request code="Validate">
                <uri getauthscheme="true">
                    <scheme>http</scheme>
                    <host>oam_server1</host>
                    <port>7777</port>
                    <resource>/index.html</resource>
                    <operation>Get</operation>
                </uri>
            </request>
            <response elapsed="969" code="Yes">
                <comment></comment>
                <status>Major code: 4(ResrcOpProtected) Minor code:
2(NoCode)</status>
                <content>
                    <line type="auth.scheme.id">LDAPScheme</line>
                    <line type="auth.scheme.level">2</line>
                    <line type="auth.scheme.required.creds">2</line>
                    <line
type="auth.scheme.redirect.url">http://dadvmh0172.us.oracle.com:14100/oam/server/
</line>
                </content>
            </response>
        </case>
        <case uuid="009b44e3-1a94-4bfc-a0c3-84a38a9e0f2a">
            <request code="Authenticate">
                <uri getauthscheme="true">
                    <scheme>http</scheme>
                    <host>oam_server1</host>
                    <port>7777</port>
                    <resource>/index.html</resource>
                    <operation>Get</operation>
                </uri>
                <identity>
                    <id>weblogic</id>
                    <password>00030d05101b050c42</password>
                    <ipaddr>111.222.3.4</ipaddr>
                </identity>
            </request>
            <response elapsed="187" code="Yes">
                <comment></comment>
                <status>Major code: 10(CredentialsAccepted) Minor code:
2(NoCode)</status>
                <content>
                    <line type="user.dn">cn=weblogic,dc=us,dc=oracle,dc=com</line>
                </content>
            </response>
        </case>
        <case uuid="84fe9b06-86d1-47df-a399-6311990743c3">
            <request code="Authorize">
                <uri getauthscheme="true">
```

```
                    <scheme>http</scheme>
                    <host>oam_server1</host>
                    <port>7777</port>
                    <resource>/index.html</resource>
                    <operation>Get</operation>
                </uri>
                <identity>
                    <id>weblogic</id>
                    <password>00030d05101b050c42</password>
                    <ipaddr>111.222.3.4</ipaddr>
                </identity>
            </request>
            <response elapsed="172" code="Yes">
                <comment></comment>
                <status>Major code: 8(Allow) Minor code: 2(NoCode)</status>
                <content/>
            </response>
        </case>
        <case uuid="61579e47-5532-42c3-bbc7-a00828256bf4">
            <request code="Validate">
                <uri getauthscheme="false">
                    <scheme>http</scheme>
                    <host>oam_server1</host>
                    <port>7777</port>
                    <resource>/index.html</resource>
                    <operation>Get</operation>
                </uri>
            </request>
            <response elapsed="171" code="Yes">
                <comment></comment>
                <status>Major code: 4(ResrcOpProtected) Minor code:
2(NoCode)</status>
                <content/>
            </response>
        </case>
    </cases>
</oamtestscript>
```

10.7.5 About the Statistics Document

The statistics file (_stats.xml) is generated together with the target output script during the test run identified in the Execution log. The script.xml file was used as input to execute the 4 captured test cases. The test run found no mismatches, and shows that 4 out of 4 requests matched.

A sample statistics document is shown in Example 10–4. The various sections that provide statistics for this run, which you can compare against statistics for an earlier "known good" run.

Example 10–4 Sample Statistics Document

```
A sample statistics document is shown here. Notice,
<oamteststats xmlns="http://xmlns.oracle.com/idm/oam/oamtest/schema"
version="1.0">
     <history description="Generated from script 'script.xml' using agent
       'agent1'" createdon="2010-02-03T23:03:02.171-05:00" createdby="test_user"/>
     <summary>
          <total>
              <nummatched>4</nummatched>
```

```
    <numtotal>4</numtotal>
    <avgelapsedsource>238</avgelapsedsource
    <avgelapsedtarget>232</avgelapsedtarget>
</total>
<validate>
    <nummatched>2</nummatched>
    <numtotal>2</numtotal>
    <avgelapsedsource>578</avgelapsedsource>
    <avgelapsedtarget>570</avgelapsedtarget>
</validate>
<authenticate>
    <nummatched>1</nummatched>
    <numtotal>1</numtotal>
    <avgelapsedsource>187</avgelapsedsource>
    <avgelapsedtarget>187</avgelapsedtarget>
</authenticate>
<authorize>
    <nummatched>1</nummatched>
    <numtotal>1</numtotal>
    <avgelapsedsource>188</avgelapsedsource>
    <avgelapsedtarget>172</avgelapsedtarget>
</authorize>
<summary>
<detail>
     <source>
          <validate>
             <yes>2</yes>
             <no>0</no>
             <error>0</error>
             <mismatch>0</mismatch>
             <elapsed>1156</elapsed>
          </validate>
      <authenticate>
             <yes>1</yes>
             <no>0</no>
             <error>0</error>
             <mismatch>0</mismatch>
             <elapsed>187</elapsed>
     </authenticate>
     <authorize>
             <yes>1</yes>
             <no>0</no>
             <error>0</error>
             <mismatch>0</mismatch>
             <elapsed>188</elapsed>
     </authorize>
</source>
<target>
     <validate>
             <yes>2</yes>
             <no>0</no>
             <error>0</error>
             <mismatch>0</mismatch>
             <elapsed>1140</elapsed>
     </validate>
<authenticate>
             <yes>1</yes>
             <no>0</no>
             <error>0</error>
             <mismatch>0</mismatch>
```

```
                       <elapsed>187</elapsed>
          </authenticate>
          <authorize>
                       <yes>1</yes>
                       <no>0</no>
                       <error>0</error>
                       <mismatch>0</mismatch>
                       <elapsed>172</elapsed>
          </authorize>
      <target>
      </detail>
    <mismatch numcases="0"/>
</oamteststats>
```

10.7.6 About the Execution Log

This sample execution log was generated together with the target output script during a test run using script.xml to execute 4 test cases. The test run found no mismatches, and shows that 4 out of 4 requests matched.

As you review this example, notice the information provided which is the same as the information you see in the Status Messages panel of the Access Tester. Notice the test cases, test name, connection configuration file, agent name, connection status, request validation status, authentication scheme, redirect URL, credentials expected, authentication status and user DN, session ID, authorization status, validation status, and summary statistics. Also notice that the target script and statistics document were generated by this run.

Example 10–5 Execution Log

```
[2/3/10 11:02 PM][info] Setting up to run script 'script.xml'
[2/3/10 11:02 PM][info] Loading test cases and control parameters from script
[2/3/10 11:02 PM][info] Loaded 4 cases
[2/3/10 11:02 PM][info] Control data for this test run:
[2/3/10 11:02 PM][info] Test name : 'samplerun1'
[2/3/10 11:02 PM][info] Configuration file : 'config.xml'
[2/3/10 11:02 PM][info] Ignore content : 'false'
[2/3/10 11:02 PM][info] Loading server configuration from file
[2/3/10 11:02 PM][info] Loaded server configuration
[2/3/10 11:02 PM][info] Connecting to server as agent 'oam_agent1'
[2/3/10 11:03 PM][info][request] Connect : Yes
...
[2/3/10 11:03 PM][info] Test 'samplerun1' will process 4 cases
[2/3/10 11:03 PM][info][request] Validate : Yes
[2/3/10 11:03 PM][info] Authentication scheme : LDAPScheme, level : 2
[2/3/10 11:03 PM][info] Redirect URL :
http://oam_server1.us.company.com:2100/server/
[2/3/10 11:03 PM][info] Credentials expected: 0x01 (password)
[2/3/10 11:03 PM][info][request] Authenticate : Yes
[2/3/10 11:03 PM][info] User DN : cn=admin1,dc=us,dc=company,dc=com
[2/3/10 11:03 PM][info] Session ID : -1
[2/3/10 11:03 PM][info][request] Authorize : Yes
[2/3/10 11:03 PM][info][request] Validate : Yes
[2/3/10 11:03 PM][info] Summary statistics
[2/3/10 11:03 PM][info] Matched 4 of 4, avg latency 232ms vs 238ms
[2/3/10 11:03 PM][info] Validate: matched 2 of 2, avg latency 570ms vs 578ms
[2/3/10 11:03 PM][info] Authenticate: matched 1 of 1, avg latency 187ms vs 187ms
[2/3/10 11:03 PM][info] Authorize: matched 1 of 1, avg latency 172ms vs 188ms
[2/3/10 11:03 PM][info] Generated target script 'samplerun1_0302171__target.xml'
```

```
[2/3/10 11:03 PM][info] Generated statistics log 'samplerun1_0302171__stats.xml'
```

11

Configuring Centralized Logout for OAM 11g

Different agents require different logout implementation steps. Oracle recommends that logout for Oracle Access Manager 11g be handled in the manner described in this chapter.

This chapter includes the following sections:

- Prerequisites
- Introduction to OAM 11g Centralized Logout
- Configuring Centralized Logout for 11g WebGate with OAM 11g Server
- Configuring Centralized Logout for the IDM Domain Agent
- Configuring Centralized Logout for 10g WebGate with OAM 11g Servers
- Configuring Centralized Logout for Oracle ADF-Coded Applications
- Validating Global Sign-On and Centralized Logout

> **Caution:** Oracle recommends using the logout mechanism provided by Oracle Access Manager, not custom logout scripts.

11.1 Prerequisites

Before you can perform tasks in this chapter:

- The partner application must be deployed on the Web server where the agent is configured and registered with OAM 11g
- One of the following agents, on any supported Web server and platform, must be running and provisioned with OAM 11g as follows:
  - OAM 11g WebGate with OAM 11g Server
  - IDM Domain Agent with OAM 11g Server
  - OAM 10g WebGate with OAM 11g Server
  - OAM 10g WebGate with OAM 10g Server
  - OSSO Agent (mod_osso)
- Policies must be configured to protect the resource in an OAM 11g application domain

11.2 Introduction to OAM 11g Centralized Logout

Oracle Access Manager 11g provides centralized logout (also known as global log out) for user sessions. With OAM, centralized logout refers to the process of terminating an active user session.

Centralized logout means:

- Applications must not provide their own logout page for use in an SSO environment.
- Applications must make their logout links configurable with a value that points to the logout URL specified by the OAM WebGate Administrator.

> **Note:** Oracle strongly recommends that applications use the ADF Authentication servlet, which interfaces with OPSS where a domain-wide configuration parameter can be used to specify the logout URL. This way applications need not be modified or redeployed to change logout configuration.

Table 11–1 describes the circumstances under which centralized logout occurs.

Table 11–1 Centralized Logout Circumstances

Explicitly	The client state is invalidated and the session ends. If a new attempt is made to access the resource, the client must re-authenticate. When the user logs out. When the administrator terminates the session When the session is terminated based on changes on the identity side
Implicitly	When no user activity occurs within the defined session timeout period, the user is logged out automatically and redirected back to the partner with a new session ID and a new prompt for credentials. This occurs if no lower-level authentication is configured for the resource. With OAM 11g, the user is not logged out if 10g WebGate simply encounters a logout URL unless the logout.html provides an explicit redirection to the Server logout. The OAM 11g WebGate redirects the user to the Server logout.

When the logout URL is encountered and the cookie is removed (ObSSOcookie for 10g WebGates; OAMAuthnCookie for 11g WebGates). WebGate logs out the user and requires re-authentication.

> **Note:** Unlike partner applications, external applications (Yahoo! Mail, for example), do not delegate authentication to OAM and do not cede logout control to the OAM single sign-on server. It is the user's responsibility to log out of each of these applications.

This section provides the following topics:

- About Centralized Logout with OAM 11g Agents and Servers
- About Centralized Logout with OAM 10g Agents and OAM 11g Servers
- About Centralized Logout with the IDM Domain Agent
- About Centralized Logout with OSSO Agents (mod_OSSO) and OAM 11g

- About Centralized Logout for Applications Using Oracle ADF Security

11.2.1 About Centralized Logout with OAM 11g Agents and Servers

This section describes the sign-out processing that occurs with OAM 11g WebGates protecting applications.

Generally speaking, during centralized logout with OAM 11g Server the SSO Engine receives a user-session-exists request. The Session Management Engine looks up the user session and responds that the user session exists. The SSO engine sends a Clear User Session request. The Session management engine clears the token and session context. The SSO engine sends a User Session Cleared response.

Clearing the user token and the session context clears the server-side state, which includes clearing the OAM_ID cookie set on the server side. When the agent is notified, the agent clears the client-side state of the partner application. For more information, see "Configuring Centralized Logout for 11g WebGate with OAM 11g Server".

11.2.2 About Centralized Logout with OAM 10g Agents and OAM 11g Servers

The following process overview outlines typical SSO Engine and Session Management Engine processing during centralized logout.

Logout is initiated when an application causes the invocation of the logout.html file configured for any registered OAM 10g WebGate.

Generally speaking, during centralized logout with OAM 10g WebGates the SSO Engine receives a user-session-exists request. The Session Management Engine looks up the user session and responds that the user session exists. The SSO engine sends a Clear User Session request. The Session management engine clears the token and session context. The SSO engine sends a User Session Cleared response.

Clearing the user token and the session context clears the server-side state, which includes clearing the OAM_ID cookie set on the server side. When the agent is notified, the agent clears the client-side state of the partner application. For more information, see "Configuring Centralized Logout for 10g WebGate with OAM 11g Servers".

11.2.3 About Centralized Logout with the IDM Domain Agent

The IDM Domain Agent is a domain-wide agent that provides single sign-on functionality for the IDM Administration Console. The IDM Domain Agent is installed and pre-configured as part of the Oracle Access Manager 11g Server installation and configuration.

For more information, see "Configuring Centralized Logout for the IDM Domain Agent" on page 11-6.

11.2.4 About Centralized Logout with OSSO Agents (mod_OSSO) and OAM 11g

With OSSO Agents (mod_osso 10g), partner applications also cede logout control to the OAM single sign-on server. When the user logs out of one partner application, she is automatically logged out of all other partner applications.

Note: No change is needed in the logout URL configuration of existing applications that use the OSSO Agent.

Process overview: Centralized logout with mod_osso

1. Clicking Logout in a partner application takes the user to the page where logout occurs
2. When a user has signed off successfully, each of the applications listed on the centralized logout page has a check mark beside the application name.
3. A broken image beside an application name identifies an unsuccessful logout.
4. Once all of the application names activated in a session have a check mark, you can click Return to go to the application from which you initiated logout.

11.2.5 About Centralized Logout for Applications Using Oracle ADF Security

Oracle Application Development Framework (Oracle ADF) security and the Oracle Platform Security Services (OPSS) comprise Oracle WebLogic Server's security framework. On the Oracle WebLogic Server, you can run a Web application that uses Oracle ADF security, integrates with Oracle Access Manager 11g SSO, and uses OPSS SSO for user authentication.

In this situation, users can terminate a single sign-on session and log out of all active partner applications simultaneously by logging out of whatever application they are working in.

For more information, see "Configuring Centralized Logout for ADF-Coded Applications with OAM 11g" on page 11-11.

11.3 Configuring Centralized Logout for 11g WebGate with OAM 11g Server

This section provides the following topics:

- About Configuring Centralized Logout for 11g WebGates
- Configuring Centralized Logout for 11g WebGates

11.3.1 About Configuring Centralized Logout for 11g WebGates

Several elements in the OAM 11g WebGate registration page enable centralized logout for OAM 11g WebGates. After registration, the ObAccessClient.xml file is populated with the information in Table 11–2.

Table 11–2 Logout Elements in OAM 11g WebGate Registration

Element	Description
Logout URL	The Logout URL triggers the logout handler, which removes the cookie (ObSSOCookie for 10g WebGates; OAMAuthnCookie for 11g WebGates) and requires the user to re-authenticate the next time he accesses a resource protected by Oracle Access Manager. ■ If there is a match, the WebGate logout handler is triggered. ■ If Logout URL is not configured the request URL is checked for "logout." and, if found (except "logout.gif" and "logout.jpg"), also triggers the logout handler Default = [] (not set) Note: This is the standard OAM 10g WebGate configuration parameter used to trigger initial logout.
Additional Logout for 11g WebGates Only	For OAM 11g WebGate single sign-off behavior, the following elements and values automate the redirect to a central logout URL, callback URL, and end URL. This replaces 10g WebGate single sign-off only through customized local logout page.

Table 11–2 (Cont.) Logout Elements in OAM 11g WebGate Registration

Element	Description
Logout Callback URL	The URL to oam_logout_success, which clears cookies during the call back. This can be a URI format without host:port (recommended), where the OAM Server calls back on the host:port of the original resource request. For example: Default = /oam_logout_success This can also be a full URL format with a host:port, where OAM 11g server calls back directly without reconstructing callback URL When the request URL matches the Logout Callback URL, WebGate clear its cookies and streams an image gif in the response. This is similar to OSSO agent behavior. When WebGate redirects to the server logout page, it records an "end" URL as a query parameter (end_url=http://host:port/..."), which becomes the landing page that the OAM 11g Server redirects back to after logout. Other OAM 11g services support the central logout page on the server. The end_url relies on the target URL query parameter passed from OPSS integrated applications.
Logout Redirect URL	This parameter is automatically populated after agent registration completes.By default, this is based on the OAM Server host name with a default port of 14200. For example: Default = http://*OAMServer_host*:14200/oam/server/logout The Logout URL triggers the logout handler, which removes the OAMAuthnCookie_*<host:port>*_*<random number>* and requires the user to re-authenticate the next time he accesses a resource protected by Oracle Access Manager. ■ When WebGate logout handler is triggered, it redirects to the central logout page specified by the Logout Redirect URL parameter if it is configured. ■ It is unlikely that the Logout Redirect URL is not configured because this is populated after agent registration., 10g behavior is triggered: serve the local logout page instead of redirecting to another. The local logout page can have a customized script to redirect to the central logout page and can clear additional 3rd party cookies if desired.
Logout Target URL	The value for this is name for the query parameter that the OPSS applications passes to WebGate during logout. This query parameter specifies the target URL of the landing page after logout. Default: end_url Note: The end_url value is configured using param.logout.targeturl in jps-config.xml. ■ If Logout Target URL is configured, WebGate searches for the value passed in the logout request's query parameter and passes it as end_url query parameter in the redirect URL to OAM Server. ■ If Logout Target URL is not configured, WebGate searches for the default name "end_url" and passes that end_url query parameter along.

Configuring 11g WebGates for logout against OAM 11g Servers requires a logoutCallbackUrl. Centralized logout for 11g agents sets the cookie from "loggedout" to empty and expires the OAMAuthnCookie_*<host:port>*_*<random number>* cookie to explicitly clear it during logout, (rather than leaving behind an empty or loggedout cookie).

OAM 11g WebGates differ only slightly from 10g WebGates, and match only the URI part of "logoutCallbackUrl".

The SSO Engine supports the central logout page on the OAM Server and:

- Calls back to "logoutCallbackUrl" of 11g WebGates during logout
- Lands on "end_url" (passed in as query parameter) after logout

The WebGate parameter "logoutCallbackUrl" can be configured (as /oam_logout_success, for example). Oracle recommends using a URI format without host:port, in which case, the OAM Server dynamically constructs the full URL based on the host:port in original request and calls back on it.

This can also be a full URL format with a host:port, where OAM 11g server calls back directly without reconstructing callback URL.

The OAM Server sets the cookie from "loggedout" to empty and expires the cookie to explicitly clear it during logout, rather than leaving behind an empty or loggedout cookie.

For details, see "Configuring Centralized Logout for 11g WebGates".

11.3.2 Configuring Centralized Logout for 11g WebGates

During OAM 11g WebGate registration, use the following procedure to configure logout with OAM 11g.

To configure centralized logout for 11g WebGates

1. Choose your method for registration:
   - Chapter 5, "Registering Partners (Agents and Applications) by Using the Console"
   - Chapter 6, "Registering Partners (Agents and Applications) Remotely"
2. When creating or editing an agent registration, include appropriate logout values for your environment (Table 11–2):
   - Logout URL
   - Logout Callback URL
   - Logout Redirect URL
   - Logout Target URL
3. Finish your agent registration, as usual.
4. Perform steps in "Validating Global Sign-On and Centralized Logout" on page 11-15.

11.4 Configuring Centralized Logout for the IDM Domain Agent

The IDM Domain Agent is pre-configured with the logout parameters needed to perform central logout against the OAM 11g Server. While similar to a 10g WebGate, the IDM Domain Agent does not have a local logout.html page to be configured. Instead, the IDM Domain Agent is delivered with a pre-deployed application oamsso_logout), that is used by the agent to perform the logout.

The logout functionality for the IDMDomainAgent requires that the oamsso_logout application is deployed in the Server where the IDMDomainAgent is used. The initial installation adds this application to AdminServer and to OAM Servers. However, you must update this application's Target servers to include all those that are using the IDMDomainAgent.

To configure logout for the IDM Domain Agent

1. Log in to the WebLogic Server Administration Console.
2. Navigate to Domain, Deployments, oamsso_logout, Targets.
3. Select all the Servers where the IDMDomainAgent is enabled and where logout is performed. For example, oim_server, oaam_admin, oaam_server, and so on.

4. Click Save.

11.5 Configuring Centralized Logout for 10g WebGate with OAM 11g Servers

This section provides the following topics:

- About Centralized Logout Processing for 10g WebGate with OAM 11g Server
- About the Centralized Logout Script for OAM 10g Agents with OAM 11g Servers
- Configuring Centralized Logout for 10g WebGates with OAM 11g

11.5.1 About Centralized Logout Processing for 10g WebGate with OAM 11g Server

The following process overview outlines the OAM 11g centralized logout process that occurs when the application is deployed on the Web server for which the protecting OAM 10g WebGate is configured.

Logout is initiated when an application causes the invocation of the logout.html file configured for the OAM agent (in this case, a 10g WebGate).

Process overview: Centralized logout for WebGate 10g with OAM 11g Server

1. The application causes invocation of the logout.html file configured for the OAM 10g WebGate.

   The application might also pass `end_url` as a query string to logout.html. The end_url parameter could either be a URI or a URL. For example:

   ```
   /oamsso/logout.html?end_url=/welcome.html
   or
   /oamsso/logout.html?end_url=http://my.site.com/welcome.html
   ```

2. WebGate clears the ObSSOCookie for its domain and loads the logout.html script.

3. If the `end_url` parameter does not include *host:port*, the logout.html script gets the *host:port* of the local server and constructs the `end_url` parameter as a URL. For example:

   ```
   http://serverhost:port/oam/server/logout?end_url=http://my.site.com/
   welcome.html
   ```

4. Logic in logout.html redirect to the OAM Server. For example:

   ```
   http://myoamserverhost:port/oam/server/logout?end_url=http://my.site.com/
   welcome.html
   ```

5. The OAM Server executes logout as follows:

   a. Cleans up the session information associated with the user at the server side.

   b. Validates the `end_url` and sends a page with callback URLs to the user's browser.

   > **Note:** The Logout Callback URL is specified in the expanded OAM Agent registration page, as described in "About the Create OAM Agent Page" on page 5-9.

c. From the callback page, a new request is initiated to a specific URI on each WebGate. When this request reaches the specific WebGate in the specific domain, the ObSSOCookie for that domain is cleared.

d. The user is redirected to the `end_url` in the logout script. However, if the `end_url` parameter is not present, an appropriate message is sent by the OAM Server.

For more information, see "About the Centralized Logout Script for OAM 10g Agents with OAM 11g Servers".

11.5.2 About the Centralized Logout Script for OAM 10g Agents with OAM 11g Servers

With an OAM 10g WebGate, the logout.html script is required for both single- and multiple DNS-domain centralized logout processing. The logout.html activates JavaScripts that perform the actual logout.

> **Note:** OAM 11g WebGates do not use the logout.html script and instead require additional details in their Agent registration configuration, as described in "Configuring Centralized Logout for 11g WebGate with OAM 11g Server" on page 11-4.

Example 11–1 is a logout.html script that you can use as a template by editing certain lines for your own environment, which are described at the top of the script. For instance, `SERVER_LOGOUTURL` must be changed. Additional information is provided after the example.

Example 11–1 logout.html Script

```
<html>
<head>
<script language="javascript" type="text/javascript">
////////////////////////////////////////////////////////////////////////////////
//Before using, you need to change the values of:
//a. "oamserverhost" to point to the host where the OAM 11g Server is running.
//b. "port" to point to the port where the OAM 11g Server is running.
////////////////////////////////////////////////////////////////////////////////
var SERVER_LOGOUTURL = "http://oamserverhost:port/oam/server/logout";
////////////////////////////////////////////////////////////////////////////////

function handleLogout() {

    //get protocol used at the server (http/https)
    var webServerProtocol = window.location.protocol;
    //get server host:port
    var webServerHostPort = window.location.host;
    //get query string present in this URL
    var origQueryString = window.location.search.substring(1);
    var newQueryString = "";

    //vars to parse the querystring
    var params = new Array();
    var par = new Array();
    var val;

    if (origQueryString != null && origQueryString != "") {
        params = origQueryString.split("&");
        for (var i=0; i<params.length; i++) {
            if (i == 0)
              newQueryString = "?";

          if (i > 0)
              newQueryString = newQueryString + "&";

          par = params[i].split("=");

          //prepare a new query string, if the end_url value needs to be changed
          newQueryString = newQueryString + (par[0]);
          newQueryString = newQueryString + "=";
          val = par[1];

          if ("end_url" == par[0]) {
          //check if val (value of end_url) begins with "/" or "%2F" (is it an URI?)
          if (val.substring(0,1) == "/" || val.substring(0,1) == "%") {
                  //modify the query string now
                  val = webServerProtocol + "//" + webServerHostPort + val;
              }
          }
          newQueryString = newQueryString + val;
      }

      //redirect the user to this URL
      window.location.href = SERVER_LOGOUTURL + newQueryString;
}

</script>
</head>

<body onLoad="handleLogout();">

</body>
</html>
```

Process overview: Logic in logout.html

1. Gets the host and port from the incoming request.
2. Gets the `end_url` parameter from the query string.

   If the `end_url` parameter is not a URL, then the logout.html script constructs a URL using the host and port from task 1. See "Guidelines for the end_url parameter in logout.html".
3. Redirects to the OAM Server logout URL (SERVER_LOGOUTURL). For example: http://*myoamserver/host:port*/oam/server/logout.
   - Use the `end_url` constructed in process 2 as the query string.
   - Preserve all other query string parameters in the query string

Guidelines for the end_url parameter in logout.html

The `end_url` parameter can be either a URI or an URL.

- If the `end_url` query string is a URI, without host and port, then the logout.html must construct the URL by determining the host and port of the Web Server where logout.html is hosted. For example:

  ```
  http://myoamserverhost:port/oam/server/logout?end_url=http://my
  .site.com/welcome.html
  ```

- If the end_url parameter is a URL with the host and port, the logout.html script simply passes that on without reconstructing it.

> **Note:** An ADF application must pass the end_url parameter indicating where to redirect the user after logout, as described in "Configuring Centralized Logout for Oracle ADF-Coded Applications" on page 11-11:
>
> ```
> /<app context root>/adfAuthentication?logout=true&end_url=<any uri>
> ```

Table 11–3 illustrates how a logout link in the logout.html file might be specified:

Table 11–3 Sample end_url Parameter Specifications

As a ...	Sample end_url Value
URI	/oamsso/logout.html?end_url=<*someUri*> For example: /oamsso/logout.html?end_url=/*welcome.html*
URL	/oamsso/logout.html?end_url=<*someUrl*> For example: */oamsso/logout.html?end_url=http://my.site.com/welc*ome.html

11.5.3 Configuring Centralized Logout for 10g WebGates with OAM 11g

The following procedures describe how to configure centralized logout for 10g WebGates with OAM 11g.

> **Note:** Optional tasks or those required for only multiple DNS domain logout are identified and can be skipped unless needed.

Chapter 17, "Managing OAM 10g WebGates with OAM 11g" includes a sample procedure that includes steps for deploying an application in a WebLogic Server domain.

Task overview: Configuring centralized logout for 10g WebGates

1. Create a default logout page (logout.html) and make it available on the WebGate installation directory:

   a. Create and edit logout.html for the WebGate based on Example 11–1, "logout.html Script".

   b. Store your logout.html script in the following directory path:

   ```
   WebGate_install_dir/oamsso/logout.html
   ```

> **Note:** If the logout.html file is located elsewhere, ensure that the logout link is correctly specified in the agent registration to point to the correct location of the logout.html file.

   c. Proceed with following steps, as needed.

2. Confirm that the logOutUrls parameter is configured for each resource WebGate, as follows:

   > **Note:** If the LogOutUrl parameter has already been configured for the 10g WebGate with a value other than "/oamsso/logout.html", then ensure that "/oamsso/logout.html" is also present as part of the LogOutUrls parameter.

   a. Confirm that the <callBackUri> is the second value as part of 'logOutUrls'.

   b. Confirm that the two values are separated by commas: "/oamsso/logout.html, <CallbackUri>".

3. Ensure that the logout.html (from Step 1) redirects the user to this central logout URI, "/oam/server/logout' on the OAM 11g Server.

4. **Optional**: Allow the application to pass the end_url parameter indicating where to redirect the user after logout, as described in "Guidelines for the end_url parameter in logout.html" on page 11-9.

5. **Multiple DNS Domains**: Perform the following steps if you have multiple DNS domains configured for SSO.

   > **Note:** The Logout Callback URL can be unique for each WebGate; however, to construct the Callback URL for each WebGate, it is sufficient for the OAM Server to know the host and port of each WebGate from each domain. The file that the Logout Callback URL points to must differ from the logout.html script in the WebGate installation directory.

   a. Configure the <CallbackUri> as the second value in the logOutUrls parameter on each resource WebGate.

      <CallbackUri> is the location on WebGate where the request must be sent to for clearing the obssocookie in that domain. The <CallbackUri> cannot be logout.html.

   b. Ensure that a file physically exists on each Web server at the <CallBackUri> location (usually, at the same location as logout.html).

      For example, if you configure a file named logout.png in the same location as logout.html, then a <CallBackUri> of logout.png should have the value:

      ```
      /oamsso/logout.png
      ```

6. Check the OHS Web server configuration file, httpd.conf, on which the 10g WebGate is configured and if the following lines exist delete them.

   ```
   <LocationMatch "/oamsso/*">
   Satisfy any
   </LocationMatch>
   ```

11.6 Configuring Centralized Logout for Oracle ADF-Coded Applications

The Oracle Access Manager SSO solution is available for applications that are coded to Oracle ADF standards and the OPSS SSO Framework. ADF-coded applications that are configured to perform logout with OAM 11g, redirect to the /oamsso/logout.html resource. The IDM Domain Agent intercepts and processes the request, cleans up the session, redirects to the central logout (done by the OAM Server) and redirects back to the end_url.

See Also: Oracle Fusion Middleware Application Security Guide

---

Note: For ADF applications, only one extra configuration step is needed (to configure the OAMSSOProvider for OPSS).

Task overview: Protecting ADF-coded applications with OAM 11g

1. Protect the ADF-coded application using either an:
   - 11g WebGate
   - 10g WebGate
2. Perform the single extra configuration step for ADF-coded applications: configure the OAMSSOProvider as described in "Configuring Centralized Logout for ADF-Coded Applications with OAM 11g" on page 11-13.
3. Perform logout configuration steps for your chosen WebGate version.

This section includes the following topics, which you can skip if you do not have applications that are coded to Oracle ADF standards and the OPSS SSO Framework:

- About Centralized Logout Processing for Applications Coded to Oracle ADF Standards
- Configuring Centralized Logout for ADF-Coded Applications with OAM 11g

11.6.1 About Centralized Logout Processing for Applications Coded to Oracle ADF Standards

ADF-coded applications refer to either applications that have been fully integrated with ADF or those that simply use ADF Authentication Servlet to integrate with OPSS.

In this case, logout is initiated when an ADF application causes the invocation of the logout URI. The following process overview outlines the OAM 11g centralized logout process for applications coded to Oracle ADF standards.

Process overview: Centralized logout for ADF applications with 10g WebGate

1. An ADF application causes the invocation of the following URI.

   ```
   /<app context root>/adfAuthentication?logout=true&end_url=<any uri>
   ```

   The `end_url` parameter specifies the URI to which the application returns control following logout.
2. ADF invokes the configured OPSS SSO provider (OAM in this case) and delegates the logout functionality to the configured logout URI by redirecting the request to the logout URI. The `end_url` value is passed as a query string to the logout URI. For example: `/oamsso/logout.html?end_url=<end_uri>`.
3. The logout URI is invoked on the WebGate front-ending the application.
4. 10g WebGate clears the ObSSOCookie for its domain and loads the logout.html script.

5. If the end_url parameter does not include *host:port*, the logout.html script gets the *host:port* of the local server and constructs the end_url parameter as a URL. For example:

```
http://serverhost:port/oam/server/logout?end_url=http://my.site.com/
welcome.html
```

6. Logic in logout.html redirect to the OAM Server. For example:

```
http://myoamserverhost:port/oam/server/logout?end_url=http://my.site.com/
welcome.html
```

7. The OAM Server executes logout as follows:

   a. Cleans up the session information associated with the user at the server side.

   b. Validates the end_url and sends a page with callback URLs to the user's browser.

   > **Note:** The Logout Callback URL is specified in the expanded (not short) OAM Agent registration, as described in Table 11–2.

   c. From the callback page, a new request is initiated to a specific URI on each WebGate. When this request reaches the specific WebGate in the specific domain, the ObSSOCookie for that domain is cleared.

   d. The user is redirected to the end_url in the logout script. However, if the end_url parameter is not present, an appropriate message is sent by the OAM Server.

11.6.2 Configuring Centralized Logout for ADF-Coded Applications with OAM 11g

The following procedure is similar to configuring logout for 10g WebGates, with specific step for ADF-coded applications. The ADF-coded application must send the end_url value to identify where to redirect the user after logout processing. However, with ADF-coded applications, logout occurs when the application causes the following URI to be invoked:

```
/<app context root>/adfAuthentication?logout=true&end_url=<any uri>
```

> **Note:** The Applcore f/w could facilitate triggering of the above URL and the ADF application could leverage that.

Some steps in this procedure require the WebLogic Scripting Tool (WLST): wlst.sh (Linux) or wlst.cmd (Windows), which you must invoke from the WLST_install_dir.

See Also:

- Appendix C, "Integrating Oracle ADF Applications with Oracle Access Manager 11g SSO"
- "Using Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

To configure centralized logout for ADF-coded applications

1. Check with the OAM Administrator to confirm the location of the logout.html script configured with the agent, which you need in following steps.

2. Configure OPSS for OAM as the SSO provider to update jps-config.xml for the WebLogic administration domain, as follows:

   a. On the computer hosting the Oracle WebLogic Server and the Web application using Oracle ADF security, locate the Oracle JRF WLST script. For example:

   ```
   cd $ORACLE_HOME/oracle_common/common/bin
   ```

   b. Connect to the computer hosting the Oracle WebLogic Server, enter the administrator ID and password, and the host and port of the WebLogic AdminServer:

   ```
   wls:/> /connect('admin_ID', 'admin_pw', 'hostname:port'
   ```

   For example, the Oracle WebLogic Administration Server host could be `localhost` using port `7001`. However, your environment might be different.

   c. Check with the OAM Administrator to confirm the location of the logout.html script configured with the agent.

   In Step d, you must use the value provided by the OAM administrator. Here, logouturival is the URI of the logout script /logout.html. The logouturl could either begin with "logout." (exceptions are logout.gif and logout.jpg) or it could be any other value configured by the OAM Administrator.

   d. Enter the loginuri for ADF authentication and the logouturi (location of the logout.html script configured with the agent); the host and port are not needed.

   ```
   wls:/>addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
   logouturi="/oamsso/logout.html", autologinuri="/obrar.cgi")
   ```

   Here, loginuri=/${app.context}/adfAuthentication; logouturival is the URI of the logout script /logout.html. The logouturl could either begin with "logout" (exceptions are logout.gif and logout.jpg) or it could be any other value configured by the OAM Administrator.

3. **Required**: The ADF application must pass the end_url parameter indicating where to redirect the user after logout, as follows:

   If the `end_url` parameter does not include *host:port*, the logout.html script gets the *host:port* of the local server and constructs the `end_url` parameter as a URL. For example:

   ```
   http://serverhost:port/oam/server/logout?end_url=http://serverhost:port/
   welcome.html
   ```

4. **OAM 11g WebGate**: Perform steps in "Configuring Centralized Logout for 11g WebGates" on page 11-6.

5. **OAM 10g WebGate**: Perform steps in "Configuring Centralized Logout for 10g WebGates with OAM 11g" on page 11-10.

> **See Also:** *Oracle Fusion Middleware Application Security Guide* for details about setting up providers for Oracle Access Manager 11g Identity Assertion.

11.7 Validating Global Sign-On and Centralized Logout

This section provides the following topics:

- Confirming Global Sign-On
- Validating Global Sign-On with Mixed Agent Types
- Observing Centralized Logout

11.7.1 Confirming Global Sign-On

Use the following procedure to observe single sign-on global login.

Prerequisites

- Agents and Servers must be registered with OAM 11g and running
- Resources and policies controlling SSO must be defined within OAM 11g application domains

To observe global sign-on

1. From a browser, enter the URL to a protected resource and sign in using proper credentials.
2. Enter the URL to another protected resource and confirm that you are not asked to re-authenticate.

11.7.2 Validating Global Sign-On with Mixed Agent Types

Use the following procedure to observe single sign-on global login with different applications and agents that have the same authentication level.

For example, suppose you have:

- OSSO Partner at `http://`*`host1.example.com:7777/private/index.html`* protected using mod_osso
- WebGate Partner at `http://`*`host2.example.com:8888/mydomain/finance/index.html`* protected using OAM Agent

Within the same browser session, you can access all applications protected by either agent with only a single sign in.

Prerequisites

- Agents and Servers must be registered with OAM 11g and running
- Resources and policies must be defined within OAM 11g application domains
- Both partners must be protected at the same authentication level
- Single sign-on must be configured as described in this chapter

To observe global sign-on with mixed agent types

1. **OSSO Agent Protected Application**:
   a. From a browser, enter the URL of the OSSO-protected resource
   b. Confirm that the login page appears and sign in using proper credentials.
   c. Confirm that the protected resource is served.

   d. Remain in the same browser session and proceed to Step 2.

2. **Same Browser Session, OAM Agent Protected Application**:

   a. In the same browser session as Step 1, enter the URL of the OAM Agent-protected resource.

   b. Confirm that the protected resource is served and that no login page appears.

3. Log out of the browser session.

4. **Fresh Browser Session, OAM Agent Protected Application**:

   a. In a fresh browser session, enter the URL of the OAM-protected resource.

   b. Confirm that the login page appears and sign in using proper credentials.

   c. Confirm that the protected resource is served.

   d. Remain in the same browser session and proceed to Step 5.

5. Same Browser Session, **OSSO Agent Protected Application**:

   a. In the same browser session as Step 4, enter the URL of the OSSO Agent-protected resource.

   b. Confirm that the protected resource is served and that no login page appears.

11.7.3 Observing Centralized Logout

Use the following procedure to observe centralized logout:

- With OAM Agents, the logout URL redirects to the server and cookies are cleared and invalidated so that a subsequent request cannot locate the cookie.
- With mod_osso, each agent destroys its own cookies. The logout URL redirects to the global logout page on the server and each partner sends cookies to the server.

Prerequisites

- Agents must be registered and running
- Resources must be protected by OAM 11g application domains
- Single sign-on must be configured with authentication and authorization policies and responses in OAM 11g application domains

To observe centralized logout

1. **Single Application**:

   a. From a browser, enter the URL of the protected resource.

   b. Confirm that the login page appears and sign in using proper credentials.

   c. Confirm that the protected resource is served.

   d. Open a new browser tab or window and access the same resource to confirm that the second attempt does not require another login.

   e. Logout from one tab.

   f. Access the resource again to confirm that a login page appears.

2. **Two Applications**:

   a. From a browser, enter the URL of the protected resource.

   b. Confirm that the login page appears and sign in using proper credentials.

c. In a new tab or window, access another protected application.

d. Log out of the first application.

e. Open a new browser tab or window and access the same resource to confirm that the second attempt does not require another login.

f. Access the second application and confirm that the login page appears.

Part IV

Session Management and Life Cycle Management

This part describes session management concepts and procedures for Oracle Access Manager 11g.

Part IV contains the following chapters:

12 Managing Sessions

This chapter describes session management concepts and procedures for Oracle Access Manager 11g. This chapter includes the following topics:

- Prerequisites
- Introduction to User Sessions and Session Management
- Configuring User Session Lifecycle Settings
- Managing Active User Sessions
- Verifying Session Management
- Security

12.1 Prerequisites

The requirements for tasks in this chapter include:

- Reviewing "Introduction to User Sessions and Session Management" on page 12-1
- Getting familiar with Chapter 4, "Managing OAM Server Registration"
- Getting familiar with Chapter 2, "Getting Started with OAM Administration and Navigation"

12.2 Introduction to User Sessions and Session Management

Generally speaking, a user's visit to a Web site is referred to as a session. With Oracle Access Manager 11g, the user must be authenticated through Oracle Access Manager authentication services and must be accessing Oracle Access Manager-protected resources.

Oracle Access Manager 11g session management refers to the process of managing the lifecycle requirements of a user session, and notification of session events to enable global logout.

The Oracle Access Manager 11g Session Management Engine (SME) interfaces with the SSO engine, which acts as the controller for session events and notifications. SME services enable the automatic generation, update, and management of user session data and enable administrators to configure the session lifecycle and to locate and remove specific active sessions.

Note: You can access resources protected by both registered OAM Agents and OSSO Agents during the same session.

Session data storage must be chosen during Oracle Access Manager installation and configuration. The same storage mechanism applies to all servers in a cluster and can be changed after installation.

Session data is stored in multiple tiers to balance latency, availability, and resource consumption. These include:

- The local in-memory cache of each managed Oracle Access Manager server.

  This cache contains session data for use in active server requests. A short TTL is used to quickly evict data that is not currently used.

- A distributed in-memory cache shared by all managed Oracle Access Manager servers.

  This cache contains session data that has been serialized for management by Oracle Coherence. Using Coherence, session data is available to any managed server that an Agent can contact to make access requests involving a session. Coherence also replicates this data across the running servers to provide fault-tolerance. Entries in the distributed cache are evicted not based on a TTL, but overall cache memory size as applied on a per-machine basis.

  If the maximum cache memory size is reached, one of two actions are taken:

  - If the session store is enabled, entries are evicted from the distributed cache to make room. They continue to exist in the database, and can be brought back into the distributed cache if needed.
  - If the session store is not enabled, as a fallback mechanism entries are written to a flat file on the local machine. As the number of entries in this file grows, along with their percentage of the total number of active sessions, performance will degrade accordingly.

  Note: When a user logs out, or when the session expires, session data is automatically deleted from the in-memory store. See "About the User Session Lifecycle" on page 12-3, for more information.

- OAM 11g requires a database to store OAM policy data and (optionally) OAM user session data. The database provides fault-tolerance and scaleability for very large deployments (with hundreds of thousands of simultaneous logins).

  The latest data is written to the session store with each session change (step-up authentication is one example of a session change). This is done asynchronously, and so does not affect latency for the request causing the session to be created or updated. Session data is available even if an unanticipated power failure occurs.

  To store OAM session data requires the database session store extended with the OAM-specific schema:

  - Use RCU with the OAM-specific schema to set up a database as a policy and session data store.
  - Use the Oracle Access Manager with Database Policy Store configuration template to enable OAM to use the database as a policy and session data store.

Oracle Access Manager 11g uses Oracle Coherence to provide a distributed cache with low-data access latencies and to transparently move data between distributed caches (and into the session store). Session data is redundant across these tiers. For example, when a session is created, it then exists within the local cache on the server that created it, the distributed cache, and (if enabled) within the session store database as well. For more information, see "Oracle Coherence and Session Management" on page 12-4.

Administrators can configure the user session lifecycle to define the maximum duration of a user session, the period of inactivity before the user must re-authenticate, and the maximum number of active sessions each user have. The session expiration configuration enables inter-operability with OSSO Agents (mod_osso), which are only visible to the server during user login and logout. For details, see "Configuring User Session Lifecycle Settings" on page 12-6.

Each session is unique and is identified with both a userID and a sessionID to distinguish different sessions for the same user. Administrators can find and delete one or more active sessions for a particular user or for all users. For example, a user with too many open sessions can contact the administrator and request that some or all of his sessions be removed so he can start fresh. For more information, see "Managing Active User Sessions" on page 12-8.

12.2.1 About the User Session Lifecycle

User session lifecycle settings can be defined using the OAM Administration Console. The WebLogic Scripting Tool does not include options for session management.

The lifecycle of a user session refers to the period of user activity from the start of a user session to the end. Session lifecycle states include:

- Active: A session starts when the user is authenticated by Oracle Access Manager. The session remains active as long as the user makes requests for Oracle Access Manager-protected content, and provided that the session has not expired.
- Inactive: A session becomes inactive when the user does not access OAM-protected content for the period defined by the Idle Timeout attribute in the session lifecycle configuration.
- Expired: The duration of the session has exceeded the period defined by the Session Lifetime attribute.

An active session becomes inactive when the user is inactive for the defined Idle Timeout period. A session expires when it exceeds the defined Session Lifetime period.

The Session Management Engine maintains a list of inactive sessions. When an active session becomes inactive, or expires, the user must re-authenticate. Data for expired sessions is automatically deleted from in-memory caches (or the optional SME database). Administrators can delete only active-user-session data.

OSSO GITO Support: The GITO cookie is needed in special cases to support timeout with multiple types of agents (mod_osso and WebGate) working with OAM 11g Server. When enabled (using the `editGITOValues` WLST command), if a user leaves an active session (with an OAM Agent) and starts a session with an OSSO Agent, when he returns to the initial session (with the OAM Agent, now inactive) the Session Management Engine reconciles the period of inactivity with the OAM Agent against the period of activity with the OSSO Agent to enable global logout for the OSSO Agent. The idle timeout is applied appropriately even if the session is operating in a disconnected state (mod_osso requests are being made but none are made by WebGate; to the server, the session appears to idle out).

> **Note:** The Session Management Engine reconciles a period of inactivity with the OAM Agent against a period of activity with the OSSO Agent to enable global logout for the OSSO Agent. For more information, see "mod_osso Cookies" on page 7-14.

User session lifecycle settings for OAM Agents can be defined using the OAM Administration Console. The WebLogic Scripting Tool does not include options for session management.

12.2.2 Oracle Coherence and Session Management

This section describes how the embedded Oracle Coherence data management and caching service is used during session management with the in-memory caches and any database that is configured as an SME session data store.

Note: To maintain the shared session state consistent among the OAM Servers, the Coherence infrastructure requires network connectivity between the cluster members. Oracle recommends the use of redundant networking infrastructure in deployments requiring OAM session data consistency in the presence of network component failures.

Oracle Coherence replicates and distributes session data across all Managed Servers in the cluster. The location of session data is transparent to the client. Oracle Coherence traffic is automatically encrypted. The Session Management Engine exposes session objects to other Oracle Access Manager components as needed. To compensate for data latencies and account for serialization and network transmission of objects, the cache is configured as a near cache to maintain short-lived session objects at the point of consumption.

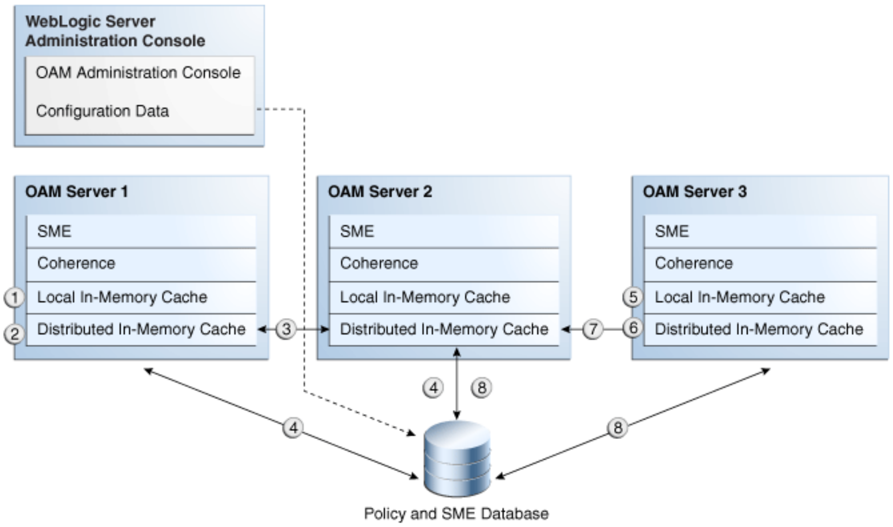
Note: Oracle Coherence traffic is automatically encrypted.

Oracle Coherence also performs failover and reconciliation. For example, if one Managed Server fails, Oracle Coherence automatically distributes data from the failed host to the distributed in-memory caches of other Managed Server hosts.

Figure 12–1 illustrates the storage of session data that occurs using embedded Oracle Coherence. A description follows the diagram.

Note: The OAM Administration Console is an application that resides on the WebLogic AdminServer. Session data is not stored on the AdminServer. To perform session management operations from the System Utilities node of the Administration Console, an OAM Server must be running.

Figure 12–1 Session Data and the Role of Oracle Coherence



Process overview: SSO session data storage after successful authentication

1. The session is created, a sessionID is assigned, and session data is stored in the distributed in-memory cache. A copy is available for a short time in the local in-memory cache on the computer hosting the resource (Managed Server 1 in this example).

2. After a brief period, the local in-memory cache transfers the session data to the distributed in-memory cache on the same host.

   Note: If the distributed in-memory cache runs out of allocated memory space, then the least recently used sessions are evicted from the cache and stored in the database if one was configured. If the Session Management Engine is configured to use just the distributed session store, then the sessions are put in a flat file.

3. With each session change, Oracle Coherence updates, replicates, and distributes session data in the distributed cache among OAM Servers (Managed Server 2 in this example).

   Note: The same session data is stored on only two hosts (the original host and one other).

4. Oracle Coherence also distributes session data from the host of origin to the optional database store, if you are using one.

   Note: Only session data from the host of origin is written to the database store.

5. A new resource request is made and session data is stored in the local in-memory cache on the computer hosting the resource (Managed Server 3 in this example).

6. After a brief period, the local in-memory cache transfers the session data to the distributed in-memory cache on the same host (Managed Server 3 in this example).
7. With each session change, Oracle Coherence updates, replicates, and distributes session data in the distributed cache among OAM Servers (Managed Server 2 and the optional SME database store).

> **Note:** The same session data is stored on only two hosts (the original host and one other). Only session data from the host of origin is written to the database store.

8. A user requesting an OSSO-protected resource occurs within the same active session used by OAM Agents; however, only the OSSO user login and logout are recognized by the OAM Server. You can enable co-existence between agents.

> **Note:** A user can access an OSSO-protected resource while working on OAM-protected resources. Leaving the OAM-protected resource can cause an idle session timeout. However when she returns to the OAM-protected resource, Oracle Coherence reconciles the period of inactivity in the OAM Agent session against the period of activity with the OSSO Agent to enable global logout.

12.3 Configuring User Session Lifecycle Settings

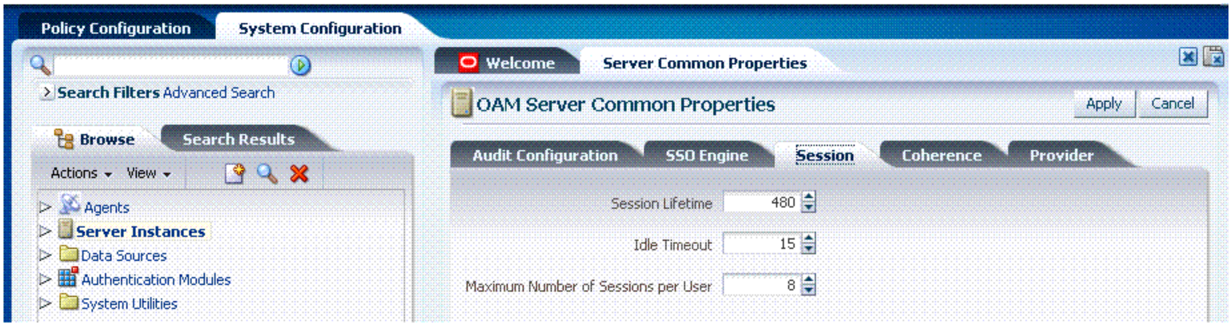
This section provides the following topics:

- About Common Session Lifecycle Setting Page
- Viewing or Modifying Common Session Lifecycle Settings

12.3.1 About Common Session Lifecycle Setting Page

User-session lifecycle settings are part of the OAM Server Common Properties shared by all OAM Servers. Figure 12–2 shows the lifecycle attributes that you can configure.

Figure 12–2 Session Tab under OAM Server Common Properties



OAM Server Common Properties

Session Lifetime 480

Idle Timeout 15

Maximum Number of Sessions per User 8

Table 12–1 describes common session lifecycle settings and their defaults. Sessions can operate in a disconnected mode (mod_osso, for example). Therefore, changes to the configuration establishing your session rules apply only to new sessions. If you need changes to apply immediately, Oracle recommends that you terminate existing sessions and force users to create new ones that adhere to your new rules.

Table 12–1 Common Session Settings

Setting	Description
Session Lifetime	The amount of time, in minutes, that a user's authentication session remains valid. When the lifetime is reached, the session expires. Default = 480 minutes A value of 0 disables this timeout setting. Note: Session data for an expired session is automatically deleted from the in-memory caches (or database).
Idle Timeout	The amount of time, in minutes, that a user's authentication session remains valid without accessing any Oracle Access Manager protected resources. When the user is idle for a longer period, they are asked to re-authenticate. Default = 15 minutes A value of 0 disables this timeout setting. Note: Session data for an inactive session is automatically deleted from the in-memory caches (or database).
Maximum Number of Sessions per User	The exact number of sessions each user can have at one time. Use this setting to configure multiple session restrictions for all users.

12.3.2 Viewing or Modifying Common Session Lifecycle Settings

Users with valid OAM Administrator credentials can use the following procedure to modify common session lifecycle settings using the OAM Administration Console.

> **See Also:** "About Common Session Lifecycle Setting Page" on page 12-6

To view or modify common session lifecycle settings

1. From the Oracle Access Manager Administration Console, click the System Configuration tab.

   The System Configuration navigation tree appears.

2. In the navigation tree, double-click Server Instances.
3. On the OAM Server Common Settings page, click the Session tab to display lifecycle settings.
4. Click the arrow keys beside each list to increase or decrease session lifecycle settings as needed (Table 12–1):
   - Session Lifetime
   - Idle Timeout
   - Maximum Number of Sessions per User
5. Click Apply to submit the changes (or close the page without applying changes).
6. Close the page when you finish.
7. Proceed to "Managing Active User Sessions".

12.4 Managing Active User Sessions

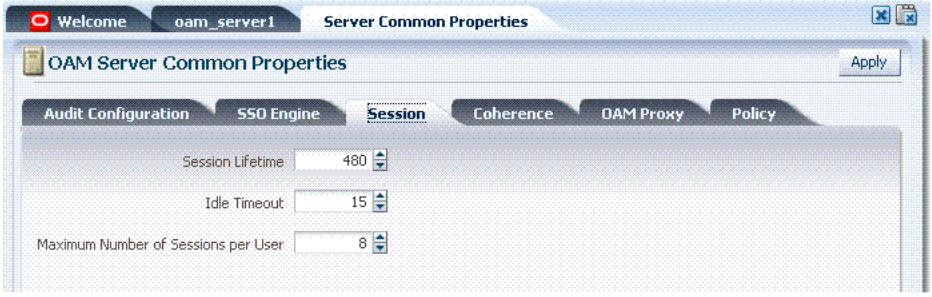
This section describes how to locate and delete one or more sessions for a single user, or for all users.

- About the Session Management Page
- Managing Active User Sessions

12.4.1 About the Session Management Page

Figure 12–3 illustrates the Session Management page, under the System Configuration tab, System Utilities node. Additional details follow the figure.

Figure 12–3 Session Management Page, under System Utilities



Session Lifetime 480

Idle Timeout 15

Maximum Number of Sessions per User 8

Table 12–2 describes Session Management page controls the results table.

Table 12–2 Session Management Page Controls and the Results Table

Tool Bar Icon	Name	Description
N/A	Delete All User Sessions ...	Choose this command button to delete the active sessions of all users. Note: A Confirmation window appears where you can confirm or decline the operation.
N/A	Username	Enter a specific userID in the field and then click the > button to display all active sessions for this user. Note: A complete and accurate userID is required. No wild cards are allowed and no automatic fill-in occurs.
N/A	Tool Bar	Choose commands from the provided menu or command buttons in the tool bar above the results table.
View	View menu	Choose commands from the View menu above the results table to configure the table. Commands include: ■ Columns: Displays a menu with the following options you can use to hide or display specific details in the table: Show All Session ID IP Creation Time Last Accessed Last Updated ■ Detach: Expands the results table to a full-screen view ■ Attach: Restores the Session Management page view. ■ Reorder Columns: Specifies a new order for columns containing session data in the results table.

Table 12–2 (Cont.) Session Management Page Controls and the Results Table

Tool Bar Icon	Name	Description
✖	Delete	Choose this command button after selecting items in the results table to delete. Note: A Confirmation window appears where you can confirm or decline the operation.
Detach	Detach	Click to expand the results table to a full-page view. Note: If the table is already a detached full-page, click Detach to restore the Session Management page.
N/A	Results table (not named)	After searching for the active sessions of a specific user, results are displayed in the table. Details include: ▪ Session ID: A unique, OAM-generated session Id. ▪ IP: The IP address of the specified user. ▪ Creation Time: The day and time the session was created. ▪ Last Accessed: The day and time the session was last accessed ▪ Last Updated: The day and time the session was last updated due to a change.

12.4.2 Managing Active User Sessions

Users with valid OAM Administrator credentials can use information in the following procedure to configure the search results table, locate the active sessions of a specific user, delete one or more sessions for a specific user, or delete all sessions for all users.

> **See Also:** "About the Session Management Page" on page 12-8

Skip any steps that do not apply to your requirements.

Prerequisites

OAM Server must be running.

To locate and manage active sessions

1. From the Oracle Access Manager Administration Console, click the System Configuration tab.

   The System Configuration navigation tree appears.

2. In the navigation tree, click System Utilities.
3. Under System Utilities, double-click Session Management.

   The Session Management page appears with the Username field and a results table.

4. **Configure the Results Table**:

   a. Click View to open the View menu, then click Columns to open the options list.

   b. In the options list, click to check (or clear) items to display (or remove) from the table.

   c. Review the results table to confirm the new setup.

5. **Find sessions for a specific user**:

   a. In the Username field, enter an exact userid.

   b. Click the > button on the right to locate sessions for this user.

   c. Review the results table.

   d. Detach Results Table: Click the Detach command button (or select Detach from the View menu) to display only the table.

6. **Delete sessions for a specific user**:

   a. In the results table, click one or more sessions for the user.

   b. Click the Delete button to delete the selected sessions.

   c. Click Yes to confirm deleting selected sessions (or click No to cancel the delete operation).

   d. Notify the user, if needed.

7. **Delete sessions for all users**:

   a. Click the Delete All User Sessions button in the upper-right corner.

   b. Click Yes when you are asked to confirm.

8. Close the Session Management page when you finish.

9. Proceed to "Verifying Session Management".

12.5 Verifying Session Management

Use the following procedure to verify session management operations.

To validate session management

1. Access a resource from a browser.

2. Under System Utilities in the OAM Administration Console, verify that a user session exists, as described in "Managing Active User Sessions" on page 12-9.

3. Multiple Sessions:

   a. From a different browser (with cookies removed), access a different resource.

   b. Repeat Step 2 and confirm that two sessions exist.

4. In the OAM Administration Console, delete all user sessions, (Step 7 of "Managing Active User Sessions" on page 12-9) and confirm that the active user sessions are removed.

5. Re-authentication:

   a. From the browser in Step 3, attempt to access a different resource.

   b. Confirm that you are prompted for credentials.

6. Verify that session data is created in the database:

   a. From the browser in Step 3, attempt to access a different resource.

   b. Confirm that you are prompted for credentials.

7. Verify that session data is created in the database:

   a. Repeat Step 4 to delete all user sessions.

b. Connect to the database as the OAM user and run the following query to get the results shown.

```
SQL> select * from oam_session
```

c. Confirm that you see the following results:

```
1 row selected
```

d. From the browser in Step 3, attempt to access a different resource.

e. Connect to the database as the OAM user and run the following query

```
SQL> select * from oam_session
```

f. Confirm that you see one row of data:

```
no rows selected
```

g. Select rows from OAM_SESSION_ATTRIBUTES and confirm that data exists for the user.

8. Optimize Logging for Session Management:

   a. Invoke WLST for your platform from the following path. For example:

   ```
   MW_HOME/oracle_common/common/bin/wlst.sh
   ```

   b. Connect to WLST and login.

   c. Execute domainRuntime() and setLogLevel(target="oam_server1",logger="oracle.oam.engine.session",level="FINEST",addLogger=1)

   d. Tail the file <domainhome>/servers/oam_server1/logs/oam_server1-diagnostic.log.

   e. Perform session operations.

   f. View log messages for the Session Management Engine and Session store modules.

   g. Repeat Step c to set level="SEVERE", perform operations and view log messages.

12.6 Security

This section discusses session security for Oracle Access Manager 11g:

- Secure HTTPS Protocol
- Coherence
- Database Persistence

12.6.1 Secure HTTPS Protocol

Oracle Access Manager 11g helps prevent session fixation by providing IP address checks by the Proxy. To further help prevent session fixation, use the secure HTTPS protocol.

12.6.2 Coherence

Data is not encrypted in-memory; however, data is protected over the wire. Coherence communicates between the different OAM instances on various servers. This communication is secured by the following two ways:

- Coherence supports communication only between hosts that have been previously identified.

  This is done as a range of IP addresses, or by specific host names. OAM configuration file contains entries of the servers that participate in the communication. During startup, this information is provided to coherence to ensure that only authorized servers participate in the communication.

- Coherence supports network filters that apply to all communication. Custom filters can be plugged in to provide filtering of required nature

  OAM provides a custom filter that ensures that all communication that occurs between the instances is encrypted/decrypted with a shared key. This 128-bit key is available in the jceks and generated during install

For more information, see the Oracle Coherence documentation.

12.6.3 Database Persistence

The Session Management Engine does not encrypt data.

Session data is not encrypted by Session Management Engine when written to the database.

If you have concerns, use an in-database encryption such as Oracle Advanced Security.

Part V

Logging and Auditing

Part V provides information to help you perform logging and auditing for OAM 11g.

Part V contains the following chapters:

- Chapter 13, "Logging Component Event Messages"
- Chapter 14, "Auditing OAM Administrative and Run-time Events"

13

Logging Component Event Messages

Logging is the mechanism by which components write messages to a file. OAM administrators can use the logging mechanism to capture critical component events. Configuring logging and locating log files are the focus of this chapter. Diagnosing problems using the information in log files is outside the scope of this manual.

This chapter includes the following topics:

- Introduction to Logging OAM Component Events
- Configuring Logging for Oracle Access Manager Using Custom WLST Commands

13.1 Introduction to Logging OAM Component Events

Oracle Access Manager 11g components use the same logging infrastructure and guidelines as any other component in Oracle Fusion Middleware 11g. This is accomplished by using the package `java.util.logging`, which is standard and available in all Java environments. The logging system writes output to flat files only. Logging to an Oracle Database instance is not supported.

Log messages are used for problem diagnosis. The logging infrastructure records messages from OAM components. The administrator controls the amount of information that is logged in a message by specifying log levels for each OAM component for which a logger is defined.

> **Note:** Generally, you enable logging to produce files that you send to Oracle Technical Support for problem diagnosis. Documentation for log messages is not available. In some cases, you might be able to diagnose problems on your own by reading log files.

By default, the log level for all OAM components is the Notification level. Logging at the Error level produces a small amount of output while other log levels can result in voluminous logging output, which can impact OAM performance. In production environments, logging is usually either disabled or the log level is set to a level that results in a small volume of logging output (the error level, for example).

Oracle Access Manager uses the WebLogic container's logging defaults:

- **Logging File**: *DOMAIN_HOME*/servers/*SERVER-NAME*/logs/*SERVER-NAME*-diagnostics.log
- **Logging Configuration File**: Provides logging level and other configuration information for logging. This file is stored in the following path: *DOMAIN_HOME*/config/fmwconfig/servers/*SERVER-NAME*/logging.xml

The following events are logged automatically:

- OAM Server events (managed run-time servers)
- OAM Administrative events (these are generated for configuration changes made using the Administration Console)

> **See Also:** Logging information in the *Oracle Fusion Middleware Application Security Guide*

13.1.1 About OAM Component Loggers

Each OAM component is associated with its own logger name, as listed in the following tables:

- Table 13–1, " OAM Server-Side Components"
- Table 13–2, " OAM Shared-Service Engine Components"
- Table 13–3, " OAM Foundation APIs Components"

Table 13–1 OAM Server-Side Components

Component Name	OAM Logger Name
Protocol Binding	oracle.oam.binding
SSO Controller	oracle.oam.controller.sso
OAM Proxy	oracle.oam.proxy.oam
OSSO Proxy	oracle.oam.proxy.osso
Credential Collector	oracle.oam.credcollector
Remote Registration of Partners	oracle.oam.engine.remotereg
Admin-Console	oracle.oam.admin.console
Admin-Service Config	oracle.oam.admin.service.config
Diagnostics and Monitoring	oracle.oam.diag

Table 13–2 OAM Shared-Service Engine Components

Component Name	OAM Logger Name
Authentication Engine	oracle.oam.engine.authn
Policy Service Engine	oracle.oam.engine.policy
Session Management Engine	oracle.oam.engine.session
Token Engine	oracle.oam.engine.token
SSO Engine	oracle.oam.engine.sso
PartnerTrustMetadata Engine	oracle.oam.engine.ptmetadata
Authorization Engine	oracle.oam.engine.authz

Table 13–3 OAM Foundation APIs Components

Component Name	OAM Logger Name
Session Access	oracle.oam.session.access
Session Access Implementation	oracle.oam.session.accessimpl

Table 13–3 (Cont.) OAM Foundation APIs Components

Component Name	OAM Logger Name
Policy Access	oracle.oam.policy.access

13.1.2 Sample Logger and Log Handler Definition

Example 13–1 illustrates the configuration of a logger and a log handler in the file `logging.xml`.

> **See Also:** Logging information in the *Oracle Fusion Middleware Application Security Guide*

Example 13–1 Configuring Loggers and Log Handlers

```
<logging_configuration>

  <log_handlers>
    <log_handler name='oam-handler' class='oracle.core.ojdl.logging.
    ODLHandlerFactory'>
      <property name='path' value='oam/diagnostic'/>
      <property name='maxFileSize' value='10485760'/>
      <property name='maxLogSize' value='104857600'/>
    </log_handler>
  </log_handlers>

  <loggers>
    <logger name='oracle.security.am' level='NOTIFICATION:1'>
      <handler name='oam-handler'/>
      ...
    </logger>
  </loggers>

</logging_configuration>
```

> **See Also:** For more information about Java EE application logging, see Appendix I, section I.1.1, in *Oracle Fusion Middleware Application Security Guide*.

13.1.3 About Logging Levels

The amount of data output by a logger is controlled by its level; the higher the level, the more information is logged. The level of a logger is specified with the element <logger> in the file logging.xml with the following format:

```
<logger name="loggerName" level="notifLevel"/>
```

where *loggerName* is a logger name (see Table 13–1, Table 13–2, and Table 13–3), and *notifLevel* is either an ODL message level or a Java message level.

Table 13–4 shows the correspondence between ODL message levels and Java message levels, in increasing level order:

Table 13–4 Mapping of ODL to Java Levels

ODL Message Level	Java Message Level
INCIDENT_ERROR:1	SEVERE.intValue()+100
ERROR:1	SEVERE (logs exceptions)

Table 13–4 (Cont.) Mapping of ODL to Java Levels

ODL Message Level	Java Message Level
WARNING:1	WARNING (logs exceptions)
NOTIFICATION:1	INFO (default)
NOTIFICATION:16	CONFIG
NOTIFICATION:32	INFO and CONFIG
TRACE:1	FINE (occasionally recommended in production environments)
TRACE:16	FINER (not recommended in production environments)
TRACE:32	FINEST (not recommended in production environments)

Any other Java level value not listed above (that is, one outside the interval [SEVERE.intValue()+100 - FINEST] is mapped to the ODL level UNKNOWN.

13.2 Configuring Logging for Oracle Access Manager Using Custom WLST Commands

There is no graphical user interface available to change logger levels; only WLST commands can be used. This section provides the following topics:

- Modifying the Oracle Access Manager Logger Level
- Adding an OAM-Specific Logger and Log Handler
-

13.2.1 Modifying the Oracle Access Manager Logger Level

Administrators can use custom WLST commands for OAM to change OAM logger settings as described in the following procedure. Your deployment and choices will be different.

Note: Use the WLST command `help("fmw diagnostics").`

See Also: *Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"*

To modify the OAM logger level

1. Confirm that the OAM Server is running.
2. Acquire the custom WLST script for OAM. For example:

   ```
   <ORACLE_HOME>/common/bin/wlst.sh
   ```

3. Connect to the WebLogic Server and log in as the WebLogic administrator. For example:

   ```
   sh wlst.sh wls:/offline> connect adminID password
   ```

4. List available OAM loggers for the OAM Server. For example:

   ```
   wls:/base_domain/serverConfig> listLoggers(pattern="oracle.oam.*",target="oam_
   server1")
   ```

Here pattern= represents the oam.controller component and target= represents the desired OAM Server as it was specified during registration.

5. View the list of OAM loggers associated with this OAM Server. For example:

```
Logger                                      | Level
--------------------------------------------+-----------------
oracle.oam                                  | <Inherited>
oracle.oam.admin.foundation.configuration   | <Inherited>
oracle.oam.agent-default                    | <Inherited>
oracle.oam.audit                            | <Inherited>
oracle.oam.binding                          | <Inherited>
oracle.oam.commonutil                       | <Inherited>
oracle.oam.config                           | <Inherited>
oracle.oam.controller                       | <Inherited>
oracle.oam.default                          | <Inherited>
oracle.oam.diagnostic                       | <Inherited>
oracle.oam.engine.authn                     | <Inherited>
oracle.oam.engine.authz                     | <Inherited>
oracle.oam.engine.policy                    | <Inherited>
oracle.oam.foundation.access                | <Inherited>
oracle.oam.idm                              | <Inherited>
oracle.oam.idm                              | <Inherited>
oracle.oam.idm                              | <Inherited>
oracle.oam.user.identity.provider           | <Inherited>
```

6. Modify the log level based on your requirements. For example, this sequence changes the log level of the oam.controller to TRACE:32 with no persistence:

```
wls:/base_domain/serverConfig> domainRuntime()
wls:/base_domain/domainRuntime> setLogLevel(logger="oracle.oam.controller",
level="TRACE:32", persist="0", target="oam_server1")
```

7. Repeat step 4 to list the loggers again and verify the log level change. For example:

```
wls:/base_domain/serverConfig> listLoggers(pattern="oracle.oam.*",target="oam_
server1")

Logger                                      | Level
--------------------------------------------+-----------------
oracle.oam                                  | <Inherited>
oracle.oam.admin.foundation.configuration   | <Inherited>
oracle.oam.agent-default                    | <Inherited>
oracle.oam.audit                            | <Inherited>
oracle.oam.binding                          | <Inherited>
oracle.oam.commonutil                       | <Inherited>
oracle.oam.config                           | <Inherited>
oracle.oam.controller                       | <Inherited>
oracle.oam.default                          | <Inherited>
oracle.oam.diagnostic                       | <Inherited>
oracle.oam.engine.authn                     | <Inherited>
oracle.oam.engine.authz                     | <Inherited>
oracle.oam.engine.policy                    | <Inherited>
oracle.oam.foundation.access                | <Inherited>
oracle.oam.idm                              | <Inherited>
oracle.oam.idm                              | <Inherited>
oracle.oam.idm                              | <Inherited>
oracle.oam.user.identity.provider           | <Inherited>
```

8. Verify the generated log file to confirm the controller is logged at the TRACE:32 level:

   ```
   DOMAIN_HOME/server/SERVER_INSTNCE_NAME/logs/
   ```

9. Proceed to "Validating Run-time Event Logging Configuration" on page 13-7.

13.2.2 Adding an OAM-Specific Logger and Log Handler

Administrators can use the following procedure to specify a log file path and necessary attributes. Your deployment and choices might be different.

> **Note:** Use the WLST command `help("fmw diagnostics")` to get more information.

Skip steps 1 through 3 if the following items are true:

- The OAM Server is running
- You have the WLST script
- You have connected to the server and logged in

> **See Also:** *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

To modify the OAM logger, level, and log handler

1. Confirm that the OAM Server is running.
2. Acquire the WLST script. For example:

   ```
   <ORACLE_HOME>/common/bin/wlst.sh
   ```

3. Connect to the WebLogic Server and log in as the WebLogic Administrator. For example:

   ```
   sh wlst.sh wls:/offline> connect
   ```

4. Add an OAM logger and level for the OAM Server. For example:

   ```
   wls:/base_domain/serverConfig> domainRuntime()
   wls:/base_domain/domainRuntime> setLogLevel(logger="oracle.oam",
   level="WARNING", persist="0", target="oam_server1")
   ```

   Here <<<???>>> represents <<<???>>> and target="oam_server1 represents the desired OAM Server.

5. Add a custom log handler and associate it with the OAM logger. For example:

   ```
   wls:/base_domain/domainRuntime> configureLogHandler(name="oam-log-handler",
   target="oam_server1", rotationFrequency="daily", retentionPeriod="week",
   path="${domain.home}/oamlogs" , maxFileSize ="10485760", maxLogSize =
   "104857600", addHandler="true", handlerType="oracle.core.ojdl.logging
   .ODLHandlerFactory", addToLogger="oracle.oam")

   wls:/base_domain/domainRuntime>configureLogHandler(name="oam-log-handler",
   addProperty="true", propertyName="supplementalAttributes", propertyValue=
   "OAM.USER, OAM.COMPONENT", target="oam_server1")
   ```

6. Verify all the OAM logs appear in the DOMAIN_HOME/oamlogs directory:

```
DOMAIN_HOME/oamlogs/
```

13.3 Validating Run-time Event Logging Configuration

You can use the following procedure to test your run-time event logging configuration.

Prerequisites

- Configure logging using WLST commands as described in this chapter.
- Ensure the Agents and Servers are running.
- Configure an application domain to protect the resource as described in Chapter 9, "Managing Policies to Protect Resources and Enable SSO".

To validate run-time event logging

1. In a browser, enter the URL to a protected resource and sign in using an invalid credential.
2. Sign in again using the proper credential.
3. On the physical server, verify all the OAM logs appear in:

   ```
   DOMAIN_HOME/oamlogs/
   ```

4. Open the log file and look for the last entries to confirm authentication failure and success, respectively.

14 Auditing OAM Administrative and Run-time Events

In Oracle Access Manager and Oracle Fusion Middleware 11g, auditing provides a measure of accountability and answers to the "who has done what and when" types of questions. Audit data can be used to create dashboards, compile historical data, and assess risks. Analyzing recorded audit data allows compliance officers to perform periodic reviews of compliance policies.

This chapter describes the Oracle Access Manager administrative and run-time events that can be audited. Configuring common auditing settings for Oracle Access Manager and validating your auditing configuration is the subject of this chapter. Analyzing and using audit data is outside the scope of this chapter.

This chapter includes the following topics:

- Prerequisites
- Introduction to Oracle Access Manager Auditing
- OAM Events You Can Audit
- Setting Up Auditing for Oracle Access Manager
- Validating Oracle Access Manager Auditing and Reports

14.1 Prerequisites

This section identifies requirements for tasks in this chapter: Review Introduction to Oracle Access Manager Auditing

14.2 Introduction to Oracle Access Manager Auditing

Many businesses must now be able to audit identity information and user access on applications and devices. Compliance audits help an enterprise conform with regulatory requirements—Sarbanes-Oxley or the Health Insurance Portability and Accountability Act (HIPAA) are two examples.

Oracle Access Manager uses the Oracle Fusion Middleware Common Audit Framework to support auditing for a large number of user authentication and authorization run-time events, and administrative events (changes to the system). The Oracle Fusion Middleware Common Audit Framework provides uniform logging and exception handling and diagnostics for all audit events.

While auditing can be enabled or disabled, it is normally enabled in production environments. Auditing has minimal performance impact, and the information captured by auditing can be useful (even mission-critical).

> **Note:** Auditing for OAM 10g was based on OAM policies. However, auditing for OAM 11g is based on configuration parameters set in the OAM Administration Console which enables data capture for a user or set of users.

Oracle Access Manager audit data can be written to either a single, centralized Oracle Database instance or to flat files. Regardless of where the audit record is stored, it contains a sequence of items that can be configured to meet particular requirements. The audit log file helps the audit administrator track errors and diagnose problems if the audit framework is not working properly.

Oracle Access Manager integrates with Oracle Business Intelligence Publisher, which provides a pre-defined set of compliance reports:

This section introduces auditing for Oracle Access Manager in the following topics:

- About OAM Auditing Configuration
- About Audit Record Storage
- About Audit Reports and Oracle Business Intelligence Publisher
- About the Audit Log

> **See Also:** Introduction to Oracle Fusion Middleware Audit Framework in the *Oracle Fusion Middleware Application Security Guide*

14.2.1 About OAM Auditing Configuration

An OAM administrator controls certain auditing parameters using the OAM Administration Console. This Oracle Access Manager auditing configuration is recorded in the file `oam-config.xml`. Additional auditing configuration is required through the Common Audit Framework.

> **Note:** The audit configuration is part of oam-config.xml. OAM audit policies cannot be configured using Fusion Middleware Control. OAM does not use JPS infrastructure to configure the audit configuration. There are no WebLogic Scripting Tool (WLST) commands for OAM auditing.

Within the OAM Administration Console, you can set the maximum log file and log directory size. Audit policies (known as Filter Presets in Oracle Access Manager) declare the types of events to be captured by the audit framework for particular components.

> **See Also:**
>
> - "OAM Events You Can Audit" on page 14-4
> - "Setting Up Auditing for Oracle Access Manager" on page 14-10

14.2.2 About Audit Record Storage

By default, Oracle Access Manager records audit data to a file. However, administrators can change the configuration to log audit data to a database. Although the formats differ, audit data content is identical in both the flat file and the database.

Database logging implements the Common Auditing Framework across a range of Oracle Fusion Middleware products. The benefit is audit-function commonality at the platform level.

> **Note:** The preferred mode in production environments is writing audit records to a stand-alone RDBMS database for audit data only.

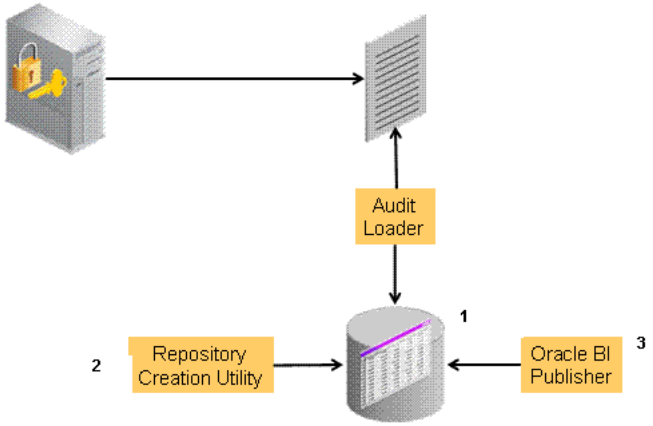
In production environments, Oracle recommends using a database audit store to provide scalability and high-availability for the Common Audit Framework. Audit data is cumulative and grows over time. Ideally this is a database for only audit data; not used by other applications.

To switch to a database as the permanent store for your audit records, you must first use the Repository Creation Utility (RCU) to create a database schema for audit data. The RCU seeds that database store with the schema required to store audit records in a database. After the schema is created, configuring a database audit store involves:

- Creating a data source that points to the audit schema you created
- Configuring the audit store to point to the data source

Figure 14–1 provides a simplified view of the audit architecture with a supported database. The Oracle Fusion Middleware Audit Framework schema for audit log tables is provided by the Repository Creation Utility (RCU), which must be run before you can log information to the database.

Figure 14–1 Audit to Database Architecture



See Also:

- *Oracle Fusion Middleware Application Security Guide*
- "Setting Up the Audit Database Store" on page 14-10

An independent audit loader process reads the flat log file and inserts records in the log table of the Oracle database. The audit store allows administrators to expose audit data with Oracle Business Intelligence Publisher using a variety of out-of-the-box reports.

14.2.3 About Audit Reports and Oracle Business Intelligence Publisher

The data in the database audit store is exposed through pre-defined reports in Oracle Business Intelligence Publisher. These reports allow you to drill down the audit data based on various criteria, such as user name, time range, application type, and execution context identifier (ECID).

Out-of-the-box, there are several sample audit reports available with Oracle Access Manager and accessible with Oracle Business Intelligence Publisher. You can also use Oracle Business Intelligence Publisher to create your own custom audit reports.

Oracle BI Enterprise Edition (Oracle BI EE) is a comprehensive set of enterprise business intelligence tools and infrastructure, including a scalable and efficient query and analysis server, an ad-hoc query and analysis tool, interactive dashboards, proactive intelligence and alerts, real-time predictive intelligence, and an enterprise reporting engine.Oracle BI EE is designed to bring greater business visibility and insight to a wide variety of users.

The components of Oracle Business Intelligence Enterprise Edition share a common service-oriented architecture, data access services, analytic and calculation infrastructure, metadata management services, semantic business model, security model and user preferences, and administration tools. Oracle Business Intelligence Enterprise Edition provides scalability and performance with data-source specific optimized analysis generation, optimized data access, advanced calculation, intelligent caching services, and clustering.

> **See Also:** Using Audit Analysis and Reporting in the *Oracle Fusion Middleware Security Guide*

For an overview of how to prepare Oracle BI EE for use with auditing reports for Oracle Access Manager, see "Preparing Oracle Business Intelligence Publisher EE" on page 14-11.

14.2.4 About the Audit Log

An audit log file helps the audit administrator track errors and diagnose problems when the audit framework is not working properly. An audit log file records several fields including: Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID ContextFields, SessionId, TargetComponentType, ApplicationName, and EventCategory to name a few.

> **See Also:** The topic on audit logs in the chapter on configuring and managing auditing in the *Oracle Fusion Middleware Security Guide*

14.3 OAM Events You Can Audit

This section provides the following topics:

- OAM Administrative Events You Can Audit
- OAM Run-time Events You Can Audit
- About Authentication Event Auditing

14.3.1 OAM Administrative Events You Can Audit

Administrative events are those generated when the OAM Administration Console is used.

The OAM-specific administrative events that can be audited and the details captured in them are listed in Table 14–1. These event definitions and configurations are implemented as part of the audit service in Oracle Platform Security Services.

> **Note:** With OAM 11g, the administrator controls the amount and type of information that is logged by choosing a filter preset from the Audit Configuration tab on the OAM Server Common Properties page.
>
> Auditable events for each filter preset are fixed in the read-only component_events.xml file. Editing or customizing this file is not supported for OAM 11g.

Table 14–1 OAM Administrative Audit Events

Administrative Event	Event Data Include
Administration Console Login success/failure	▪ User name ▪ Remote IP ▪ Roles
Authentication Policy Creation	▪ Policy name ▪ Authentication scheme details ▪ Resource details ▪ Policy type (authentication or authorization)
Authentication Policy Modification	▪ Policy name ▪ Authentication scheme details ▪ Resource details ▪ Policy type (authentication or authorization ▪ Old Policy name ▪ Old Authentication scheme details ▪ Old Resource details
Authentication Policy Removal	▪ Policy name ▪ Authentication scheme details ▪ Resource details ▪ Policy type (authentication or authorization
Resource Creation	▪ Resource name ▪ URI ▪ Operation ▪ Resource type
Resource Modification	▪ Resource name ▪ URI ▪ Operation ▪ Resource type ▪ Old Resource name ▪ Old URI ▪ Old Operation
Resource Removal	▪ Resource name ▪ URI ▪ Operation ▪ Resource type

Table 14–1 (Cont.) OAM Administrative Audit Events

Administrative Event	Event Data Include
Authentication Scheme Creation	▪ Scheme name ▪ Authentication modules ▪ Level
Authentication Scheme Modification	▪ Scheme name ▪ Authentication modules ▪ Level ▪ Old Scheme name ▪ Old Authentication modules ▪ Old Level
Authentication Scheme Removal (Delete)	▪ Scheme name ▪ Authentication modules ▪ Level
Response Creation	▪ Response name ▪ Response key ▪ Data source ▪ Response Type
Response Modification	▪ Response name ▪ Response key ▪ Data source ▪ Response Type ▪ Old Response name ▪ Old Response key ▪ Old Data source
Response Removal (Delete)	▪ Response name ▪ Response key ▪ Data source ▪ Response Type
Partner Addition	▪ Partner name ▪ Partner ID ▪ Partner URL ▪ Logout URL
Partner Modification	▪ Partner name ▪ Partner ID ▪ Partner URL ▪ Logout URL ▪ Old Partner name ▪ Old Partner URL ▪ Old Logout URL
Partner Removal	▪ Partner name ▪ Partner ID ▪ Partner URL ▪ Logout URL
Constraints creation	▪ Constraint Name ▪ Constraint type ▪ Constraint data

Table 14–1 (Cont.) OAM Administrative Audit Events

Administrative Event	Event Data Include
Constraints Modification	▪ Constraint Name ▪ Constraint type ▪ Constraint data ▪ Old Constraint name ▪ Old Constraint type ▪ Old Constraint data
Constraints Removal	▪ Constraint Name ▪ Constraint type ▪ Constraint data
Server Domain creation	▪ Domain Name
Server Domain Modification	▪ Domain Name ▪ Old Domain Name
Server Domain Removal	▪ Domain Name
Server configuration change	▪ New details ▪ Old details ▪ Instance Name ▪ Host Name ▪ Application Name ▪ User Name ▪ Remote ID ▪ Roles ▪ Date and time

14.3.2 OAM Run-time Events You Can Audit

Run-time events are those generated by some of the events the Oracle Access Manager component engines issue when interacting with one another.

The run-time events that can be audited, when they are issued, and the details captured in them are listed in Table 14–2. These event definitions and configurations are implemented as part of the audit service in Oracle Platform Security Services.

> **Note:** With OAM 11g, the administrator controls the amount and type of information that is logged by choosing a filter preset from the Audit Configuration tab on the OAM Server Common Properties page.
>
> Auditable events for each filter preset are fixed in the read-only component_events.xml file. Editing or customizing this file is not supported for OAM 11g.

Table 14–2 OAM Run-time Audit Events

Run-time Event	Issued When	Event Details Include
Authentication Attempt	A user attempts to access a protected resource and the request arrives at the SSO server; this event might be followed by the events credential submit and authentication success or failure.	■ Remote IP ■ Resource ID ■ Partner ID ■ Resource ID ■ Authentication scheme ID ■ Authentication Policy ID
Authentication Success	A client submits credentials and credential validation is successful.	■ Remote IP ■ User Name ■ User DN ■ Resource ID ■ Authentication scheme ID ■ Authentication Policy ID ■ Partner ID
Authentication Failure	A client submits credentials and credential validation fails.	■ Remote IP ■ User Name ■ User DN ■ Resource ID ■ Authentication Scheme ID ■ Failure Error Code ■ Retry count ■ Authentication Policy ID ■ Partner ID
Session Creation	Authentication succeeds.	■ SSO Session ID ■ User Name ■ User DN ■ Remote IP ■ Resource ID ■ Authentication scheme ID ■ Authentication Policy ID
Session Destroy	Authentication succeeds.	■ SSO Session ID ■ User Name ■ User DN ■ Partner ID
Login success	A client finishes the login procedure and it is forwarded to the agent.	■ Remote IP ■ User Name ■ User DN ■ Authentication level ■ Resource ID ■ Authentication scheme ID ■ Authentication Policy ID ■ Partner ID

Table 14–2 (Cont.) OAM Run-time Audit Events

Run-time Event	Issued When	Event Details Include
Login failure	A client fails to login; this event is issued only when all the retry authentication attempts allowed have failed or when the account is locked.	■ Remote IP ■ User Name ■ Authentication level ■ Resource ID ■ Authentication scheme ID ■ Authentication Policy ID ■ Partner ID
Logout success	A client finishes the logout procedure and is forwarded to the agent.	■ Remote IP ■ User DN ■ Authentication level ■ SSO Session ID ■ Partner ID
Logout failure	A client fails to logout.	■ Remote IP ■ User DN ■ SSO Session ID ■ Failure details ■ Partner ID
Credential Collection	A client is redirected to the credential collection page.	■ Remote IP ■ Resource Name ■ Resource ID ■ Authentication scheme ID ■ Authentication Policy ID
Credential Submit	A client submits credentials.	■ Remote IP ■ User Name ■ Resource ID ■ Authentication scheme ID ■ Authentication Policy ID
Authorization Success	A client has been authorized to access a resource.	■ Remote IP ■ User DN ■ Resource ID ■ Authorization Policy ID
Authorization Failure	A client has not been authorized to access a resource.	■ Remote IP ■ User DN ■ Resource ID ■ Authorization Policy ID
Server Start Up	The server starts up.	■ Date and time ■ Instance Name ■ Host Name ■ Application Name ■ User Name
Server Shut Down	The server shuts down.	■ Date and time ■ Instance Name ■ Host Name ■ Application Name ■ User Name

14.3.3 About Authentication Event Auditing

Auditing events during authentication can help administrators scrutinize security weaknesses in their systems. Information about users requesting authentication or brute force attacks can be stored in the file system or in a back-end database.

The events that an administrator can configure for auditing during authentication are:

- Authentication success
- Authentication failure
- Create, modify, delete, or view Authentication Policy data

Information related to the user being authenticated include the following:

- IP address
- Browser type
- User Login ID
- Time of Access

> **Note:** Oracle recommends that you avoid auditing, logging, or tracing sensitive user attributes, such as user passwords.

14.4 Setting Up Auditing for Oracle Access Manager

The following overview provides a list of the tasks that must be performed before you can perform auditing with Oracle Access Manager.

Task overview: Configuring auditing for Oracle Access Manager includes

1. Set up the audit data store, as described in "Setting Up the Audit Database Store" on page 14-10.
2. Set up publishing for audit reports, as described in "Preparing Oracle Business Intelligence Publisher EE" on page 14-11.
3. Edit the Audit Configuration in the OAM Administration Console, as described in:
   - About the Auditing Configuration Page in Oracle Access Manager
   - Adding, Viewing, or Editing Common Audit Settings within Oracle Access Manager
4. Confirm that auditing is working as desired; see: "Validating Oracle Access Manager Auditing and Reports" on page 14-14.

14.4.1 Setting Up the Audit Database Store

This topic provides an overview of the tasks required to create the audit database and extend the schema using the Repository Creation Utility (RCU). This task is required before you can audit events for Oracle Access Manager if you choose a database store for audit data.

See Also:

- *Oracle Fusion Middleware Application Security Guide* for details on managing the audit store
- *Oracle Fusion Middleware Repository Creation Utility User's Guide*

Task overview: Creating the database audit store

1. Create an audit database, version 11.1.0.7 or later, as described in he *Oracle Fusion Middleware Application Security Guide*.
2. Run the Oracle Repository Creation Utility (RCU) against the database, as described in "Create the Audit Schema using RCU" in the *Oracle Fusion Middleware Repository Creation Utility User's Guide*.
3. Set up audit data sources for the audit loader and configure it for the OAM Server as described in "Set Up Audit Data Sources" in the *Oracle Fusion Middleware Application Security Guide*:
   - Use the Java EE audit loader configuration for WebLogic Server.
   - Use the JNDI name of the datasource jdbc/AuditDB that points to the database that was set up in step 2 above
4. In the service instance specified in the domain file (`DOMAIN_HOME/config/fmwconfig/jps-config.xml`), enable database auditing by changing the value of the property `audit.loader.repositoryType` to `DB`. For example:

```
<serviceInstance name="audit" provider="audit.provider">
  <property name="audit.filterPreset" value="None"/>
  <property name="audit.maxDirSize" value ="0"/>
  <property name="audit.maxFileSize" value ="104857600"/>
  <property name="audit.loader.jndi" value="jdbc/AuditDB"/>
  <property name="audit.loader.interval" value="15" />
  <property name="audit.loader.repositoryType" value="DB" />
</serviceInstance>
```

5. Restart the WebLogic Server.
6. Ensure that the audit loader is configured for the OAM Server and that it points to the proper database, as described in "Configure a Database Audit Store for Java Components" in the *Oracle Fusion Middleware Application Security Guide*.
7. Maintain the bus-stop files, as described in "Tuning the Bus-stop Files" in the *Oracle Fusion Middleware Application Security Guide*.

14.4.2 Preparing Oracle Business Intelligence Publisher EE

You must prepare Oracle Business Intelligence Publisher Enterprise Edition (EE) for use with Oracle Access Manager audit reports as outlined in the following procedure.

See Also:

- *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*

Task overview: Prepare Oracle BI Publisher

1. Install Oracle BI Publisher, as described in the *Oracle Business Intelligence Enterprise Edition Installation and Upgrade Guide*.
2. Perform tasks as described in "Set Up Oracle Reports in Oracle Business Intelligence Publisher" in the *Oracle Fusion Middleware Application Security Guide*:

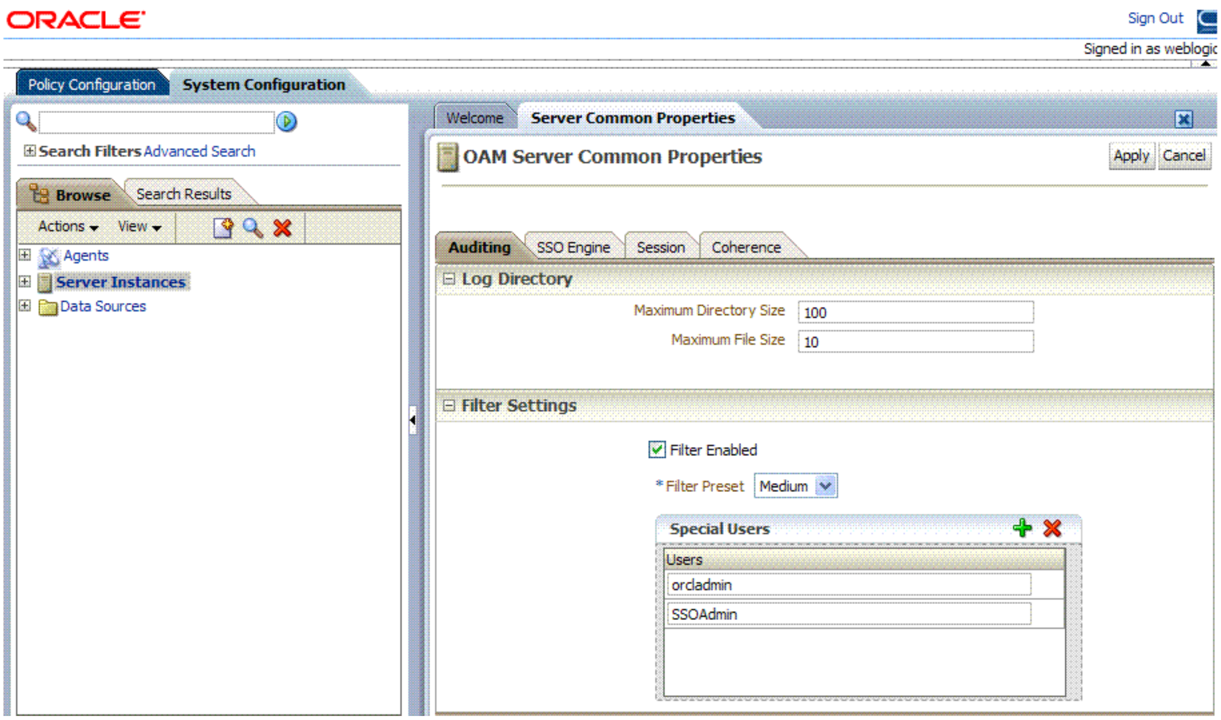
- Unjar the AuditReportTemplate.jar into your Reports folder.
- Set up the JNDI connection for the audit data source or the JDBC connection the audit database

3. Set up audit report templates, as described in the section "Set Up Audit Report Templates" of the *Oracle Fusion Middleware Application Security Guide*.
4. Set up audit report filters, as described in the section "Set Up Audit Report Filters" of the *Oracle Fusion Middleware Application Security Guide*.
5. View reports from the following path: Reports/Oracle_Fusion_Middleware_Audit reports.

14.4.3 About the Auditing Configuration Page in Oracle Access Manager

Within Oracle Access Manager, certain Audit Configuration settings are accessible as OAM Server Common Properties under the System Configuration. These settings are not required when you audit to a database. Figure 14–2 shows the Audit Configuration page.

Figure 14–2 Server Common Properties - Auditing



Log Directory

Maximum Directory Size 100

Maximum File Size 10

Filter Settings

☑ Filter Enabled

* Filter Preset Medium

Users
orcladmin
SSOAdmin

The Auditing page is divided into two sections: The Log Directory section and the Filter Settings section.

> **Note:** The actual log directory cannot be configured using the Administration Console. It is the default directory for the Common Audit Framework audit loader. Changing the directory impacts the audit loader and is not supported.

Table 14–3 describes the elements in the Audit Configuration page.

Table 14–3 Audit Configuration Elements

Elements	Description
Maximum Directory Size	The maximum size, in MBs, of the directory that contains audit output files. For example, assuming that the maximum file size is 10, a value of 100 for this parameter implies that the directory allows a maximum of 10 files. Once the maximum directory size is reached, the audit logging stops. For example, a value of 100 specifies a maximum of 10 files if the file size is 10 MB. If the size exceeds this, the creation of audit logs stops. This is configured using the `max.DirSize` property described in the configuration file`jps-config.xml`. This property controls the maximum size of a bus-stop directory for Java components as described in the *Oracle Fusion Middleware Application Security Guide*.
Maximum File Size	The maximum size, in MBs, of an audit log file. Once the size of a file reaches the maxi mum size, a new log file is created. For example, specifying 10 directs file rotation when the file size reaches 10 MB. This is configured using the `max.fileSize` property described in the configuration file`jps-config.xml`. This property controls the maximum size of a bus-stop file for Java components as described in the *Oracle Fusion Middleware Application Security Guide*.
Filter Enabled	Check this box to enable event filtering.
Filter Preset	Defines the amount and type of information that is logged when the filter is enabled. The default value is Low. ■ All: captures and records all auditable OAM events ■ Low: captures and records a specific set of auditable OAM events ■ Medium: captures and records events covered by the Low setting plus a number of other auditable OAM events ■ None: no OAM events are captured and recorded Events for each filter preset are fixed in the read-only component_ events.xml file. Editing or customizing this file is not supported for OAM 11g. Only items that are configured for auditing at the specified filter preset can be audited.
Special Users	Specifies the list of users whose actions are included only when the filter is enabled. All actions of the special users are audited regardless of the filter preset. Administrators can add, remove or edit special users from this table.

14.4.4 Adding, Viewing, or Editing Common Audit Settings within Oracle Access Manager

The following procedure describes how to add, view, or edit OAM Server Common Audit Configuration settings using the OAM Administration Console.

To view or edit auditing configuration in the Administration Console

1. From the System Configuration tab, navigation tree, double-click Server Instances.

   Alternatively: From any OAM Server page, click the Server Instances link.

2. On the Audit Configuration page, enter appropriate details for your environment (Table 14–3):

   - Log directory and file maximums
   - Filter settings include specific users from the audit.

3. Click Apply to submit the changes (or close the page without applying changes).
4. Restart the OAM Administration Server and OAM Servers after changes are made.

14.5 Validating Oracle Access Manager Auditing and Reports

You can use the following procedure to test your run-time event auditing configuration.

Prerequisites

- Configure server common auditing parameters as described in this chapter.
- Ensure the Agents and Servers are running.
- Configure an application domain to protect the resource as described in Chapter 9, "Managing Policies to Protect Resources and Enable SSO".
- Prepare BI EE Publisher as described in "Preparing Oracle Business Intelligence Publisher EE" on page 14-11.

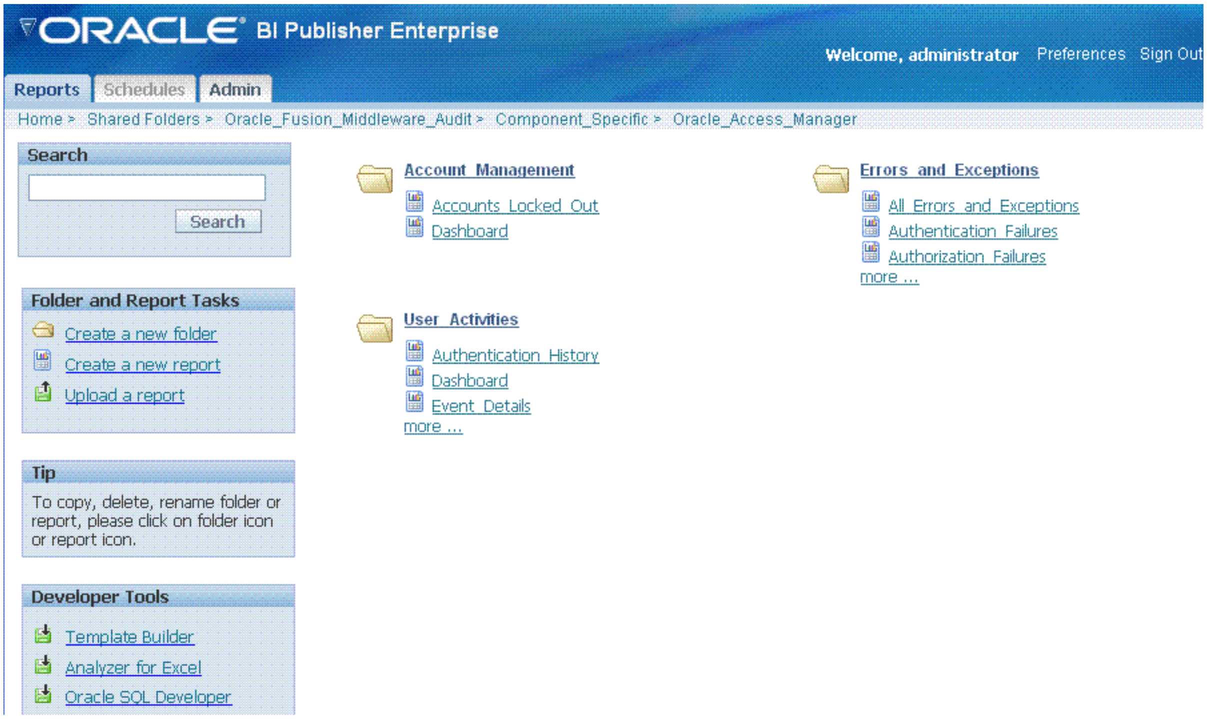
To validate your OAM 11g auditing configuration

1. In a browser, enter the URL to a protected resource and sign in using an invalid credential.
2. Sign in again using the proper credential.
3. Sign in to Oracle BI EE. For example:

   http://*host:port*/xmlpserver

   Here, *host* is the computer hosting Oracle BI Publisher; *port* is the listening port for BI Publisher; xmlpserver is the login page for BI Publisher.
4. In Oracle BI Publisher Enterprise, locate the desired Oracle Access Manager reports. For example:

   Click Shared Folders, click Oracle_Fusion_Middleware_Audit, click Component_Specific, click Oracle_Access_Manager, and then select the desired report, as shown:



To copy, delete, rename folder or report, please click on folder icon or report icon.

5. Perform any analysis as desired, or edit your auditing configuration as needed.

   ```
   /Middleware_home/user_projects/domains/base_domain/servers/oam_server1/logs/
   auditlogs/OAM/
   ```

6. Archive and manage audit logs according to your company policies.

Part VI

Monitoring OAM Performance

Part VI provides information to help you monitor OAM 11g performance.

Part VI contains the following chapters:

- Chapter 15, "Monitoring OAM Metrics by Using Oracle Access Manager"
- Chapter 16, "Monitoring OAM Performance by Using Fusion Middleware Control"

15

Monitoring OAM Metrics by Using Oracle Access Manager

Oracle Access Manager uses the Oracle Dynamic Monitoring Service (DMS) to measure application specific performance information for both OAM Servers and Agents. This chapter provides the following topics:

- Introduction to Monitoring and Metrics by Using OAM
- Monitoring Agents and Servers
- Reviewing Performance Metrics
- Performance Tuning Parameters

> **Note:** This chapter describes how to use Oracle Access Manager to monitor performance metrics. However, you can also use Oracle Enterprise Manager Fusion Middleware Control to monitor Oracle Access Manager performance metrics, as follows:
>
> - Select Oracle Access Manager under Identity Management to go to the home page.
> - On the Home page you can monitor Oracle Access Manager.
> - Select Performance from the Oracle Access Manager menu to view performance metrics.

15.1 Introduction to Monitoring and Metrics by Using OAM

Metric collection is the mechanism by which components collect information in memory for particular events. Based on these events, you can monitor the time spent in a particular area or track particular occurrences or state changes. These metrics are kept only in memory and there are several mechanisms to extract and display them: EM, dmsSpy, dmsDump, for instance.

Administrators can monitor performance for Oracle Access Manager 11g using the Monitoring command from the System Configuration tab, Actions Menu.

- About OAM Proxy Metrics

15.1.1 About OAM Proxy Metrics

The OAM Proxy provides the same or comparable throughput as the Oracle Access Manager 10g Access Server. Throughput refers to the number of requests processed per second.

Latency refers to the time required to process a particular request. There is less than a 20% latency increase with the introduction of a proxy between WebGate and OAM Server.

Performance of the OAM Proxy can be tuned by changing its configuration through the Java EE container Administration Console. Both the Java EE container Administrator and the OAM Administrator can tune performance.

Table 15–1 OAM Proxy Metrics

Metric	Description
handshakes.active	Number of active threads doing handshake
handshakes.avg	Average time spent performing initial handshake
handshakes.completed	Number of times an initial handshake has been executed
handshakes.maxTime	Maximum time spent performing initial handshake
handshakes.minTime	Minimum time spent performing initial handshake
handshakes.time	Total time spent performing initial handshake
failedHandshakes.count	Count of failed handshakes
peerCompatibilityFailures.count	Count of how many Peer Compatibility Check Failures have happened
openSecurityMode.count	Count of how many Open Security Mode handshakes have happened
simpleSecurityMode.count	Count of how many Simple Security mode handshakes have happened
SSLSecurityMode.count	Count of how many SSL Security Mode handshakes have happened
negotiateSecurityMode.active	Number of active threads doing security mode negotiation

15.2 Monitoring Agents and Servers

To monitor an agent or server instance

1. From the System Configuration tab, navigation tree, locate and select the name of the instance to monitor:
   - OAM Agent Name
   - OSSO Agent Name
   - OAM Server Name
2. From the Actions menu, click Monitor and review metrics on the page that opens as described in following steps.
3. OAM Server: On the instance page that opens, view the results.
   - Server Processes Overview
   - Session Operations
   - Server Operations
   - OAM Agents
4. OAM Agent:

- Connectivity
- Operations Overview
- Operations Detail
- Information

5. OSSO Agent: On the instance page that opens, view the results.

   - Processes Overview
   - Operation Detail

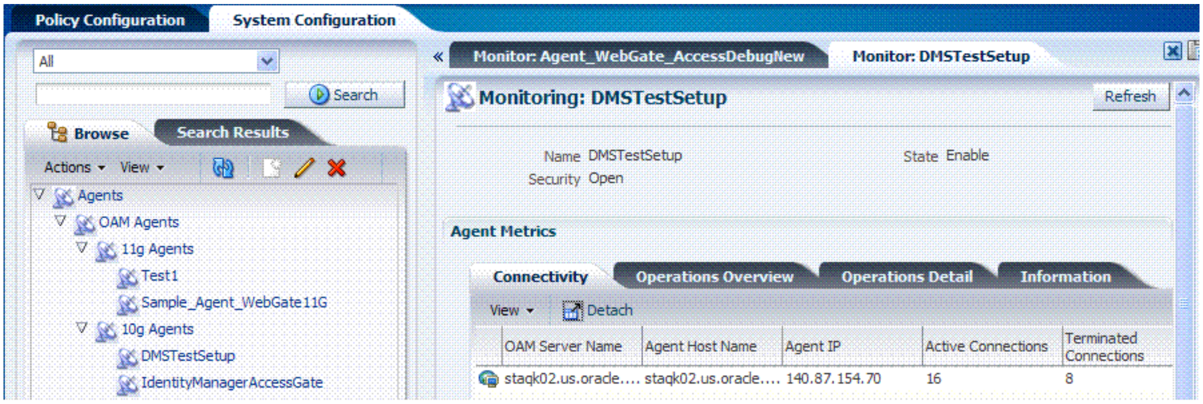
15.3 Reviewing Performance Metrics

This section describes how to review metrics for various components and how to determine whether tuning is needed. The following topics are included:

- Reviewing OAM Agent Metrics
- Reviewing OSSO Agent Metrics

15.3.1 Reviewing OAM Agent Metrics

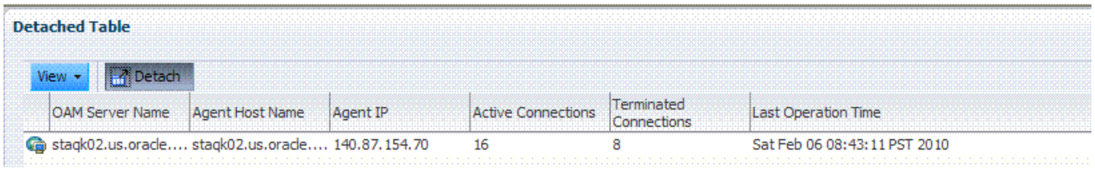
Figure 15–1 OAM 10g Agent Monitoring Page



Name DMSTestSetup
Security Open
State Enable

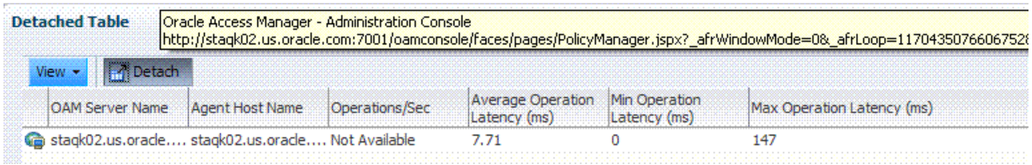
OAM Server Name	Agent Host Name	Agent IP	Active Connections	Terminated Connections
stagk02.us.oracle....	stagk02.us.oracle....	140.87.154.70	16	8

Figure 15–2 Detached OAM 10g Agent Connection Table



OAM Server Name	Agent Host Name	Agent IP	Active Connections	Terminated Connections	Last Operation Time
stagk02.us.oracle....	stagk02.us.oracle....	140.87.154.70	16	8	Sat Feb 06 08:43:11 PST 2010

Figure 15–3 Detached OAM 10g Agent Operations Overview Table

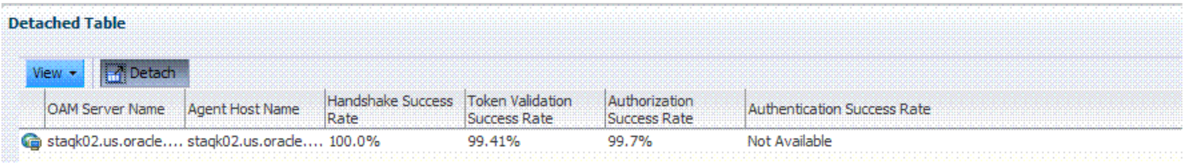


Detached Table

Oracle Access Manager - Administration Console
http://staqk02.us.oracle.com:7001/oamconsole/faces/pages/PolicyManager.jspx?_afrWindowMode=0&_afrLoop=117043507660675

OAM Server Name	Agent Host Name	Operations/Sec	Average Operation Latency (ms)	Min Operation Latency (ms)	Max Operation Latency (ms)
staqk02.us.oracle....	staqk02.us.oracle....	Not Available	7.71	0	147

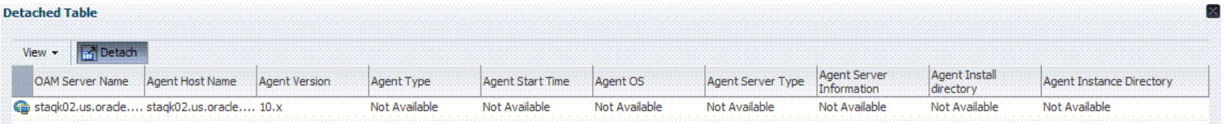
Figure 15–4 Detached OAM 10g Agent Operations Overview Table



Detached Table

OAM Server Name	Agent Host Name	Handshake Success Rate	Token Validation Success Rate	Authorization Success Rate	Authentication Success Rate
staqk02.us.oracle....	staqk02.us.oracle....	100.0%	99.41%	99.7%	Not Available

Figure 15–5 Detached OAM 10g Agent Information Table

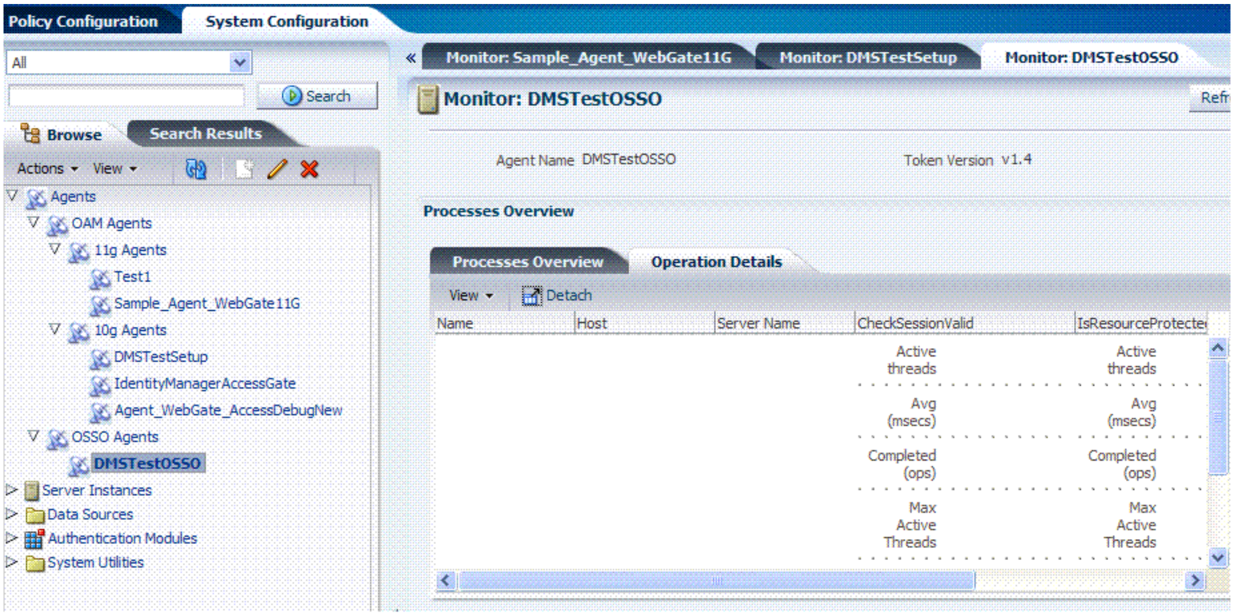


Detached Table

OAM Server Name	Agent Host Name	Agent Version	Agent Type	Agent Start Time	Agent OS	Agent Server Type	Agent Server Information	Agent Install directory	Agent Instance Directory
staqk02.us.oracle....	staqk02.us.oracle....	10.x	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available

15.3.2 Reviewing OSSO Agent Metrics

Figure 15–6 OSSO 10g Agent Monitoring Page with Operation Details



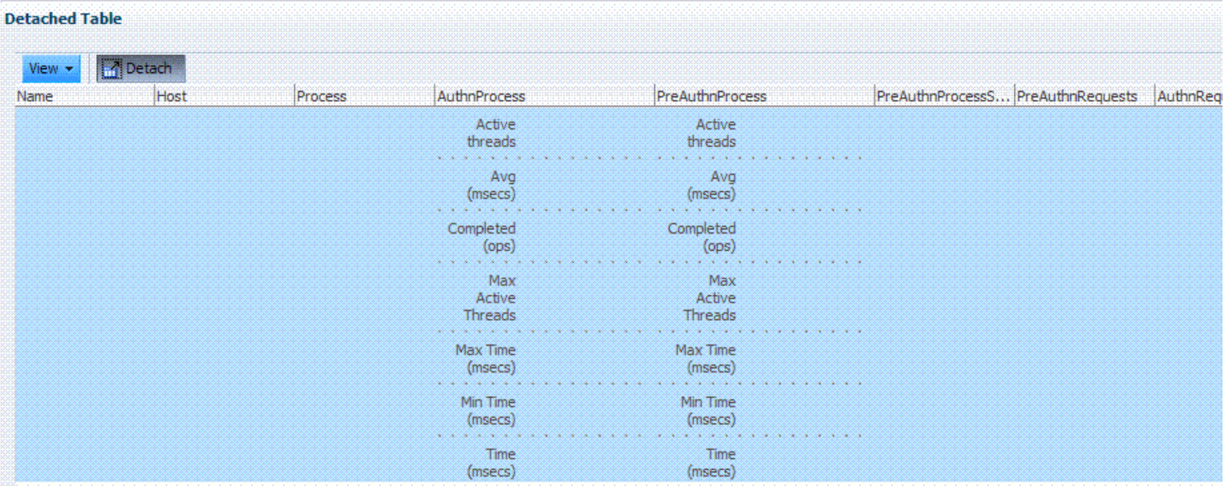
Monitor: DMSTestOSSO

Agent Name DMSTestOSSO

Token Version v1.4

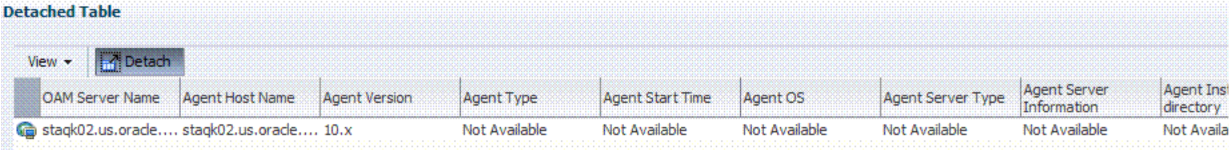
Processes Overview

Figure 15–7 OSSO 10g Agent Monitoring Process Overview Table Detached



Detached Table

Figure 15–8 OSSO 10g Agent Information Table Detached



Detached Table

OAM Server Name	Agent Host Name	Agent Version	Agent Type	Agent Start Time	Agent OS	Agent Server Type	Agent Server Information	Agent Ins directory
staqk02.us.oracle....	staqk02.us.oracle....	10.x	Not Available	Not Available	Not Available	Not Available	Not Available	Not Availa

15.4 Performance Tuning Parameters

This section provides the following information:

15.4.1 OAM Proxy Server Tuning Parameters

Table 15–2 provides the tuning parameters for the OAM Proxy.

Table 15–2 OAM Proxy Tuning Parameters

Purpose	Parameter	Type	Value	Description
Throttle	MaxGlobalBufferSize Note: Proxy server can limit (throttle) the quantity of requests within a specified amount of time not to be exceeded by the proxy server to avoid crashes due to unavailability of resources (like memory. In such cases, a status code is returned indicating that the client should temporarily route requests to other servers	Integer		The maximum memory in KB of the message queue across all the connections. If this value is exceeded, OAM proxy will not accept further requests on a connection. If a value of 0 or less than 0 is specified, this parameter will not be used
Denial of Service Attacks	ConnectionValidationInterval	Integer	120	The time interval in seconds for validating the connections periodically for denial of service attacks
	BacklogQueue	Integer	50	Maximum length of backlog queue
	MaxNAPHandShakeTime	Integer	100	The maximum time in milliseconds within which the client should complete the NAP handshake with client. If NAP handshake over a connection is not completed within this time, the connection will be marked as malicious

16

Monitoring OAM Performance by Using Fusion Middleware Control

This chapter describes how to monitor OAM 11g performance using Oracle Fusion Middleware Control. This chapter focuses on general tasks that administrators can perform for Oracle Access Manager from Fusion Middleware Control. This information does not replace details in the *Oracle Fusion Middleware Administrator's Guide*.

This chapter includes the following topics:

- Prerequisites
- Introduction to OAM 11g and Fusion Middleware Control
- Logging In to and Out of Fusion Middleware Control
- Displaying OAM 11g Menus and Pages in Fusion Middleware Control
- Viewing OAM Performance in Fusion Middleware Control
- Managing Log Level Changes in Fusion Middleware Control
- Displaying OAM MBeans in Fusion Middleware Control
- Displaying Farm Routing Topology in Fusion Middleware Control

16.1 Prerequisites

Oracle Fusion Middleware Control must be deployed with Oracle Access Manager 11g on the WebLogic Administration Server, as described in the Oracle Fusion Middleware Installation Guide for Oracle Identity Management.

16.2 Introduction to OAM 11g and Fusion Middleware Control

Within Fusion Middleware Control OAM information is updated dynamically during live sessions.

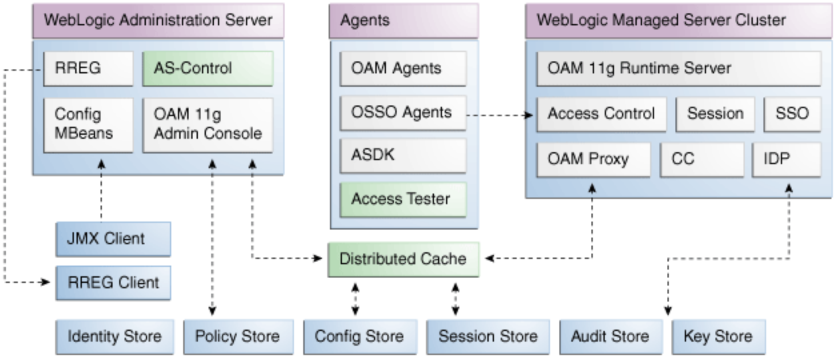
Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct Web-based pages. This helps administrators easily locate the most important monitoring data and the most commonly used administrative functions from a Web browser.

> **Note:** Enterprise Manager Grid Control is an independently licensed product that provides additional capabilities not found in Fusion Middleware Control (primarily, the ability to collect and maintain data for historical purposes and trending).

Oracle Access Manager 11g is deployed as a Java EE application in a WebLogic container. For high availability and failover, OAM is typically deployed in a WebLogic cluster environment. A WebLogic Server domain can have multiple clusters. To provide monitoring and performance statistics for all clustered components requires a composite target. This target provides status and rolled-up load and response performance metrics for member instances. In addition to the metrics exposed for OAM, generic performance metrics are also available for Java EE application and composite Java EE applications.

Fusion Middleware Control must be deployed with Oracle Access Manager 11g on the WebLogic Administration Server, as shown in Figure 16–1.

Figure 16–1 Fusion Middleware Control (AS-Control) Deployment Architecture



Using Fusion Middleware Control for OAM targets is supported through the Oracle Dynamic Monitoring Service instrumentation within OAM. This instrumentation is used to provide:

- OAM performance overview and drill down
- OAM log message searches and dynamic log level changes
- OAM routing topology overview
- OAM Mbean browser

16.3 Logging In to and Out of Fusion Middleware Control

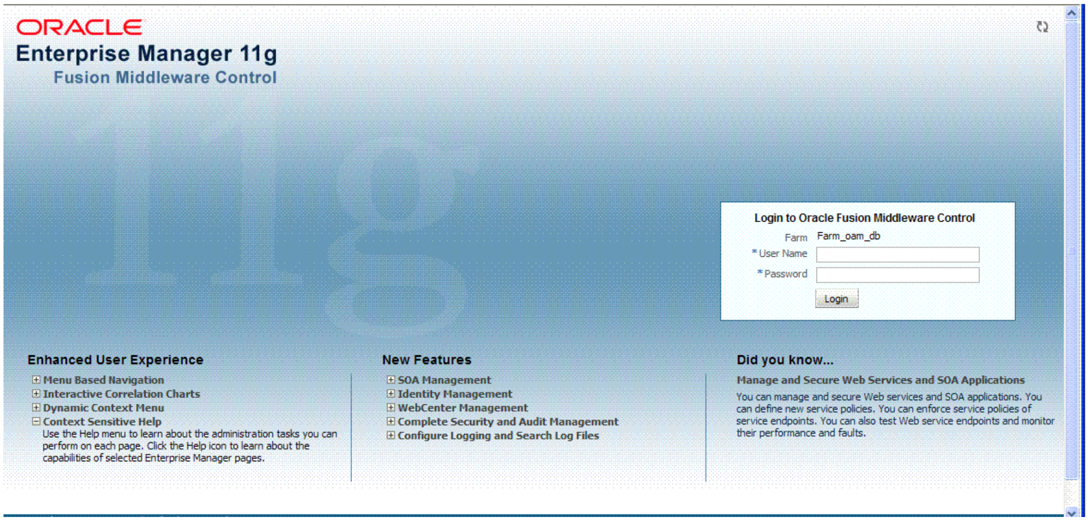
This section provides the following topics:

- About the OAM Farm Page in Fusion Middleware Control
- Logging In To Fusion Middleware Control
- Logging Out of Fusion Middleware Control

16.3.1 About the Login Page for Fusion Middleware Control

The Fusion Middleware Control Login page provides the usual fields for the User Name and Password. The bottom of the Fusion Middleware Control Login page provides topics that you can click for additional information. The Login page is shown in Figure 16–2.

Figure 16–2 Fusion Middleware Control Login Page with Help Topics



Farm Farm_oam_db

Use the Help menu to learn about the administration tasks you can perform on each page. Click the Help icon to learn about the capabilities of selected Enterprise Manager pages.

You can manage and secure Web services and SOA applications. You can define new service policies. You can enforce service policies of service endpoints. You can also test Web service endpoints and monitor their performance and faults.

16.3.2 Logging In To Fusion Middleware Control

Only Fusion Middleware Control administrators can perform this task.

> **See Also:** *Oracle Fusion Middleware Administrator's Guide* for details about getting started using Fusion Middleware Control

To log in to Fusion Middleware Control

1. In a browser window, enter the URL to Fusion Middleware Control. For example:

   ```
   http://host.domain.com:8888/em/
   ```

2. Expand a topic at the bottom of the Login page to learn about the enhanced user experience or new features.
3. Log in as a Fusion Middleware Control administrator.
4. Choose the farm containing OAM 11g, if needed.
5. Help: From the Farm Resource Center on the OAM Farm page, choose topics of interest (or click Help in the upper-right corner of the page) to get more information.
6. Proceed to any topic in this chapter for viewing and configuration details.

16.3.3 Logging Out of Fusion Middleware Control

You can use the following procedure to sign out of Fusion Middleware Control.

To log out of Fusion Middleware Control

1. Click the Log Out link in the upper-right corner of Fusion Middleware Control.
2. Close the browser window.

16.4 Displaying OAM 11g Menus and Pages in Fusion Middleware Control

This section provides the following topics:

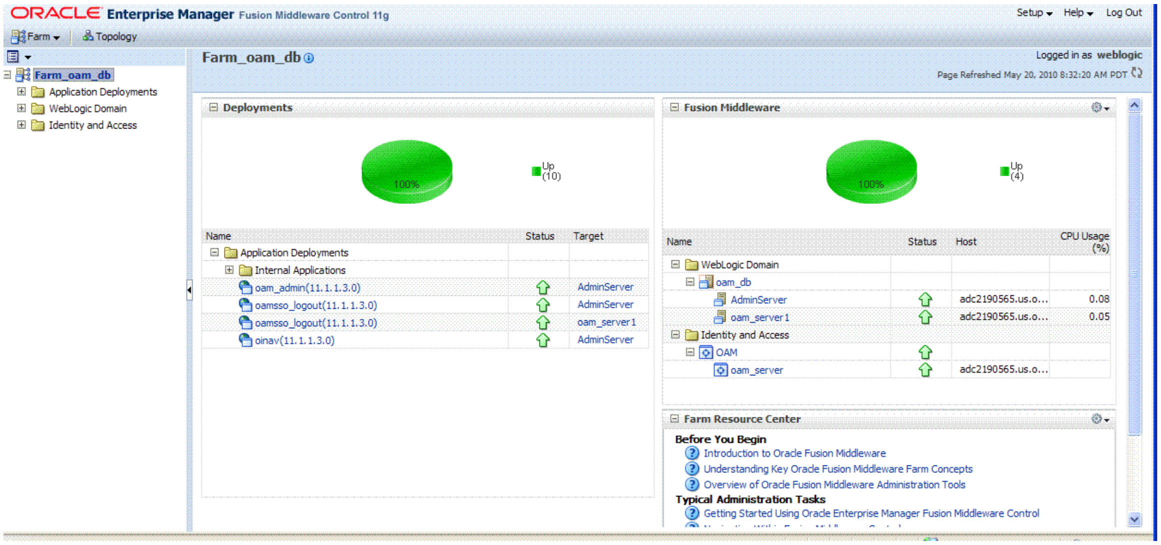
- About the OAM Farm Page in Fusion Middleware Control
- About Context Menus and Pages in Fusion Middleware Control
- Displaying Context Menus and Target Details in Fusion Middleware Control

> **See Also:** *Oracle Fusion Middleware Administrator's Guide* for details about getting started using Fusion Middleware Control

16.4.1 About the OAM Farm Page in Fusion Middleware Control

Figure 16–3 illustrates the OAM Farm page in Fusion Middleware Control. Each Farm page includes similar information. The Farm Resource Center provides immediate access to online information.

Figure 16–3 OAM Farm Page in Fusion Middleware Control



Farm_oam_db

Deployments

Name	Status	Target
Application Deployments		
Internal Applications		
oam_admin(11.1.1.3.0)	⇧	AdminServer
oamsso_logout(11.1.1.3.0)	⇧	AdminServer
oamsso_logout(11.1.1.3.0)	⇧	oam_server1
oinav(11.1.1.3.0)	⇧	AdminServer

Fusion Middleware

Name	Status	Host	CPU Usage (%)
WebLogic Domain			
oam_db			
AdminServer	⇧	adc2190565.us.o...	0.08
oam_server1	⇧	adc2190565.us.o...	0.05
Identity and Access			
OAM	⇧		
oam_server	⇧	adc2190565.us.o...	

Farm Resource Center

Before You Begin

- Introduction to Oracle Fusion Middleware
- Understanding Key Oracle Fusion Middleware Farm Concepts
- Overview of Oracle Fusion Middleware Administration Tools

Typical Administration Tasks

- Getting Started Using Oracle Enterprise Manager Fusion Middleware Control

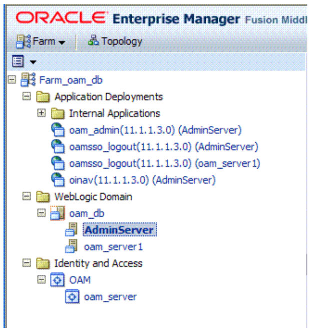
Sections on the Farm page are described in Table 16–1.

Table 16–1 Farm Page Sections

Farm Page Sections	Description
Deployments	Within the farm, this section displays the Status and Target of each Internal Application within the Application Deployment. Clicking any link in the Deployments section (or in the navigation tree) displays a page containing more information.
Fusion Middleware	Within the farm, this section displays the status, host, and CPU usage for server instances in the: ■ WebLogic Server domain ■ Identity and Access Clicking any link on the page (or in the navigation tree) displays a page containing a more detailed summary.
Farm Resource Center	Provides a wealth of online information in the following categories: ■ Information that is useful before you begin using Fusion Middleware Control ■ Administrator tasks using Fusion Middleware Control ■ Other resources Clicking any link in the resource center displays information on the chosen subject. With a wealth of information online, these details are not repeated in this book.

The navigation tree on the left side of the page, like the one in Figure 16–4, enables you to choose a specific instance (target) on which to operate regardless of the page you are currently viewing. Target names in your environment will be different.

Figure 16–4 Farm Navigation Tree in Fusion Middleware Control


ORACLE Enterprise Manager Fusion Middl
Farm
Topology
Farm_oam_db
Application Deployments
Internal Applications
oam_admin(11.1.1.3.0) (AdminServer)
oamsso_logout(11.1.1.3.0) (AdminServer)
oamsso_logout(11.1.1.3.0) (oam_server1)
oinav(11.1.1.3.0) (AdminServer)
WebLogic Domain
oam_db
AdminServer
oam_server1
Identity and Access
OAM
oam_server

For more information, see "Logging In To Fusion Middleware Control".

> **See Also:** "Displaying OAM 11g Menus and Pages in Fusion Middleware Control" on page 16-4

16.4.2 About Context Menus and Pages in Fusion Middleware Control

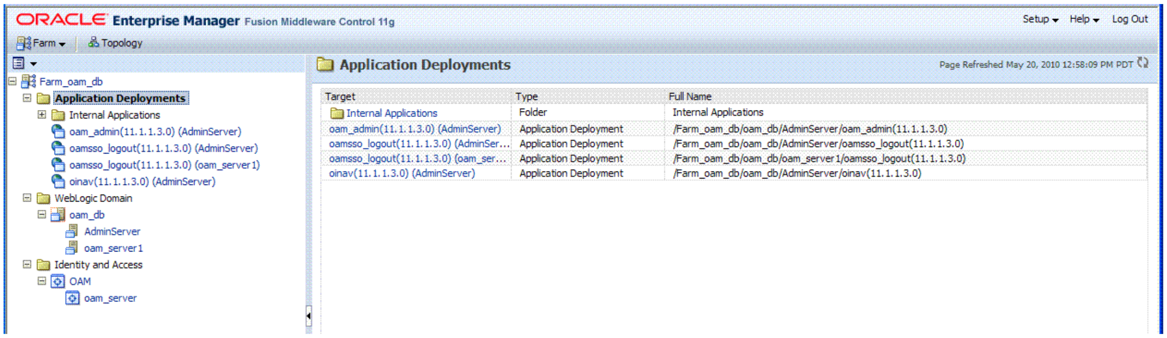
OAM Farm details in Fusion Middleware Control are divided into the following nodes within the navigation tree:

- Application Deployments
- Internal Applications (includes logout page and other details for the OAM AdminServer and OAM Server instances)
- WebLogic Server domains (WebLogic Server details, including the OAM Farm)

- Identity and Access (includes OAM Cluster or individual OAM Server instances)

Clicking a node in the navigation tree displays an information page with individual links and a description of the Target, Type, and Full Name, as shown in Figure 16–5 for Application Deployments.

Figure 16–5 Node Information Page in Fusion Middleware Control


ORACLE Enterprise Manager Fusion Middleware Control 11g
Setup
Help
Log Out
Farm
Topology
Farm_oam_db
Application Deployments
Internal Applications
oam_admin(11.1.1.3.0) (AdminServer)
oamsso_logout(11.1.1.3.0) (AdminServer)
oamsso_logout(11.1.1.3.0) (oam_server1)
oinav(11.1.1.3.0) (AdminServer)
WebLogic Domain
oam_db
AdminServer
oam_server1
Identity and Access
OAM
oam_server

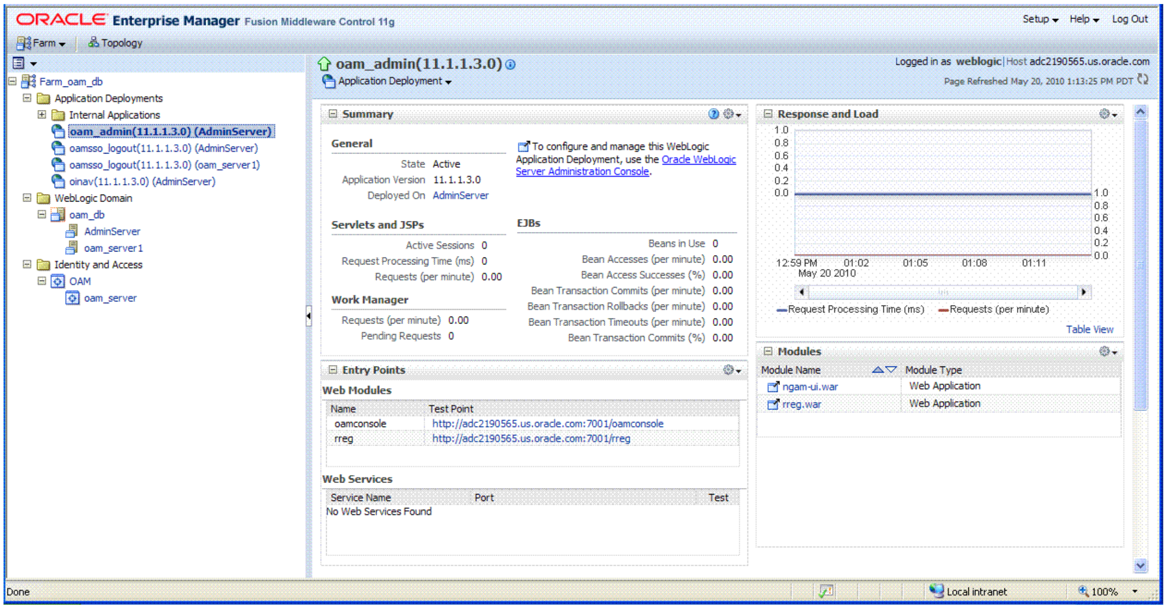
Application Deployments

Page Refreshed May 20, 2010 12:58:09 PM PDT

Target	Type	Full Name
Internal Applications	Folder	Internal Applications
oam_admin(11.1.1.3.0) (AdminServer)	Application Deployment	/Farm_oam_db/oam_db/AdminServer/oam_admin(11.1.1.3.0)
oamsso_logout(11.1.1.3.0) (AdminSer...	Application Deployment	/Farm_oam_db/oam_db/AdminServer/oamsso_logout(11.1.1.3.0)
oamsso_logout(11.1.1.3.0) (oam_ser...	Application Deployment	/Farm_oam_db/oam_db/oam_server1/oamsso_logout(11.1.1.3.0)
oinav(11.1.1.3.0) (AdminServer)	Application Deployment	/Farm_oam_db/oam_db/AdminServer/oinav(11.1.1.3.0)

Clicking an instance (target) name (from either the navigation tree or a page), displays a context menu and a more detailed summary page, as shown in Figure 16–6. Notice that the Internal Application target is highlighted in the navigation tree and a page of the same name is displayed on the right. The context menu is available beneath the target name at the top of the page.

Figure 16–6 Application Deployment Summary for the Selected Internal Application



oam_admin(11.1.1.3.0)

Logged in as **weblogic** | Host adc2190565.us.oracle.com

Page Refreshed May 20, 2010 1:13:25 PM PDT

Summary

General

State Active
Application Version 11.1.1.3.0
Deployed On AdminServer

To configure and manage this WebLogic Application Deployment, use the Oracle WebLogic Server Administration Console.

Servlets and JSPs

Active Sessions 0
Request Processing Time (ms) 0
Requests (per minute) 0.00

Work Manager

Requests (per minute) 0.00
Pending Requests 0

EJBs

Beans in Use 0
Bean Accesses (per minute) 0.00
Bean Access Successes (%) 0.00
Bean Transaction Commits (per minute) 0.00
Bean Transaction Rollbacks (per minute) 0.00
Bean Transaction Timeouts (per minute) 0.00
Bean Transaction Commits (%) 0.00

Entry Points

Web Modules

Name	Test Point
oamconsole	http://adc2190565.us.oracle.com:7001/oamconsole
rreg	http://adc2190565.us.oracle.com:7001/rreg

Web Services

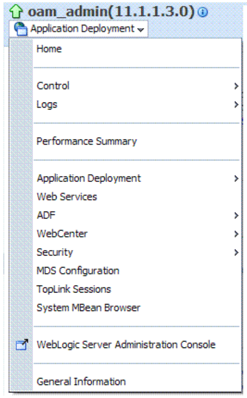
Service Name	Port	Test
No Web Services Found		

Modules

Module Name	Module Type
ngam-ui.war	Web Application
rreg.war	Web Application

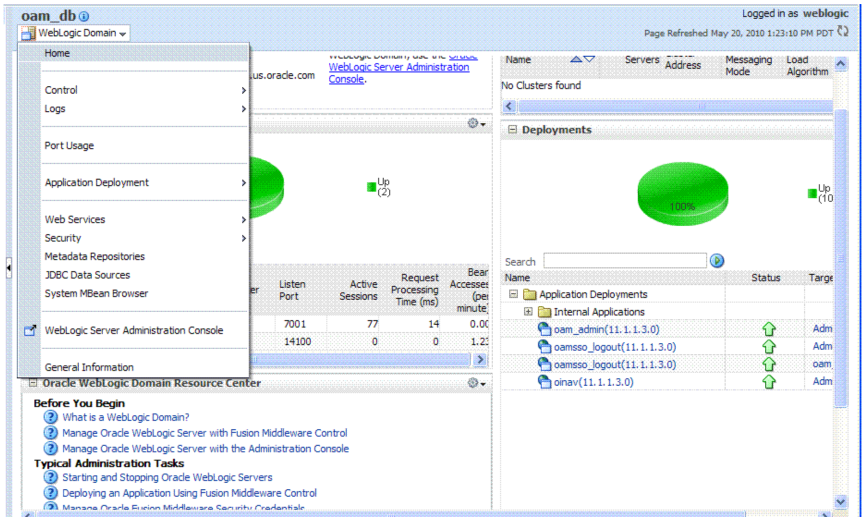
The Application Deployment menu is shown in Figure 16–7.

Figure 16–7 Application Deployment Menu



WebLogic Server domain: The WebLogic Server domain page is shown in Figure 16–7 with the corresponding menu displayed. The Oracle WebLogic Server domain Resource Center, with links to online documentation, is visible in the bottom-left corner. This page more closely resembles the Farm landing page.

Figure 16–8 WebLogic Server Domain Summary with Context Menu Exposed


oam_db
WebLogic Domain
Logged in as weblogic
Page Refreshed May 20, 2010 1:23:10 PM PDT
Home
Control
Logs
Port Usage
Application Deployment
Web Services
Security
Metadata Repositories
JDBC Data Sources
System MBean Browser
WebLogic Server Administration Console
General Information
us.oracle.com
WebLogic Server Administration Console.
Name
Servers
Address
Messaging Mode
Load Algorithm
No Clusters found
Up (2)
Listen Port
Active Sessions
Request Processing Time (ms)
Bean Accesses (per minute)
7001
77
14
0.00
14100
0
0
1.23
Deployments
100%
Up (10
Search
Oracle WebLogic Domain Resource Center
Before You Begin
What is a WebLogic Domain?
Manage Oracle WebLogic Server with Fusion Middleware Control
Manage Oracle WebLogic Server with the Administration Console
Typical Administration Tasks
Starting and Stopping Oracle WebLogic Servers
Deploying an Application Using Fusion Middleware Control
Manage Oracle Fusion Middleware Security Credentials

Name	Status	Targe
Application Deployments		
Internal Applications		
oam_admin(11.1.1.3.0)	⇧	Adm
oamsso_logout(11.1.1.3.0)	⇧	Adm
oamsso_logout(11.1.1.3.0)	⇧	oam
oinav(11.1.1.3.0)	⇧	Adm

Selecting a target name within the WebLogic Server domain node displays a target summary page that more closely resembles the Application Deployment page in Figure 16–5.

For more information, see "Displaying Context Menus and Target Details in Fusion Middleware Control".

> **See Also:** "Viewing OAM Performance in Fusion Middleware Control" on page 16-8 for information about the Identity and Access node and related pages.

16.4.3 Displaying Context Menus and Target Details in Fusion Middleware Control

Fusion Middleware Control administrators can use the following procedure to view context menus and target pages.

Note: From the Farm Resource Center on the OAM Farm page, choose topics of interest (or click Help in the upper-right corner of the page) to get more information.

See Also: "About Context Menus and Pages in Fusion Middleware Control" on page 16-5

To display context menus and target information

1. Log in as described in "Logging In To Fusion Middleware Control" on page 16-3.
2. Expand the Farm containing OAM 11g, if needed.
3. **Information Pages**: From the navigation tree, click one of the following to display the related information page:
   - Application Deployments
   - WebLogic Server domain
   - Identity and Access
4. **Menus and Summary Pages**: Click an instance name (in either the navigation tree or the related page) to display a summary page and menu (Figure 16–6 and Figure 16–7).
5. **OAM Cluster or OAM Server Pages**: See "Viewing OAM Performance in Fusion Middleware Control".

16.5 Viewing OAM Performance in Fusion Middleware Control

Fusion Middleware Control provides administrators with:

- A cluster-wide view of OAM performance
- A per-server drill-down of key performance metrics
- The ability to quickly add or remove performance metrics

Using Fusion Middleware Control, you can view performance metrics for live OAM 11g sessions in a variety of formats. Table 16–2 summarizes the pages for selected nodes and target instances.

Table 16–2 Resulting Pages for Selected Nodes and Targets

Node	Target	Information Summary Page	Performance Overview	Performance Summary w/Metrics
Application Deployment				
Internal Applications	...AdminServer	Yes	No	Yes
	oamsso_logout(11.1.1.3.0) AdminServer	Yes	No	Yes
	oamsso_logout(11.1.1.3.0) oam_server	Yes	No	Yes
WebLogic Server domain				
	oam_bd (Cluster name)	Yes	No	No
	AdminServer	Yes	No	Yes
	oam_server	Yes	No	Yes
Identity and Access				
	OAM (Oracle Access Manager Cluster)	No	Yes	Yes
	oam_server (Oracle Access Manager Server)	No	Yes	Yes

This section provides the following topics:

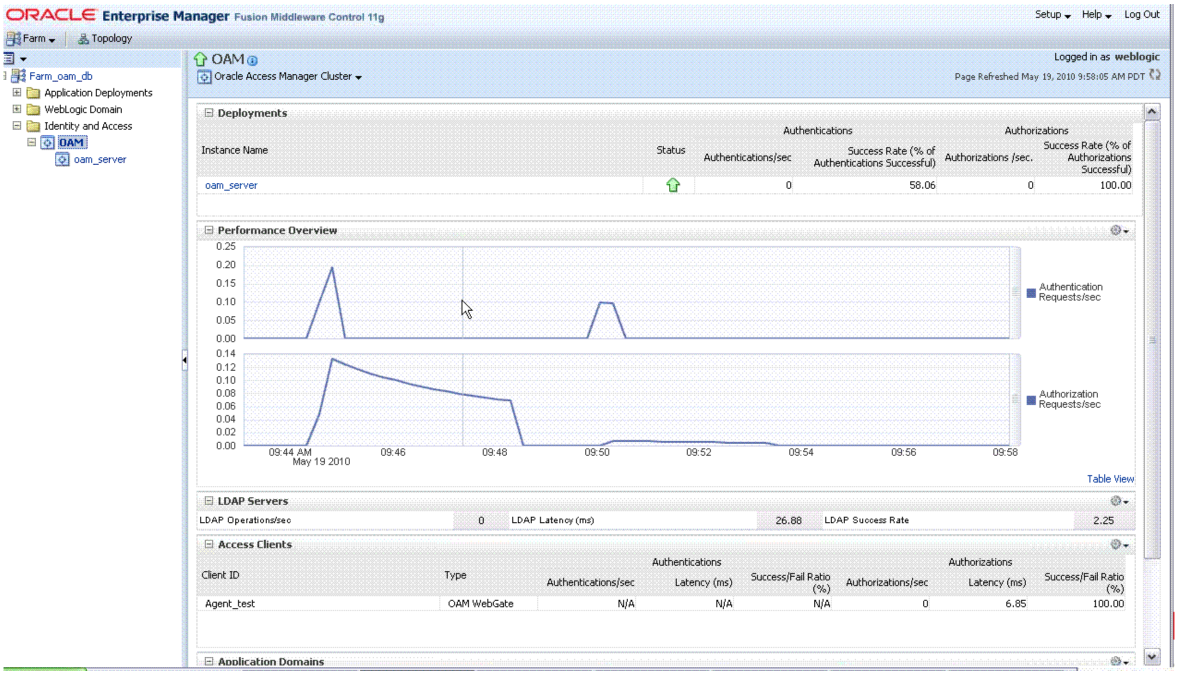
- About OAM Performance Overview Pages in Fusion Middleware Control
- About Metrics and the Performance Summary Page
- Viewing OAM Performance in Fusion Middleware Control
- Configuring OAM Performance Metrics in Fusion Middleware Control

16.5.1 About OAM Performance Overview Pages in Fusion Middleware Control

The Fusion Middleware Control Performance Overview for OAM can be used to reflect WebLogic cluster information down to specific performance metrics for individual OAM Cluster and OAM Server targets.

Oracle Access Manager Cluster Pages: The top node within Identity and Access leads to a page for the OAM Cluster Deployment, which includes a Performance Overview. In Figure 16–9, notice the Oracle Access Manager Cluster selected in the navigation tree, beneath the Identity and Access node. Figure 16–9 illustrates the Oracle Access Manager Cluster Performance Overview and Deployments section.

Figure 16–9 Oracle Access Manager Cluster Performance Overview



Deployments

Instance Name	Status	Authentications Authentications/sec	Success Rate (% of Authentications Successful)	Authorizations Authorizations /sec.	Success Rate (% of Authorizations Successful)
oam_server	⇧	0	58.06	0	100.00

LDAP Servers

LDAP Operations/sec	0	LDAP Latency (ms)	26.88	LDAP Success Rate	2.25

Access Clients

Client ID	Type	Authentications Authentications/sec	Latency (ms)	Success/Fail Ratio (%)	Authorizations Authorizations/sec	Latency (ms)	Success/Fail Ratio (%)
Agent_test	OAM WebGate	N/A	N/A	N/A	0	6.85	100.00

OAM Server Pages: Selecting an OAM target name from the navigation tree (or the open page), displays a Performance Overview for the target with a summary of Key Metrics at the top (replacing the Deployment section, which appears on the Cluster page).

Table 16–3 describes the elements of the Performance Overview for OAM Clusters and OAM Server instances in Fusion Middleware Control. There are only a few differences.

Table 16–3 Summary of OAM Performance Overviews in Fusion Middleware Control

Element	Description
Deployments	This section appears only on OAM Cluster pages to describe the status of instances in the overview.
Instance Name	The name of the OAM Server to which the metrics apply. For example: *OAM_server_name*
Status	The status of the OAM Server instance, illustrated with one of the following: ■ Green Up Arrow (running) ■ Red Down Arrow (not running)
Authentications/sec	The number of authentications per second.
Success Rate (% of Authentications Successful)	A numeric value representing the percentage of successful authentications.
Authorizations/sec	The number of authorizations per second.
Success Rate (% of Authorizations Successful)	A numeric value representing the percentage of successful authorizations.

Table 16–3 (Cont.) Summary of OAM Performance Overviews in Fusion Middleware Control

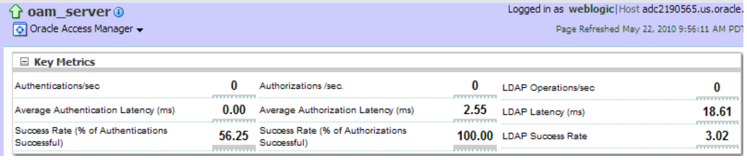
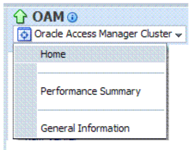
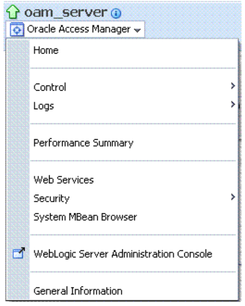
Element	Description
Key Metrics	Summary of statistics displayed only for the selected OAM Server instance.  oam_server Oracle Access Manager Logged in as weblogic | Host adc2190565.us.oracle. Page Refreshed May 22, 2010 9:56:11 AM PDT Key Metrics Authentications/sec 0 · Authorizations /sec. 0 · LDAP Operations/sec 0 Average Authentication Latency (ms) 0.00 · Average Authorization Latency (ms) 2.55 · LDAP Latency (ms) 18.61 Success Rate (% of Authentications Successful) 56.25 · Success Rate (% of Authorizations Successful) 100.00 · LDAP Success Rate 3.02
Oracle Access Manager Menus	Dynamic context menus provide functions related to the selected target (also available when you right-click a target in the navigation tree). The Oracle Access Manager Cluster menu is shown here:  OAM Oracle Access Manager Cluster Home Performance Summary General Information The OAM Server menu is shown next.  oam_server Oracle Access Manager Home Control Logs Performance Summary Web Services Security System MBean Browser WebLogic Server Administration Console General Information
Performance Overview	This section provides a graphic representations of authentication and authorization. The date is identified beneath the tables, which include information on: ■ Authentication Requests/sec ■ Authorization Requests/sec Within each table: ■ Coordinates along the horizontal axis (the x axis) identify the time period. ■ Coordinates along the vertical axis (the y axis) identify the number of authentications (or authorizations) during the time period. Note: The metrics on the Performance Overview are not configurable. The Metrics Palette is available for only the Performance Summary.
Table View	Click this item on the right side of the page to display the Performance Overview in tabular format within a pop up window.
LDAP Servers	This section is available only when the OAM Cluster is selected to provide information for the primary LDAP user identity store: ■ LDAP operations/sec ■ LDAP Latency (milliseconds) ■ LDAP Success Rate
Access Clients	This section of the page provides information for all active Access Clients.
Client ID	Displays the name of the Agent, as defined in the Agent registration in the OAM Administration Console. For example: Agent_test

Table 16–3 (Cont.) Summary of OAM Performance Overviews in Fusion Middleware Control

Element	Description
Type	Displays the Agent. type For example: OAM WebGate
Authentications/sec	Displays the number of authentications per second processed by this Agent.
Authentications Latency (ms)	Displays the number of milliseconds the authentication was delayed.
Success/Fail Ratio (%)	Displays a numeric value representing the percentage of successful authentications as compared to failed authentications.
Authorizations/sec	Displays the number of authorizations per second processed by this Agent.
Authorizations Latency (ms)	Displays the number of milliseconds the authorization was delayed.
Success/Fail Ratio (%)	Displays a numeric value representing the percentage of successful authorizations as compared to failed authorizations.
Application Domains	This section of the page provides information for all Application Domains that were used during authentication and authorization processing.
Application Domain Name	Displays the name of the application domain that contains the authentication and authorization policies used for the request.
Authentications/sec	Within the named application domain, the number of authentications per second processed by this Agent.
Authentications Latency (ms)	Within the named application domain, the number of milliseconds the authentication was delayed.
Success/Fail Ratio (%)	Within the named application domain, the numeric value representing the percentage of successful authentications as compared to failed authentications.
Authorizations/sec	Within the named application domain, the number of authorizations per second processed by this Agent.
Authorizations Latency (ms)	Within the named application domain, the number of milliseconds the authorization was delayed.
Success/Fail Ratio (%)	Within the named application domain, the numeric value representing the percentage of successful authorizations as compared to failed authorizations.
Authentications/sec	Within the named application domain, the number of authentications per second processed by this Agent.
Authorizations/sec	Within the named application domain, the number of authorizations per second processed by this Agent.

16.5.2 About Metrics and the Performance Summary Page

The Performance Summary command on the Oracle Access Manager Cluster or OAM Server menu displays metrics charts for the selected target. On the Performance Summary page, a chart is displayed for each selected metric.

An OAM Server Performance Summary page is shown in Figure 16–10 with an open Metric Palette from which you can choose the exact metrics you want to chart. Stacked charts allow you to easily compare multiple metrics for the same time frame, change the time frame to go back in time, or zoom in or out.

Figure 16–10 Performance Summary Page with Metric Palette

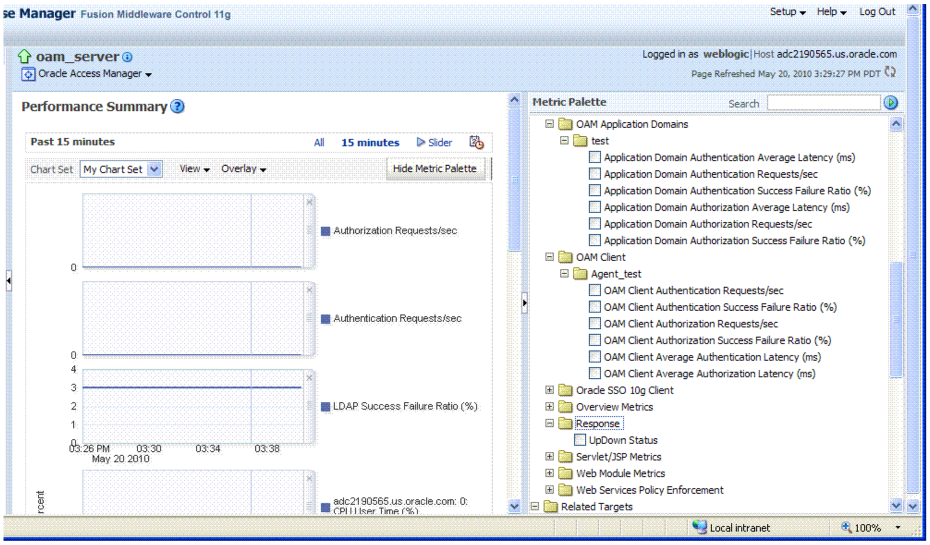

se Manager Fusion Middleware Control 11g
Setup
Help
Log Out
oam_server
Oracle Access Manager
Logged in as weblogic | Host adc2190565.us.oracle.com
Page Refreshed May 20, 2010 3:29:27 PM PDT
Performance Summary
Past 15 minutes
All
15 minutes
Slider
Chart Set
My Chart Set
View
Overlay
Hide Metric Palette
Authorization Requests/sec
Authentication Requests/sec
LDAP Success Failure Ratio (%)
adc2190565.us.oracle.com: 0: CPU User Time (%)
0
0
4
3
2
1
0
03:26 PM
May 20 2010
03:30
03:34
03:38
rcent
Metric Palette
Search
OAM Application Domains
test
Application Domain Authentication Average Latency (ms)
Application Domain Authentication Requests/sec
Application Domain Authentication Success Failure Ratio (%)
Application Domain Authorization Average Latency (ms)
Application Domain Authorization Requests/sec
Application Domain Authorization Success Failure Ratio (%)
OAM Client
Agent_test
OAM Client Authentication Requests/sec
OAM Client Authentication Success Failure Ratio (%)
OAM Client Authorization Requests/sec
OAM Client Authorization Success Failure Ratio (%)
OAM Client Average Authentication Latency (ms)
OAM Client Average Authorization Latency (ms)
Oracle SSO 10g Client
Overview Metrics
Response
UpDown Status
Servlet/JSP Metrics
Web Module Metrics
Web Services Policy Enforcement
Related Targets
Local intranet
100%

Table 16–4 describes the status and controls available on the Performance Summary page.

Table 16–4 Status and Controls on Performance Summary Pages

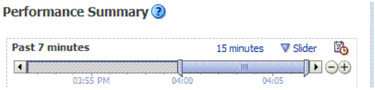
Status or Control	Description
Past *n* minutes	Status is based on the specified time period, which can be adjusted using the slider.
All	
n Minutes	The specified time period, which can be adjusted using the slider.
Slider	The tool you use to adjust the time period.  Performance Summary Past 7 minutes 15 minutes Slider 03:55 PM 04:00 04:05
Chart Set	A list from which you can choose the set of saved charts to view.

Table 16–4 (Cont.) Status and Controls on Performance Summary Pages

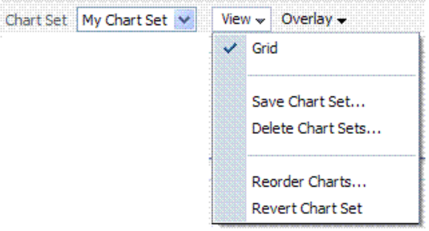
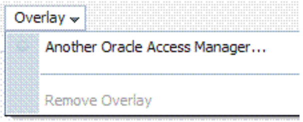
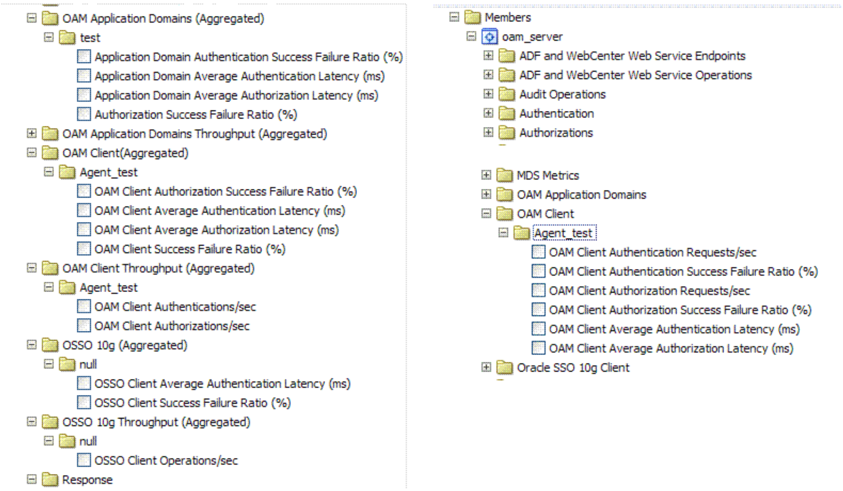
Status or Control	Description
View	A menu that enables you to add a grid, save a chart, and order information on the page. 
Overlay	A menu that enables you to search for and view another instance of the same type and compare this against the instance in the summary. 
Metric Palette	A listing from which you can select performance metrics to chart. Items unique to Oracle Access Manager 11 are shown here. 

Chart Set
My Chart Set
View
Overlay
✓ Grid
Save Chart Set...
Delete Chart Sets...
Reorder Charts...
Revert Chart Set

Overlay
Another Oracle Access Manager...
Remove Overlay

OAM Application Domains (Aggregated)
test
□ Application Domain Authentication Success Failure Ratio (%)
□ Application Domain Average Authentication Latency (ms)
□ Application Domain Average Authorization Latency (ms)
□ Authorization Success Failure Ratio (%)
OAM Application Domains Throughput (Aggregated)
OAM Client(Aggregated)
Agent_test
□ OAM Client Authorization Success Failure Ratio (%)
□ OAM Client Average Authentication Latency (ms)
□ OAM Client Average Authorization Latency (ms)
□ OAM Client Success Failure Ratio (%)
OAM Client Throughput (Aggregated)
Agent_test
□ OAM Client Authentications/sec
□ OAM Client Authorizations/sec
OSSO 10g (Aggregated)
null
□ OSSO Client Average Authentication Latency (ms)
□ OSSO Client Success Failure Ratio (%)
OSSO 10g Throughput (Aggregated)
null
□ OSSO Client Operations/sec
Response
Members
oam_server
ADF and WebCenter Web Service Endpoints
ADF and WebCenter Web Service Operations
Audit Operations
Authentication
Authorizations
MDS Metrics
OAM Application Domains
OAM Client
Agent_test
□ OAM Client Authentication Requests/sec
□ OAM Client Authentication Success Failure Ratio (%)
□ OAM Client Authorization Requests/sec
□ OAM Client Authorization Success Failure Ratio (%)
□ OAM Client Average Authentication Latency (ms)
□ OAM Client Average Authorization Latency (ms)
Oracle SSO 10g Client

16.5.3 Configuring OAM Performance Metrics in Fusion Middleware Control

Fusion Middleware Control administrators can use the following procedure to add or change the metrics that are displayed in the OAM Performance Summary.

See Also:

- "About OAM Performance Overview Pages in Fusion Middleware Control"
- "About Metrics and the Performance Summary Page"

To add or change metrics in the Performance Summary

1. Log in as described in "Logging In To Fusion Middleware Control" on page 16-3.

2. Expand the desired node and select a target. For example:

   Identity and Access
     oam_server

3. From the OAM context menu, select Performance Summary.
4. From the Performance Summary page, click Show Metrics Palette.
5. From the Metrics Palette, expand nodes and check (or clear) boxes to add (or remove) metrics from the summary.
6. Review the updated the Summary page.
7. Click Hide Metrics Palette when you finish.
8. **Saving a Chart Set**:
   a. From the View menu on the Performance Summary page, click Save Chart Set.
   b. In the dialog box that appears, enter a unique name for this chart set and click OK when the operation is confirmed.
9. Proceed to "Viewing OAM Performance in Fusion Middleware Control".

16.5.4 Viewing OAM Performance in Fusion Middleware Control

Fusion Middleware Control administrators can use the following procedure to view and compare OAM performance data for either the entire OAM Cluster or a single OAM Server instance.

See Also:

- "About OAM Performance Overview Pages in Fusion Middleware Control"
- "About Metrics and the Performance Summary Page"

To view OAM performance

1. Log in as described in "Logging In To Fusion Middleware Control" on page 16-3.
2. **Performance Overview**:
   a. Expand the desired node and select a target. For example: Identity and Access.

      Identity and Access
        oam_server

   b. Review the Performance Overview.
   c. Notice the target menu.
3. **Performance Summary**: From the context menu (Step 2), select Performance Summary. Otherwise,
   a. Expand the desired node and select a target. For example: Identity and Access.

      Identity and Access
        oam_server

   b. From the context menu, select Performance Summary.
   c. Review the Summary Page.

4. **Changing Metrics**: See "Configuring OAM Performance Metrics in Fusion Middleware Control" on page 16-14.

5. **Saving a Chart Set**:

   a. From the View menu on the Performance Summary page, click Save Chart Set.

   b. In the dialog box that appears, enter a unique name for this chart set and click OK when the operation is confirmed.

   c. Click Hide Metrics Palette when you finish.

   d. Review the updated information on the Summary Page.

6. **Adding an Overlay**:

   a. From the Overlay menu on the Performance Summary page, click Another Oracle Access Manager.

   b. In the Search and Select Targets dialog, enter the target name and host name, then click Go.

   c. In the target results table, click the name of the desired target and then click Select.

   d. When finished viewing the overlay, click Remove Overlay from the Overlay menu.

7. **Testing**:

   a. Using the Access Tester, perform several authentication and authorization tests (see Chapter 10).

   b. In Fusion Middleware Control, check performance metrics.

16.6 Managing Log Level Changes in Fusion Middleware Control

Oracle Fusion Middleware components generate log files containing messages that record all types of events. Administrators can set log levels using Fusion Middleware Control, as described in this chapter.

> **Note:** Alternatively, administrators can set OAM logger levels using custom WebLogic Scripting Tool (WLST) commands, as described in Chapter 13.

Topics in this section include:

- About Dynamic Log Level Changes
- Setting OAM Log Levels Dynamically Using Fusion Middleware Control

16.6.1 About Dynamic Log Level Changes

Using Fusion Middleware Control, administrators can change OAM 11g log levels dynamically.

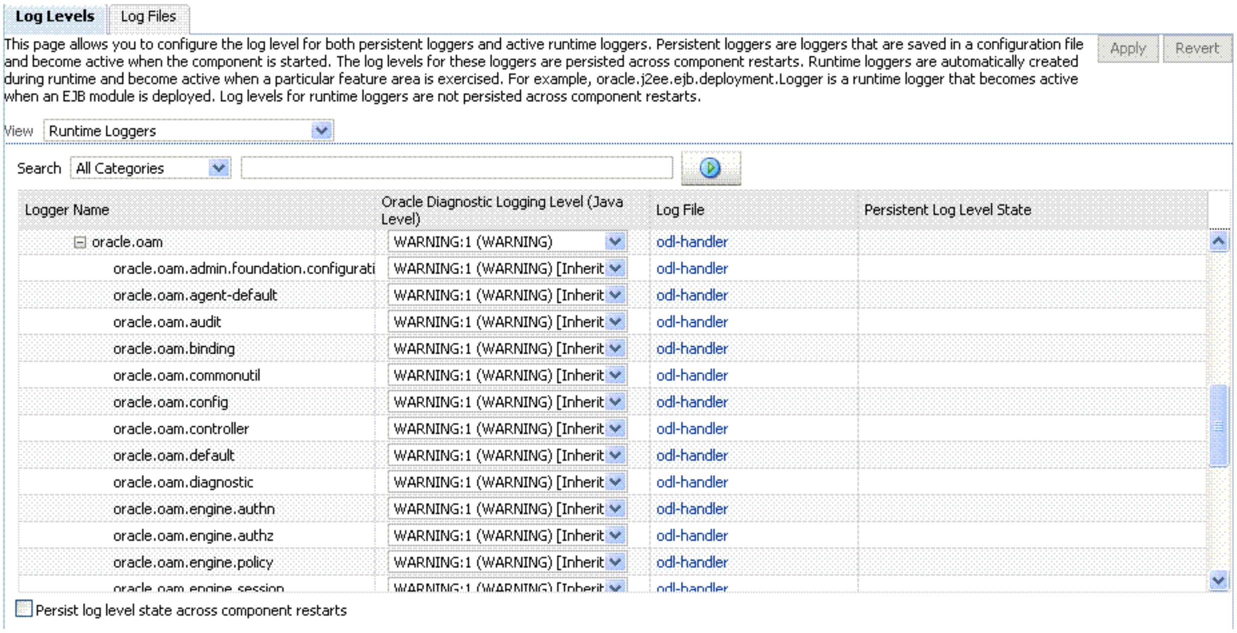
Table 16–5 outlines OAM 11g log availability and functions in Fusion Middleware Control

Table 16–5 OAM Log Availability and Functions in Fusion Middleware Control

Node	Target	View Log Messages	Log Configuration
Application Deployment			
Internal Applications	...AdminServer	Yes	Yes
	oamsso_logout(11.1.1.3.0) AdminServer	Yes	Yes
	oamsso_logout(11.1.1.3.0) oam_server	Yes	Yes
WebLogic Server domain			
	oam_bd (Cluster name)	Yes	No
	AdminServer	Yes	Yes
	oam_server	Yes	Yes
Identity and Access			
	OAM (Oracle Access Manager Cluster)	No	No
	oam_server (Oracle Access Manager Server)	Yes	Yes

Figure 16–11 shows the Log Levels configuration page Fusion Middleware Control.

Figure 16–11 Log Level Configuration Tab



This page allows you to configure the log level for both persistent loggers and active runtime loggers. Persistent loggers are loggers that are saved in a configuration file and become active when the component is started. The log levels for these loggers are persisted across component restarts. Runtime loggers are automatically created during runtime and become active when a particular feature area is exercised. For example, oracle.j2ee.ejb.deployment.Logger is a runtime logger that becomes active when an EJB module is deployed. Log levels for runtime loggers are not persisted across component restarts.

View: Runtime Loggers

Search: All Categories

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
⊟ oracle.oam	WARNING:1 (WARNING)	odl-handler	
oracle.oam.admin.foundation.configurati	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.agent-default	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.audit	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.binding	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.commonutil	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.config	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.controller	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.default	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.diagnostic	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.engine.authn	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.engine.authz	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.engine.policy	WARNING:1 (WARNING) [Inherit	odl-handler	
oracle.oam.engine.session	WARNING:1 (WARNING) [Inherit	odl-handler	

□ Persist log level state across component restarts

The Log Levels tab on the Log Configuration page allows you to configure the log level for both persistent loggers and active runtime loggers:

- Persistent loggers are saved in a configuration file and become active when the component is started.

  The log levels for these loggers are persisted across component restarts.

- Runtime loggers are automatically created during runtime and become active when a particular feature area is exercised.

For example, oracle.j2ee.ejb.deployment.Logger is a runtime logger that becomes active when an EJB module is deployed. Log levels for runtime loggers are not persisted across component restarts.

Table 16–6 explains the configuration status and options for log levels.

Table 16–6 Log Levels Tab on Log Configuration Page

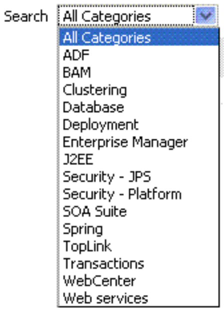
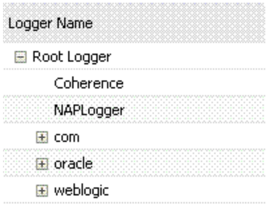
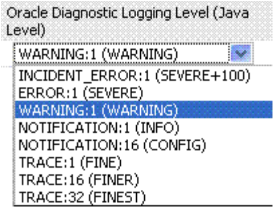
Element	Description
Apply	Submits and applies log level configuration changes, which take affect immediately.
Revert	Restores the target's previous log level configuration, which take affect immediately.
View	Use this list to view runtime loggers or loggers with a persistent log level state. ▪ Runtime Loggers ▪ Loggers with Persistent Log Level State
Search	Use this list to specify the categories you would like to search.  Search All Categories All Categories ADF BAM Clustering Database Deployment Enterprise Manager J2EE Security - JPS Security - Platform SOA Suite Spring TopLink Transactions WebCenter Web services
Table	
Logger Name	The name of the loggers found during the search. You can expand names in the list to see any loggers beneath the top node.  Logger Name Root Logger Coherence NAPLogger com oracle weblogic

Table 16–6 (Cont.) Log Levels Tab on Log Configuration Page

Element	Description
Oracle Diagnostic Logging Level (Java Level)	Choose the logging level for the corresponding logger; c.  Oracle Diagnostic Logging Level (Java Level) WARNING:1 (WARNING) INCIDENT_ERROR:1 (SEVERE+100) ERROR:1 (SEVERE) WARNING:1 (WARNING) NOTIFICATION:1 (INFO) NOTIFICATION:16 (CONFIG) TRACE:1 (FINE) TRACE:16 (FINER) TRACE:32 (FINEST) Click Apply and review confirmation messages displayed in a pop-up window: Updating log levels Updating the log levels of runtime loggers The log levels of runtime loggers have been updated successfully The log levels have been updated successfully
Log File	Clicking a name in the Log File column displays the Log Files page, which you can use to create and edit the file where log messages are logged, the format of the log messages, rotation policies, and other logging parameters. See Also: "Managing OAM Log File Configuration from Fusion Middleware Control" on page 16-20.
Persistent Log Level State	Identifies the persistent state for this specific logger, which is set when you create or edit the value using the Log Files tab.

16.6.2 Setting OAM Log Levels Dynamically Using Fusion Middleware Control

Fusion Middleware Control administrators can use the following procedure to set the log level for OAM 11g components dynamically.

See Also: "About Dynamic Log Level Changes" on page 16-16

Note: Alternatively, administrators can set OAM logger levels using custom WebLogic Scripting Tool (WLST) commands, as described in Chapter 13.

To configure logging levels dynamically in Fusion Middleware Control

1. Log in as described in "Logging In To Fusion Middleware Control" on page 16-3.
2. Expand the desired node, and select a target. For example: Identity and Access.

   Identity and Access
     oam_server

3. From the OAM context menu, select Logs and then choose Log Configuration.
4. Click the Log Levels tab, if needed, to display the configuration page (Table 16–6).
5. From the View list, choose the loggers to display. For example: **Runtime Loggers**.
6. From the Search list, choose the desired category.
7. In the table, expand nodes to reveal information as needed.
8. In the table, choose new log levels, as needed, and then click Apply (or Revert).

9. Proceed to "Managing OAM Log File Configuration from Fusion Middleware Control"

16.7 Managing OAM Log File Configuration from Fusion Middleware Control

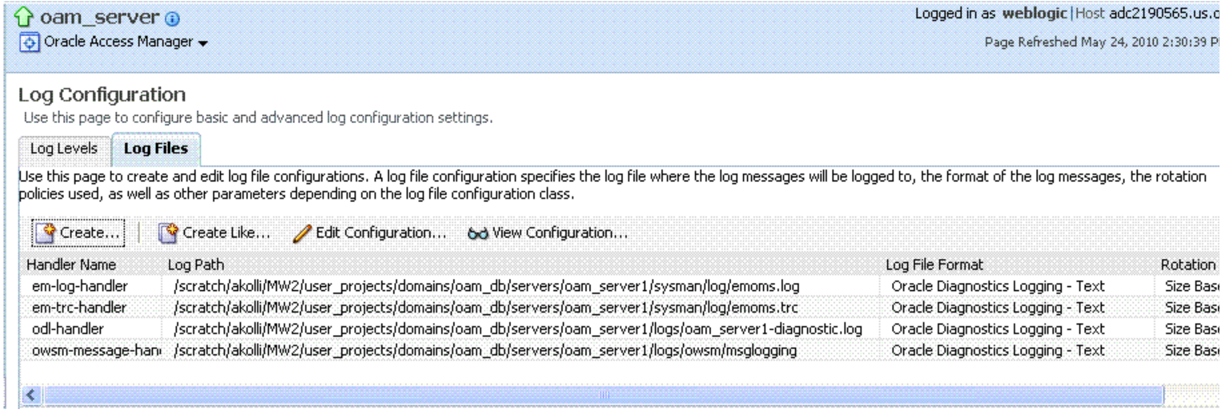
This section provides the following information:

- About Log File Configuration
- Managing OAM Log File Configuration by Using Fusion Middleware Control

16.7.1 About Log File Configuration

Figure 16–6 shows the Log Files Configuration. Use this page to create and edit where the log messages will be logged to, the format of the log messages, the rotation policies used, as well as other parameters depending on the log file configuration class.

Figure 16–12 Log Files Configuration Page



Log Configuration

Use this page to configure basic and advanced log configuration settings.

Use this page to create and edit log file configurations. A log file configuration specifies the log file where the log messages will be logged to, the format of the log messages, the rotation policies used, as well as other parameters depending on the log file configuration class.

Handler Name	Log Path	Log File Format	Rotation
em-log-handler	/scratch/akolli/MW2/user_projects/domains/oam_db/servers/oam_server1/sysman/log/emoms.log	Oracle Diagnostics Logging - Text	Size Bas
em-trc-handler	/scratch/akolli/MW2/user_projects/domains/oam_db/servers/oam_server1/sysman/log/emoms.trc	Oracle Diagnostics Logging - Text	Size Bas
odl-handler	/scratch/akolli/MW2/user_projects/domains/oam_db/servers/oam_server1/logs/oam_server1-diagnostic.log	Oracle Diagnostics Logging - Text	Size Bas
owsm-message-han	/scratch/akolli/MW2/user_projects/domains/oam_db/servers/oam_server1/logs/owsm/msglogging	Oracle Diagnostics Logging - Text	Size Bas

Table 16–7 describes the OAM log files configuration parameters.

Table 16–7 Log Files Elements

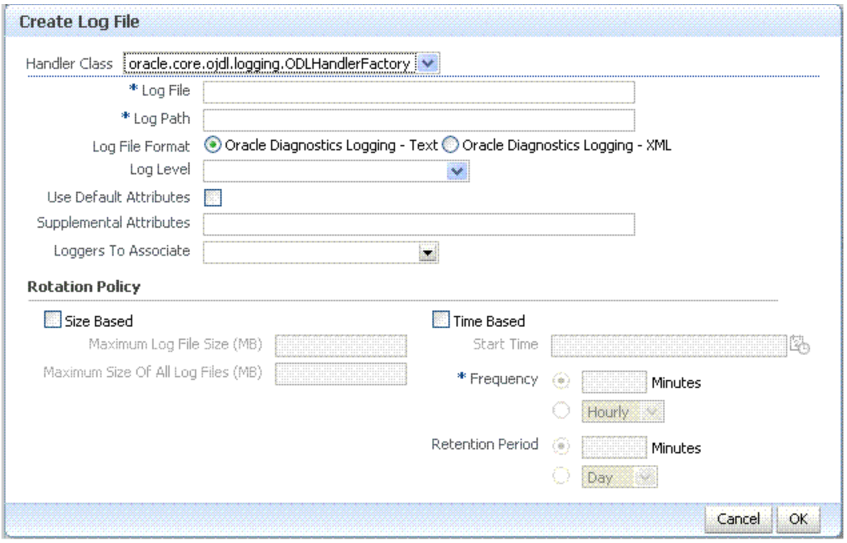
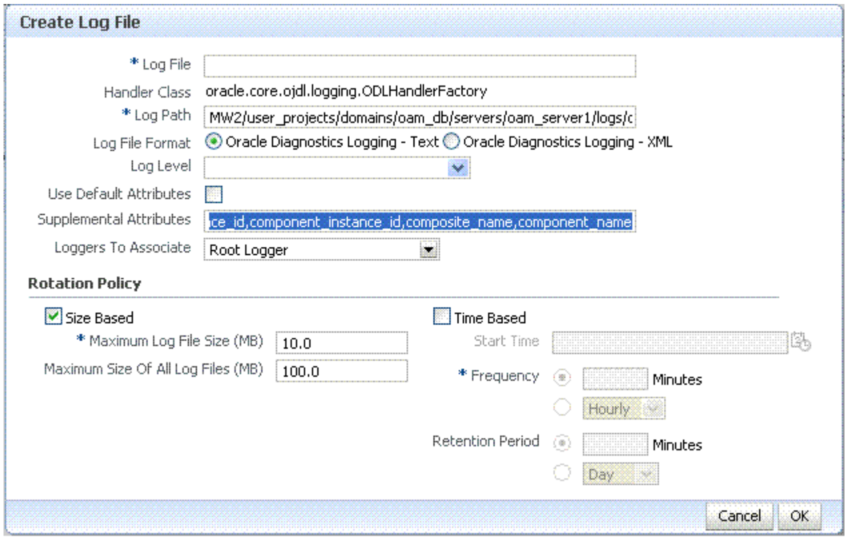
Element	Description
Create	Click this button to display the fresh form to create a new file for logged messages. Notes: - Log File is the name of the log handler (odl-handler for OAM) - Log Path points to the logging output file in your environment *DOMAIN_HOME*/servers/*SERVER-NAME*/logs/*SERVER-NAME*-diagnostics.log - The output logging file in your environment can have a unique file name.  Create Log File Handler Class: oracle.core.ojdl.logging.ODLHandlerFactory * Log File: * Log Path: Log File Format: ● Oracle Diagnostics Logging - Text ○ Oracle Diagnostics Logging - XML Log Level: Use Default Attributes: □ Supplemental Attributes: Loggers To Associate: Rotation Policy □ Size Based Maximum Log File Size (MB): Maximum Size Of All Log Files (MB): □ Time Based Start Time: * Frequency: ● Minutes ○ Hourly Retention Period: ● Minutes ○ Day Cancel OK
Create Like	Click this button to display a partially filled-in form to create a new file for logged messages.  Create Log File * Log File: Handler Class: oracle.core.ojdl.logging.ODLHandlerFactory * Log Path: MW2/user_projects/domains/oam_db/servers/oam_server1/logs/c Log File Format: ● Oracle Diagnostics Logging - Text ○ Oracle Diagnostics Logging - XML Log Level: Use Default Attributes: □ Supplemental Attributes: ce_id,component_instance_id,composite_name,component_name Loggers To Associate: Root Logger Rotation Policy ☑ Size Based * Maximum Log File Size (MB): 10.0 Maximum Size Of All Log Files (MB): 100.0 □ Time Based Start Time: * Frequency: ● Minutes ○ Hourly Retention Period: ● Minutes ○ Day Cancel OK
Edit Configuration	Click this button to display and edit the selected log file configuration.
View Configuration	Click this button to view a read-only description of the selected log file configuration.
Table	The information in this table is based on log file configuration parameters in this table.
Handler Name	The Log File name assigned during log file creation.

Table 16–7 (Cont.) Log Files Elements

Element	Description
Log Path	The file system directory path assigned during log file creation.
Log File Format	The Log File format assigned during log file creation.
Rotation Policy	The rotation policy selected during log file creation.

16.7.2 Managing OAM Log File Configuration by Using Fusion Middleware Control

Fusion Middleware Control administrators can use the following procedure to create an OAM log file, edit the configuration, or view a read-only version of the log file configuration.

> **See Also:** "About Log File Configuration" on page 16-20

To configure logging for OAM in Fusion Middleware Control

1. Log in as described in "Logging In To Fusion Middleware Control" on page 16-3.
2. Expand the desired node, and select a target. For example: Identity and Access.

   Identity and Access
   oam_server

3. From the OAM context menu, select Logs and then choose Log Configuration.
4. Click the Log Files tab if needed to display the configuration page (Table 16–7).
5. **Create a Log File**:
   a. Click the Create button to display a fresh Create Log File form.
   b. Enter a name and file system path for this log file. For example:

      Log File ***oam-odl-handler***

      Log Path domains***/oam_db/servers/oam-server1/log/oam.log***

   c. Click the desired Log File Format. For example: **... Text**
   d. Set the logging attributes. For example:

      Use Default Attributes ***X***

      Supplemental Attributes

   e. Associate a Logger. For example: **Root Logger**
   f. Specify the Rotation Policy. For example: **Size Based**

      Maximum Log File Size (MB) ***10.0***

      Maximum Size of All Log File Size (MB) ***1000.0***

   g. Click OK to submit the configuration.
6. **Create Like**:
   a. From the Log Files tab, click the name of an existing log file.
   b. Click the Create Like button.
   c. On the Create Log File form, enter a new:

      Log File name

      Log Level

Attributes

d. Edit any other details as needed, then click OK to submit the configuration.

7. **Edit Configuration**:

   a. From the Log Files tab, click the name of an existing log file configuration.

   b. Click the Edit Configuration button.

   c. Change configuration details as needed.

   d. Click OK to submit the configuration.

8. **View Configuration**:

   a. From the Log Files tab, click the name of an existing log file configuration.

   b. Click the View Configuration button.

   c. Review the information, then click OK to dismiss the configuration page.

9. Proceed to "Locating and Viewing OAM Log Messages in Fusion Middleware Control".

16.8 Locating and Viewing OAM Log Messages in Fusion Middleware Control

This section includes the following topics:

- About Finding, Viewing, and Exporting Log Messages
- Locating and Viewing Logged OAM Information in Fusion Middleware Control

16.8.1 About Finding, Viewing, and Exporting Log Messages

By using context menus in Fusion Middleware Control, administrators can locate, view, and export key log information for:

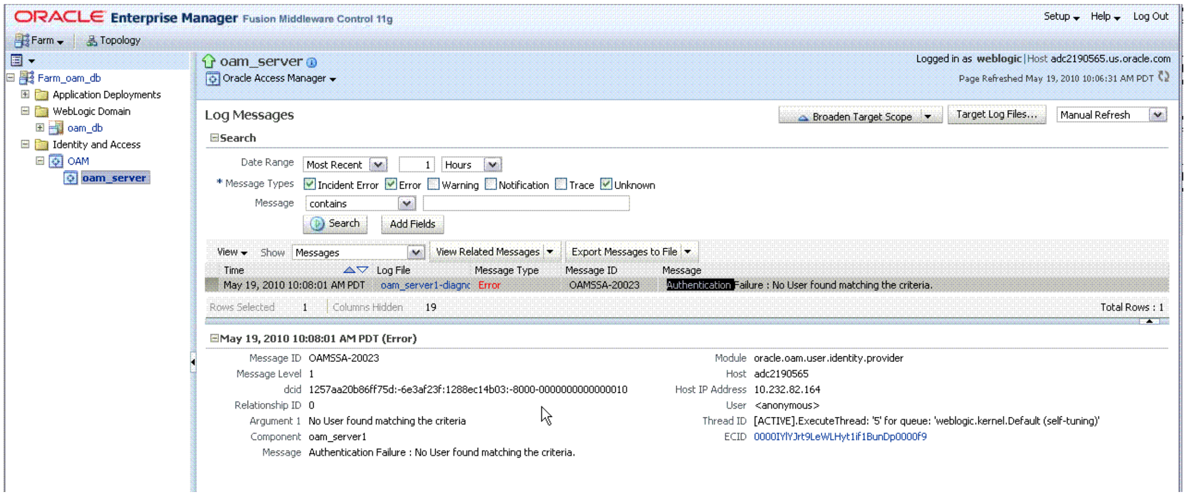
- Application Deployment targets, including the WebLogic (and OAM) AdminServer and the OAM SSO logout pages on both AdminServer and OAM Servers
- WebLogic Server domain targets, including the OAM Farm, AdminServer, and OAM Servers
- Identity and Access targets, including the OAM Farm and OAM Servers

Using log files to troubleshoot common problems requires that you:

- Get familiar with the Oracle Diagnostic Logging (ODL) format used by Oracle Fusion Middleware components, as described in the *Oracle Fusion Middleware Application Security Guide*
- Configure log files to collect the appropriate level of information
- Search, view and export key log information in the farm
- Correlate messages in log files across components

Figure 16–13 shows the OAM 11g Log Messages page in Fusion Middleware Control.

Figure 16–13   Typical Log Messages Page in Fusion Middleware Control



Logged in as **weblogic** | Host adc2190565.us.oracle.com

Page Refreshed May 19, 2010 10:06:31 AM PDT

Log Messages

Search

Date Range: Most Recent, 1, Hours

* Message Types: ☑ Incident Error ☑ Error □ Warning □ Notification □ Trace ☑ Unknown

Message: contains

Show: Messages

Time	Log File	Message Type	Message ID	Message
May 19, 2010 10:08:01 AM PDT	oam_server1-diagnc	Error	OAMSSA-20023	Authentication Failure : No User found matching the criteria.

Rows Selected 1 | Columns Hidden 19 | Total Rows : 1

May 19, 2010 10:08:01 AM PDT (Error)

Message ID	OAMSSA-20023	Module	oracle.oam.user.identity.provider
Message Level	1	Host	adc2190565
dcid	1257aa20b86ff75d:-6e3af23f:1288ec14b03:-8000-0000000000000010	Host IP Address	10.232.82.164
Relationship ID	0	User	<anonymous>
Argument 1	No User found matching the criteria	Thread ID	[ACTIVE].ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'
Component	oam_server1	ECID	0000IYlYJrt9LeWLHyt1if1BunDp0000f9
Message	Authentication Failure : No User found matching the criteria.		

Table 16–8 describes elements on the Log Messages page in Fusion Middleware Control.

Table 16–8   OAM Log Message Search Controls in Fusion Middleware Control

Element	Description
Broaden Target Scope	Select items on this list to expand (or narrow) the targets that are used in this search: ■ Oracle WebLogic Server domain ■ Oracle Access Manager Cluster ■ Oracle WebLogic Server ■ Oracle Fusion Middleware Farm
Target Log Files...	Displays a list of all log files for the target scope from which you can select a specific log file to view or download.
Refresh Options	Select an item from this list to specify the refresh method: ■ Manual Refresh ■ 30 Second Refresh ■ 1 Minute Refresh
Search Options	
Date Range	The period during which the desired set of messages was logged: ■ Most Recent Minutes Hours Days ■ Time interval Date Range Start Date End Date
Message Types	Check all message types that apply for this search: ■ Incident Error ■ Error ■ Warning ■ Notification ■ Trace ■ Unknown

***Table 16–8 (Cont.) OAM Log Message Search Controls in* Fusion Middleware Control**

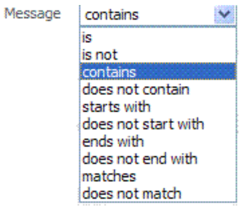
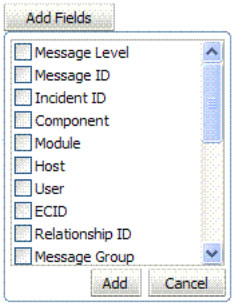
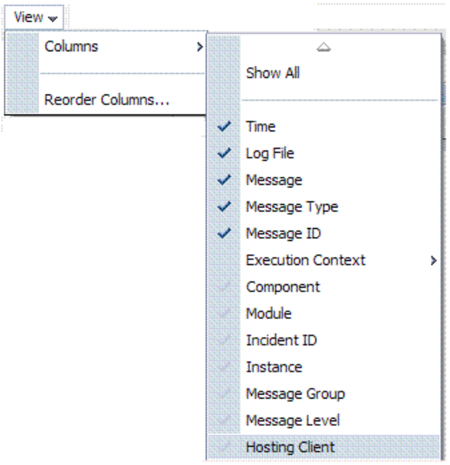
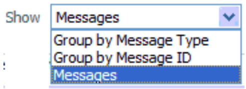
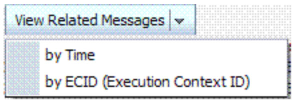
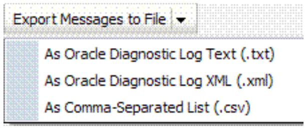
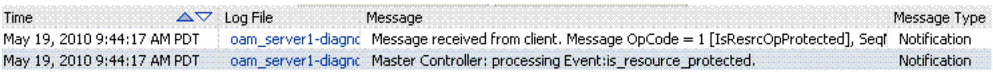
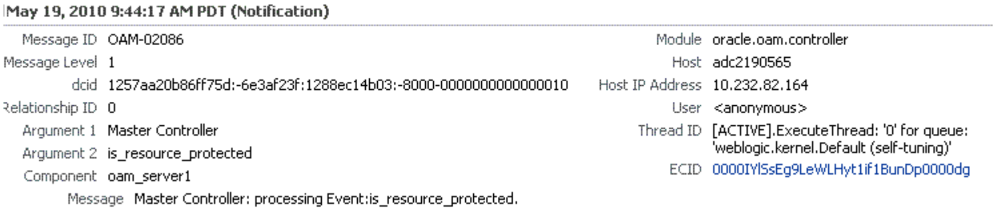
Element	Description
Message	Choose an identifier from this list and add a value in the blank field beside it to refine your search criteria:  Message contains is is not contains does not contain starts with does not start with ends with does not end with matches does not match
Add Fields	Click this button to display a list of additional search criteria you can include.  Add Fields □ Message Level □ Message ID □ Incident ID □ Component □ Module □ Host □ User □ ECID □ Relationship ID □ Message Group Add Cancel
Search	Click this button to initiate a search using the specified criteria.
Viewing Options	
View	Choose items from this menu to view or reorder columns in the search results table:  View Columns Reorder Columns... Show All ✓ Time ✓ Log File ✓ Message ✓ Message Type ✓ Message ID Execution Context Component Module Incident ID Instance Message Group Message Level Hosting Client
Show	Select the entity to view:  Show Messages Group by Message Type Group by Message ID Messages

Table 16–8 (Cont.) OAM Log Message Search Controls in Fusion Middleware Control

Element	Description
View Related Messages	This menu is available when at least one message is listed in the search results.  View Related Messages by Time by ECID (Execution Context ID)
Export Messages to a File	A menu of viewing commands that are available when at least one message is listed in the search results. You can choose from the following commands:  Export Messages to File As Oracle Diagnostic Log Text (.txt) As Oracle Diagnostic Log XML (.xml) As Comma-Separated List (.csv)
Results Table Columns	These are based on selections in the View menu on the Log Messages page.  Time Log File Message Message Type May 19, 2010 9:44:17 AM PDT oam_server1-diagnc Message received from client. Message OpCode = 1 [IsResrcOpProtected], Seqr Notification May 19, 2010 9:44:17 AM PDT oam_server1-diagnc Master Controller: processing Event:is_resource_protected. Notification
Message Area	Displays details for the selected message in the search results table.  May 19, 2010 9:44:17 AM PDT (Notification) Message ID OAM-02086 Message Level 1 dcid 1257aa20b86ff75d:-6e3af23f:1288ec14b03:-8000-0000000000000010 Relationship ID 0 Argument 1 Master Controller Argument 2 is_resource_protected Component oam_server1 Module oracle.oam.controller Host adc2190565 Host IP Address 10.232.82.164 User <anonymous> Thread ID [ACTIVE].ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)' ECID 0000IYlSsEg9LeWLHyt1if1BunDp0000dg Message Master Controller: processing Event:is_resource_protected.

16.8.2 Locating and Viewing Logged OAM Information in Fusion Middleware Control

Fusion Middleware Control administrators can use the following procedure to view and download log messages for the target. This procedure explains how to search for messages, view messages (or view related messages), view all messages in a single log file, and export or download messages.

> **See Also:** "About Finding, Viewing, and Exporting Log Messages" on page 16-23

To view OAM Server log messages within Fusion Middleware Control

1. Log in as described in "Logging In To Fusion Middleware Control" on page 16-3.
2. Expand the desired node and select a target. For example: Identity and Access.

   Identity and Access
   oam_server

3. From the OAM context menu, select Logs and then choose View Log Messages.
4. **Search** (Table 16–8):
   a. Specify the Date Range.

   b. Check all Message Types to be included in your search.

   c. Define Message content options.

   d. Add Fields: Enter details to further refine message content.

   e. Click Search to display a list of messages that fit your search criteria.

5. **View Messages:** Click one or more messages to view from the table of search results to review messages.

6. **View Related**: Use one of the following methods to organize the table of search results.

   a. By **Time**: From the View Related menu, select **by Time**.

   b. By **ECID**: Click ECID in the message on the screen (or, from the View Related menu, select **by ECID Execution Context ID**).

   c. From the Scope menu, select a time period.

7. **Log File**: From the table of search results, click the name in the Log File column to view all messages in the file.

8. **Export Messages**

   a. Select one or more messages in the search results table.

   b. From the **Export Messages** menu, choose the desired export format. For example: **As Oracle Diagnostic Log (.txt)**.

   c. In the dialog box, click **Open with** and then choose the desired program.

   d. From the open program, save the file to a new path.

9. **Download**

   a. Select one or more messages in the search results table.

   b. Click the Download button.

   c. In the dialog box, click **Open with** and then choose the desired program.

   d. From the open program, save the file to a new path.

10. **Testing**:

   a. Using the Access Tester, enter an invalid user name and try to authenticate (see Chapter 10).

   b. In Fusion Middleware Control, go to the log viewer and review the error.

   c. Using the Access Tester, enter an invalid password and try to authenticate.

   d. In the Fusion Middleware Control log viewer, check the error and then view all related log messages.

   e. Repeat this test using different log levels, as described in "Managing Log Level Changes in Fusion Middleware Control" on page 16-16.

16.9 Displaying OAM MBeans in Fusion Middleware Control

A Java object is a unit of code that runs the computer. Each object is an instance of a particular class or subclass that rely on the class's methods or procedures or data variables. Within the Java programming language, a Java object that represents a manageable resource (application, service, component, or device) is known as an MBean (managed bean).

Fusion Middleware Control enables you to:

- View information on key MBean Attributes and Operations
- Invoke methods

This section provides the following topics:

- About the System MBean Browser
- Viewing, Editing, and Invoking OAM 11g Mbeans
-

16.9.1 About the System MBean Browser

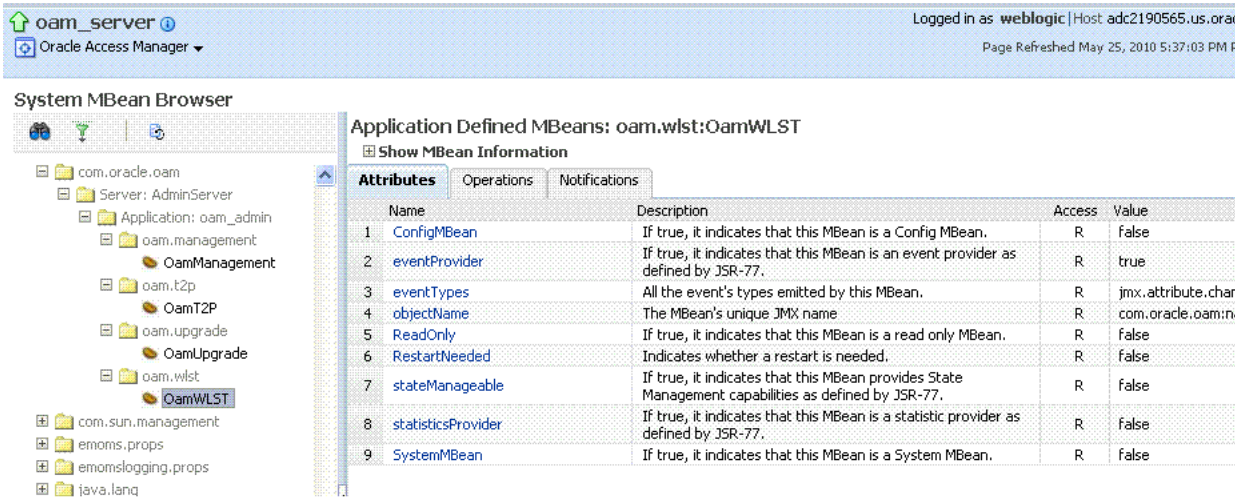
The Fusion Middleware Control System Mbean Browser can be used to view the items outlined in Table 16–9.

Table 16–9 System MBean Browser

Node	Target	System Mbean Browser
Application Deployment		
Internal Applications	...AdminServer	Yes
	oamsso_logout(11.1.1.3.0) AdminServer	Yes
	oamsso_logout(11.1.1.3.0) oam_server	Yes
WebLogic Server domain		
	oam_bd (Cluster name)	Yes
	AdminServer	Yes
	oam_server	Yes
Identity and Access		
	OAM (Oracle Access Manager Cluster)	No
	oam_server (Oracle Access Manager Server)	Yes

Figure 16–14 Shows the System MBean Browser page.

Figure 16–14 System MBean Browser and Attributes Tab



	Name	Description	Access	Value
1	ConfigMBean	If true, it indicates that this MBean is a Config MBean.	R	false
2	eventProvider	If true, it indicates that this MBean is an event provider as defined by JSR-77.	R	true
3	eventTypes	All the event's types emitted by this MBean.	R	jmx.attribute.char
4	objectName	The MBean's unique JMX name	R	com.oracle.oam:n
5	ReadOnly	If true, it indicates that this MBean is a read only MBean.	R	false
6	RestartNeeded	Indicates whether a restart is needed.	R	false
7	stateManageable	If true, it indicates that this MBean provides State Management capabilities as defined by JSR-77.	R	false
8	statisticsProvider	If true, it indicates that this MBean is a statistic provider as defined by JSR-77.	R	false
9	SystemMBean	If true, it indicates that this MBean is a System MBean.	R	false

Table 16–10 describes the System MBean Browser and associated tabs.

Table 16–10 System MBean Browser

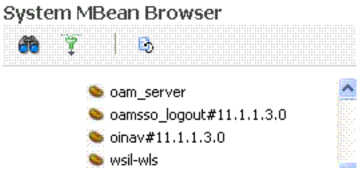
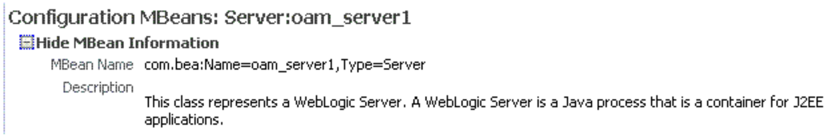
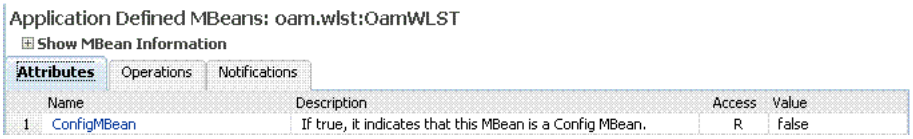
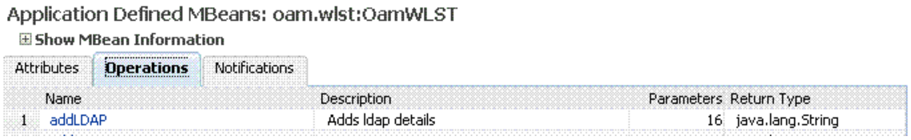
System MBean Browser	
System MBean Browser	Expand items in this section to display Mbeans for the selected target.  System MBean Browser oam_server oamsso_logout#11.1.1.3.0 oinav#11.1.1.3.0 wsil-wls
MBean Information	Details for Attributes and Operations related to the MBean for the selected target are displayed on the right.  Configuration MBeans: Server:oam_server1 Hide MBean Information MBean Name com.bea:Name=oam_server1,Type=Server Description This class represents a WebLogic Server. A WebLogic Server is a Java process that is a container for J2EE applications.
Attributes	This tab describes MBean attributes for the selected target.  Application Defined MBeans: oam.wlst:OamWLST Show MBean Information Attributes Operations Notifications Name Description Access Value 1 ConfigMBean If true, it indicates that this MBean is a Config MBean. R false

Table 16–10 (Cont.) System MBean Browser

System MBean Browser	
Operations	This tab describes MBean operations for the selected target.  Application Defined MBeans: oam.wlst:OamWLST ⊞ Show MBean Information Attributes | Operations | Notifications Name | Description | Parameters | Return Type 1 | addLDAP | Adds ldap details | 16 | java.lang.String
Notifications	This tab lists any notifications resulting from the invocation of an MBean.
Name	Clicking a name on either tab displays a full description of related MBeans.
Apply	Submits and applies the selected MBean attribute value.
Revert	Restores previous MBean attribute values following a change (and before clicking Apply).
Return	Returns you to the MBean Information page.
Invoke	Invokes the selected MBean and value.

16.9.2 Viewing, Editing, and Invoking OAM 11g Mbeans

Fusion Middleware Control administrators can use the following procedure to view OAM 11g MBeans. Additionally, you can apply values (or revert the change) and invoke OAM 11g MBeans.

To view, edit, or invoke OAM 11g MBeans

1. Log in as described in "Logging In To Fusion Middleware Control" on page 16-3.
2. Expand the desired node and select a target. For example:

   Identity and Access
     oam_server

3. From the OAM context menu, select System MBean Browser.
4. **System MBean Browser**: Expand classes and select a target to display related attributes and operations.
5. **Manage MBean Attributes**:
   a. Click the Attributes tab.
   b. Review the name and description of MBean attributes for the selected target.
   c. Edit values for one or more attributes and click Apply to submit changes (or click Revert to cancel changes).

      Alternatively: Click a Name in the Attributes table to display a full description and the value; change the value and click Apply (or click Revert to cancel the change).
6. **Manage MBean Operations**:
   a. Click the Operations tab.
   b. Review the name, description, number of parameters, and the return type for each MBean operation for the selected target
   c. Click a name in the Operations table to display the parameters and related name, description, type, and value.

d. Edit values for the operation and click Apply to submit changes (or click Revert to cancel changes).

e. Click Invoke to invoke the MBean and review the message that appears. <<<then what?>>>

7.

16.10 Displaying Farm Routing Topology in Fusion Middleware Control

Fusion Middleware Control enables you to view a graphical representation of the Oracle Access Manager routing topology.

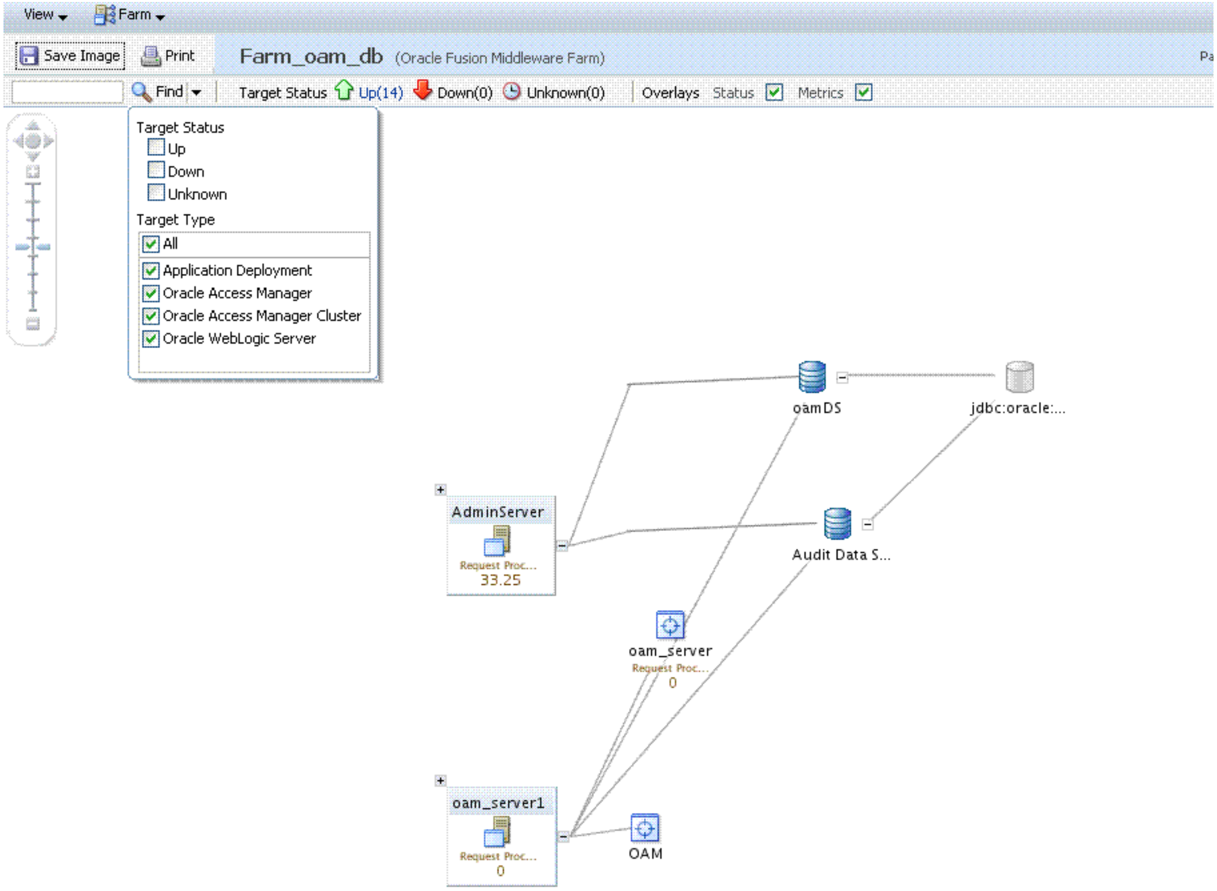
This section provides the following topics:

- About the Routing Topology
- Viewing the OAM Routing Topology using Fusion Middleware Control

16.10.1 About the Routing Topology

Figure 16–15 shows the Farm routing topology page in Fusion Middleware Control.

Figure 16–15 Routing Topology with Context Menu

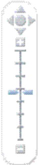
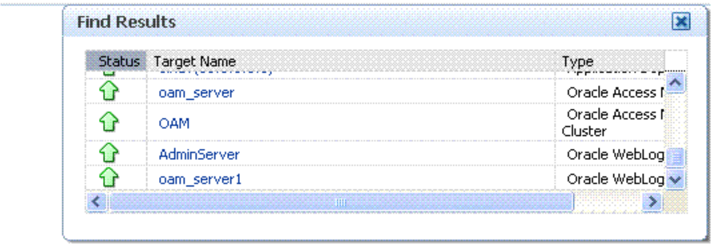
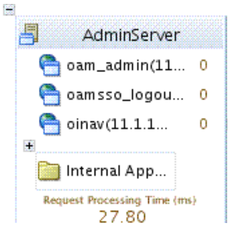
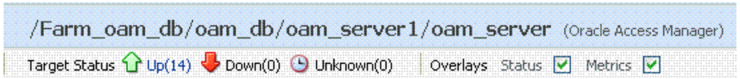


Target Status
☐ Up
☐ Down
☐ Unknown

Target Type
☑ All
☑ Application Deployment
☑ Oracle Access Manager
☑ Oracle Access Manager Cluster
☑ Oracle WebLogic Server

Table 16–11 describes the status and controls on the Farm topology page.

Table 16–11 Farm Topology

Element	Description
Save Image	Saves the image.
Print	Prints the image.
	Scales the image.
Find	Enter a value or simply click Find to display results.  Find Results Status | Target Name | Type ⇧ | oam_server | Oracle Access I ⇧ | OAM | Oracle Access I Cluster ⇧ | AdminServer | Oracle WebLog ⇧ | oam_server1 | Oracle WebLog
+	Expands the instance on the topographical view to provide more information.  AdminServer oam_admin(11... 0 oamsso_logou... 0 oinav(11.1.1... 0 Internal App... Request Processing Time (ms) 27.80
Status Bar	Displays the full farm name and targets within the farm., as well as the up and down status. You can choose to overlay the status and metrics on individual instances in the topology view.  /Farm_oam_db/oam_db/oam_server1/oam_server (Oracle Access Manager) Target Status ⇧ Up(14) Down(0) Unknown(0) Overlays Status ☑ Metrics ☑

16.10.2 Viewing the OAM Routing Topology using Fusion Middleware Control

Fusion Middleware Control Administrators can use the following procedure to view the routing topology of the farm that includes OAM 11g.

See Also: "About the Routing Topology"

To view Farm routing topology

1. Log in as described in "Logging In To Fusion Middleware Control" on page 16-3.
2. Select the Farm in the navigation tree.
3. Click Topology above the navigation tree.

4. In the Topology Browser window, click the name of the farm and click OK.
5. Use the scaling tool to shrink or grow the image.
6. Expand instances in the topology to display details about each one.
7. Use the Overlay options to add status and metrics information to the instances.
8. Use the Find option to locate specific information (Table 16–11).
9. Click Print or Save, as needed.
10.

Part VII

Using OAM 10g WebGates with OAM 11g

When your enterprise includes Web server types other than Oracle HTTP Server, you can install OAM 10g WebGates to use with OAM 11g

Part VII contains the following chapters:

- Chapter 17, "Managing OAM 10g WebGates with OAM 11g"
- Chapter 18, "Configuring Apache, OHS, IHS for 10g WebGates"
- Chapter 19, "Configuring the IIS Web Server for 10g WebGates"
- Chapter 20, "Configuring the ISA Server for 10g WebGates"
- Chapter 21, "Configuring Lotus Domino Web Servers for 10g WebGates"

17

Managing OAM 10g WebGates with OAM 11g

The Oracle Fusion Middleware Installation Guide for Oracle Identity Management describes initial deployment of Oracle Access Manager 11g with the Oracle HTTP Server. However, when your enterprise includes Web server types other than Oracle HTTP Server you might want to use existing OAM 10g WebGates or install fresh OAM 10g WebGates for use with OAM 11g. Also, you might want to switch from using the pre-registered IDM Domain Agent to using a 10g WebGate to protect Oracle Identity Management Consoles.

The following sections describe how to install fresh instances of OAM 10g WebGates for use with OAM 11g:

- Prerequisites
- Introduction to OAM 10g Agents for OAM 11g
- Provisioning a 10g WebGate with OAM 11g
- Locating and Installing the Latest OAM 10g WebGate for OAM 11g
- Configuring Centralized Logout for 10g WebGate with OAM 11g
- Replacing the IDM Domain Agent with an OAM 10g WebGate
- Deploying Applications in a WebLogic Container
- Removing a 10g WebGate from the OAM 11g Deployment

> **Note:** **Existing OSSO 10g Customers**: If OSSO is already in place as the enterprise solution for your existing Oracle deployment, Oracle Fusion Middleware continues to support this as a solution. Additionally, you can provision existing OSSO 10g mod_osso modules as agents for OAM 11g as described in Chapter 5.

17.1 Prerequisites

Review the latest certification matrix from Oracle Technology Network to locate the latest WebGates for your deployment:

```
http://www.oracle.com/technology/products/id_mgmt/coreid_
acc/pdf/oracle_access_manager_certification_10.1.4_r3_matrix.xls
```

Ensure that your OAM 11g Administration Console is running and get familiar with:

- Introduction to Policy Enforcement Agents on page 5-1

- Introduction to OAM 10g Agents for OAM 11g in this chapter

17.2 Introduction to OAM 10g Agents for OAM 11g

This section provides the following topics:

- About Replacing the IDM Domain Agent with an OAM 10g WebGate
- About Legacy OAM 10g Deployments and WebGates
- About Installing Fresh OAM 10g WebGates to Use With OAM 11g

17.2.1 About Replacing the IDM Domain Agent with an OAM 10g WebGate

As described in Chapter 5, the IDM Domain Agent is a Java agent filter that is pre-registered with OAM 11g out of the box. This agent provides SSO protection for Oracle Identity Management Consoles and resources in the Identity Management domain.

The following overview outlines the tasks that must be performed if you choose to move from the IDM Domain agent to an OAM 10g WebGate to protect Oracle Identity Management Consoles and resources in the Identity Management domain.

Details for each of the tasks in the following overview are located in "Replacing the IDM Domain Agent with an OAM 10g WebGate" on page 17-15.

17.2.2 About Legacy OAM 10g Deployments and WebGates

Oracle Access Manager 11g Servers support OAM 10g WebGates, which can include:

- Legacy 10g WebGates currently operating with OAM 10g as described in the *Oracle Access Manager Access Administration Guide*.
- Legacy 10g WebGates configured as the Identity Assertion Provider (IAP) for SSO (for applications using WebLogic container-based security with OAM 10g, as described in the *Oracle Fusion Middleware Application Security Guide*).
- Legacy 10g WebGates currently operating with Web Applications coded for Oracle ADF Security and the OPSS SSO Framework as described in Appendix C.

You can register these agents to use Oracle Access Manager 11g SSO using either the OAM 11g Administration Console or the remote registration tool. After registration, OAM 10g WebGates directly communicate with Oracle Access Manager 11g services through a JAVA-based OAM proxy that acts as a bridge.

The following overview outlines the tasks that must be performed to set up an existing OAM 10g WebGate to operate with OAM 11g.

Task overview: Setting up a legacy 10g WebGate to operate with OAM 11g

1. Provisioning a 10g WebGate with OAM 11g
2. Configuring Centralized Logout for 10g WebGate with OAM 11g
3. Optional: Deploying Applications in a WebLogic Container

17.2.3 About Installing Fresh OAM 10g WebGates to Use With OAM 11g

You can install fresh OAM 10g WebGates for use with OAM 11g as described in this chapter. OAM 10g WebGates are available for a number of Web server platforms.

After installation and registration, OAM 10g WebGates directly communicate with Oracle Access Manager 11g services through a JAVA-based OAM proxy that acts as a bridge.

> **Note:** When installing fresh OAM 10g WebGates for OAM 11g, Oracle recommends that you use the latest WebGates. Oracle also recommends that you install multiple WebGates for failover and load balancing.

There are several differences between installing an OAM 10g WebGate to operate in an OAM 11g deployment versus installing the 10g WebGate in an OAM 10g deployment. Table 17–1 outlines these differences.

Table 17–1 Installation Comparison with OAM 10g WebGates

10g WebGates in OAM 11g Deployments	10g WebGates in OAM 10g Deployments
1. Packages: OAM 10g WebGate installation packages are found on media and virtual media that is separate from the core components.	1. Packages: OAM 10g WebGate installation packages are found on media and virtual media that is separate from the core components.
2. Provisioning: Before installation, provision WebGate with OAM 11g as described in "Provisioning a 10g WebGate with OAM 11g" on page 17-4.	2. Provisioning: Before installation, you create a WebGate instance in the Access System Console.
3. Associating with OAM Server: Occurs during WebGate registration (task 2 of this sequence).	3. Associating with AAA: Before installation, you associated the WebGate with an Access Server in the Access System Console.
4. Installing: Install the 10g WebGate in front of the application (or for Fusion Middleware, in front of the WebLogic Server).	4. Installing: Using 10g WebGate packages.
5. Language Packs: 10g WebGate Language Packs are supported with OAM 11g.	5. Language Packs: 10g WebGate Language Packs could be installed during WebGate installation (or later).
6. Web Server Configuration: Copy OAM 11g generated files to the WebGate installation directory path to update the Web server configuration.	6. Web Server Configuration: Automatic during WebGate installation (or manually after WebGate installation).
7. Certificate Installation: Copy files to the WebGate installation directory path.	7. Certificate Installation: You copied files to the WebGate installation directory path.
8. Forms: 10g forms provided with 10g WebGates cannot be used with OAM 11g Servers. Using 10g WebGates with OAM 11g Servers is similar in operation and scope to a resource WebGate (one that redirects in contrast to the Authentication WebGate). With a 10g WebGate and 11g OAM Server, the 10g WebGate always redirects to the OAM 11g credential collector which acts like the authenticating WebGate.	8. Forms: Were provided for use in 10g deployments.
9. Single Log Out: Configure using information in Chapter 11, "Configuring Centralized Logout for OAM 11g".	9. Centralized Log Out for OAM 10g.
10. Multi-Domain Support: Does not apply with OAM 11g.	10. Multi-Domain Support: Could be configured for OAM 10g.

The following overview lists the topics in this chapter that describe OAM 10g WebGate installation and registration tasks for OAM 11g in detail. You must complete all procedures for successful operation with OAM 11g.

Task overview: Provisioning and installing a 10g WebGate for OAM 11g

1. Provisioning a 10g WebGate with OAM 11g
2. Locating and Downloading 10g WebGates for Use with OAM 11g
3. Configuring Centralized Logout for 10g WebGate with OAM 11g
4. Optional: Deploying Applications in a WebLogic Container

17.3 Provisioning a 10g WebGate with OAM 11g

Whether you have a legacy OAM 10g WebGate or you are installing a fresh 10g WebGate instance to use with Oracle Access Manager 11g, you must provision WebGate to use OAM 11g authentication and authorization services.

You can use either the OAM 11g Administration Console or the remote registration tool to perform this task. The remote registration tool enables you to specify all WebGate parameters before registration using a template.

The following procedure walks through provisioning using the remote registration tool, in-band mode. In this example, OAMRequest_short.xml is used as a template to create an agent named *my10g-agent1*, protecting */.../**, and declaring a public resource, /public/index.html. Your values will be different.

See Also:

- "Replacing the IDM Domain Agent with an OAM 10g WebGate" on page 17-15 if needed
- Chapter 6 for more information about the remote registration tool, processing, and request files
- Chapter 5 if you prefer using the OAM Administration Console

To provision a 10g WebGate for OAM 11g

1. Acquire the remote registration tool and set up the script for your environment. For example:

   a. Locate RREG.tar.gz file in the following path:

   ```
   WLS_home/Middleware/domain_home/oam/server/rreg/client/RREG.tar.gz
   ```

   b. Untar RREG.tar.gz file to any suitable location. For example: rreg/bin/oamreg.

   c. In the oamreg script, set the following environment variables based on your situation (client side or server side) and information in Table 6–7:

   OAM_REG_HOME = *exploded_dir_for_RREG*.tar/rreg
   JDK_HOME = *Java_location_on_the_computer*

2. Create the registration request:

   a. Locate OAMRequest_short.xml and copy it to a new file. For example:

   ```
   WLS_home/Middleware/domain_home/oam/server/rreg/bin/oamreg/
   ```

   Copy: OAMRequest_short.xml

   To: *my-10g-agent1*.xml

   b. Edit *my-10g-agent1*.xml to include details for your environment. For example:

   ```
   <OAMRegRequest>
       <serverAddress>http://sample.us.oracle.com:7001</serverAddress>
       <hostIdentifier>my-10g</hostIdentifier>
       <agentName>my-10g-agent1</agentName>
       <primaryCookieDomain>.us.example.com</primaryCookieDomain>
       <autoCreatePolicy>false</autoCreatePolicy>
       <logOutUrls><url>/oamsso/logout.html</url></logOutUrls>
   </OAMRegRequest>
   ```

See Also: "Creating the Registration Request" on page 6-19

3. Provision the agent. For example:

   a. Locate the remote registration script.

      Linux: rreg/bin/oamreg.sh

      Windows: rreg\bin\oamreg.bat

   b. From the directory containing the script, execute the script using inband mode. For example:

      $ ./bin/oamreg.sh inband input/*10g-agent1*.xml

      ```
      Welcome to OAM Remote Registration Tool!
      Parameters passed to the registration tool are:
      Mode: inband
      Filename: ...
      ```

   c. When prompted, enter the following information using values for your environment:

      ```
      Enter your agent username: userame
         Username:  userame
      Enter agent password: ********
      Do you want to enter a Webgate password?(y/n)
          n
      iv.Do you want to import an URIs file?(y/n)
          n
      ```

   d. Review the final message to confirm that this was a successful registration:

      ```
      Inband registration process completed successfully! Output artifacts are
      created in the output folder"
      ```

4. Log in to the OAM Console and review the new registration:

   a. From the OAM 11g Console System Configuration tab, navigation tree, expand the following nodes:

      Agents
         OAM Agents
            10g Agents

   b. Double-click the agent's name to display the registration page and review the details.

      If you will install a fresh WebGate for this registration, you must enter the following details during the installation. For example:

      Agent Name—Enter this as the WebGate ID during WebGate installation.

      Access Client Password—Enter this as the WebGate password during WebGate installation. If no password was entered, you will leave the field blank.

      Access Server Host Name—Enter the DNS host name for the primary OAM 11g Server with which this WebGate is registered.

   c. **OAM Proxy Port**—From the OAM Console, System Configuration tab, navigation tree, double click Server Instances and locate the port on which the OAM Proxy is running.

5. Ignore the Obaccessclient.xml file that is created as a result of provisioning for now.
6. Proceed as needed for your environment:
   - Existing WebGate: Configuring Centralized Logout for 10g WebGate with OAM 11g
   - New WebGate: Locating and Installing the Latest OAM 10g WebGate for OAM 11g
   - Replacing the IDM Domain Agent with an OAM 10g WebGate on page 17-15

17.4 Locating and Installing the Latest OAM 10g WebGate for OAM 11g

Use the procedures in this section if you need to install a fresh OAM 10g WebGate for use with OAM 11g. Otherwise, skip this section and proceed to "Configuring Centralized Logout for 10g WebGate with OAM 11g".

Task overview: Installing the WebGate includes

1. Preparing for a Fresh 10g WebGate Installation with OAM 11g
2. Locating and Downloading 10g WebGates for Use with OAM 11g
3. Starting WebGate 10g Installation
4. Specifying a Transport Security Mode
5. Specifying WebGate Configuration Details
6. Requesting or Installing Certificates for Secure Communications
7. Updating the WebGate Web Server Configuration
8. Finishing WebGate Installation
9. Installing Artifacts and Certificates
10. Confirming WebGate Installation

17.4.1 Preparing for a Fresh 10g WebGate Installation with OAM 11g

Table 17–2 outlines the requirements that must be met before starting an OAM 10g WebGate installation.

Table 17–2 Preparing for 10g WebGate Installation with OAM 11g

About the ...	Description
Latest Supported WebGates	Always use the latest supported 10*g* (10.1.4.3) WebGates with OAM 11g. However, if the desired 10*g* (10.1.4.3) WebGate is not provided, use the next latest WebGate (10*g* (10.1.4.2.0). See Also: "Locating and Downloading 10g WebGates for Use with OAM 11g"
Location for installation	Consider: ■ WebGate in front of the application server. ■ Applications using WebLogic Server container-managed security: In front of the WebLogic Application Server in which your application is deployed
User Accounts	The account that is used to install the WebGate is not the account that runs the WebGate: ■ The 10g WebGate should be installed using the same user and group as the Web server. ■ Unix: You can be logged in as root to install the WebGate. The WebGate can be installed using a non-root user if the Web server process runs as a non-root user
Root Level versus Site Level	■ The WebGate can be installed at the root level or the site level. ■ Installing WebGate on multiple virtual sites amounts to only one instance of WebGate.

Table 17–2 (Cont.) Preparing for 10g WebGate Installation with OAM 11g

About the ...	Description
Transport Security Mode	Ensure that at least one OAM Server is configured to use the same mode as the agent to be installed. See Also Appendix E
Computer Level or Virtual Web Server Level	The WebGate can be configured to run at either the computer level or the virtual Web server level. Do not install at both the computer level and the virtual Web server levels.
Oracle HTTP Server Web Server:	The 10g WebGate for Oracle HTTP Server is based on open source Apache. WebGate package names include: ■ OHS (based on Apache v1.3) ■ OHS2 (based on Apache v2) ■ OHS11g (based on Apache v2.2 and is not the subject of this chapter)
Apache Web Servers	Oracle Access Manager 11g provides a single package for components that support Apache with or without SSL enabled: ■ The APACHE2_WebGate supports v2 with or without SSL (and with or without reverse proxy enabled on Solaris and Linux). See also Chapter 18 ■ The APACHE22_WebGate supports v2.2 with or without SSL (and with or without reverse proxy enabled on Solaris and Linux). See also Chapter 18 Note: For SSL-enabled communication, Oracle Access Manager supports Apache with mod_ssl only, not Apache-SSL. mod_ssl is a derivative of, and alternative to, Apache-SSL.
IBM HTTP Server (IHS) v2 Web Servers:	IHS2_WebGate is powered by Apache v2 on IBM-AIX. Oracle Access Manager supports IHS v2 and IHS v2 Reverse Proxy servers with or without SSL enabled. For details, see Chapter 18.

Table 17–2 (Cont.) Preparing for 10g WebGate Installation with OAM 11g

About the ...	Description
Domino Web Servers:	Before you install the OAM 10g WebGate with a Domino Web server, you must have properly installed and set up the Domino Enterprise Server R5. See Also: Chapter 21, "Configuring Lotus Domino Web Servers for 10g WebGates".
IIS Web Servers	Before installing WebGate, ensure that your IIS Web server is *not* in lock down mode. Otherwise things will appear to be working until the server is rebooted and the metabase re-initialized, at which time IIS will disregard activity that occurred after the lock down. If you are using client certificate authentication, before enabling client certificates for the WebGate you must enable SSL on the IIS Web server hosting the WebGate. Setting various permissions for the /access directory is required for IIS WebGates only when you are installing on a file system that supports NTFS. For example, suppose you install the ISAPI WebGate in Simple or Cert mode on a Windows 2000 computer running the FAT32 file system. The last installation panel provides instructions for manually setting various permissions that cannot be set on the FAT32 fleshiest. In this case, these instructions may be ignored. Each IIS Virtual Web server can have it's own WebGate.dll file installed at the virtual level, or can have one WebGate affecting all sites installed at the site level. Either install the WebGate.dll at the site level to control all virtual hosts or install the WebGate.dll for one or all virtual hosts. You may also need to install the postgate.dll file at the computer level. The postgate.dll is located in the \WebGate_install_dir, as described in "Installing the Postgate ISAPI Filter" on page 19-12. If you perform multiple installations, multiple versions of this file may be created which may cause unusual Oracle Access Manager behavior. In this case, you should verify that only one webgate.dll and one postgate.dll exist. **See Also**: Chapter 19, "Configuring the IIS Web Server for 10g WebGates" **Removal**: To fully remove a WebGate and related filters from IIS, you must do more than simply remove the filters from the list in IIS. IIS retains all of its settings in a metabase file. On Windows 2000 and later, this is an XML file that can be modified by hand. There is also a tool available, MetaEdit, to edit the metabase. MetaEdit looks like Regedit and has a consistency checker and a browser/editor. To fully remove a WebGate from IIS, use MetaEdit to edit the metabase.
ISA Proxy Servers	On the ISA proxy server, all ISAPI filters must be installed within the ISA installation directory. They can be anywhere within the ISA installation directory structure: **1.** Before installing the WebGate on the ISA proxy server: Check for general ISAPI filter with ISA instructions on: `http://msdn.microsoft.com/library/default.asp?url=/library/en-us/isa/isaisapi_5cq8.asp` Ensure that the internal and external communication layers are configured and working properly. **2.** During installation you will asked if this is an ISA installation; be sure to: Indicate that this is an ISA proxy server installation, when asked. Specify the ISA installation directory path as the WebGate installation path. Use the automatic Web server update feature to update the ISA proxy server during WebGate installation. **3.** After WebGate installation, locate the file configureISA4webgate.bat, which calls a number of scripts and the process to configure the ISA server filters that must be added programmatically. **See Also**: Chapter 20, "Configuring the ISA Server for 10g WebGates"

17.4.2 Locating and Downloading 10g WebGates for Use with OAM 11g

Use the following procedure to obtain an OAM 10g WebGate, if needed. Be sure to choose the appropriate installation package for your Web server.

To find and download OAM 10g WebGates

1. Review the latest Oracle Access Manager 10g certification information on the Oracle Technology Network at:

   `http://www.oracle.com/technology/products/id_mgmt/coreid_acc/pdf/oracle_access_manager_certification_10.1.4_r3_matrix.xls`

2. Go to Oracle Fusion Middleware 11gR1 Software Downloads at:

   `http://www.oracle.com/technology/software/products/middleware/htdocs/fmw_11_download.html`

3. Click **Accept License Agreement**, at the top of the page.
4. From the **Access Manager WebGates (10.1.4.3.0)** row, click the download link for the desired platform and follow on-screen instructions.
5. Store the WebGate installer in the same directory with any 10g Access System Language Packs you want to install.
6. Proceed to "Starting WebGate 10g Installation".

17.4.3 Starting WebGate 10g Installation

The following procedure walks through the steps, which are the same regardless of Web server type.

Installation options are identified and can be skipped if they do not apply to your environment. During WebGate installation, information is saved at specific points. You can cancel WebGate installation processing if needed. However, if you cancel WebGate installation after being informed that the WebGate is being installed, you must uninstall the component.

> **Note:** On HP-UX and AIX systems, you can direct an installation to a directory with sufficient space using the -is:tempdir path parameter. The path must be an absolute path to a file system with sufficient space.

To start WebGate 10g installation

1. On the computer to host WebGate 10g, log in as a user with Web server Administrator privileges.
2. Stop the Web server instance.
3. Launch the WebGate installer for your preferred platform, installation mode, and Web server. For example:

   GUI Method

   Windows— Oracle_Access_Manager10_1_4_3_0_Win32_*API*_WebGate.exe

   Console Method

   Solaris—./ Oracle_Access_Manager10_1_4_3_0_sparc-s2_*API*_WebGate

   Linux—./ Oracle_Access_Manager10_1_4_3_0_linux_*API*_WebGate

   where *API* refers to the API used by your Web server (ISAPI for IIS Web servers, for example).
4. Dismiss the Welcome screen by clicking Next.
5. Respond with administrator privileges when asked.
6. Specify the installation directory for the WebGate. For example:

   \OracleAccessManager\WebComponent\
7. **Linux or Solaris**: Specify the location of the GCC runtime libraries on this computer.
8. **Language Pack—**Choose a Default Locale and any other Locales to install, then click Next.

9. Record the installation directory name in the preparation worksheet if you haven't already, then click Next to continue.

   The WebGate installation begins, which may take a few seconds. On Windows systems, a screen informs you that the Microsoft Managed Interfaces are being configured.

   The installation process is not yet complete. You are asked to specify a transport security mode. At this point, you cannot go back to restate information.

10. Specify the location where you unzipped the previously downloaded GCC libraries, if needed.

17.4.4 Specifying a Transport Security Mode

Transport security between at least one OAM Server must match.

See Also: Appendix E

To specify a transport security mode

1. Choose Open, Simple, or Cert for the WebGate.
2. Click Next.

   You are now asked to specify WebGate configuration details.

3. Proceed according to your specified transport security mode:
   - **Simple or Certificate Mode**—Go to "Requesting or Installing Certificates for Secure Communications".
   - **Open Mode**—Skip to "Updating the WebGate Web Server Configuration".

17.4.5 Requesting or Installing Certificates for Secure Communications

If your OAM 11g environment uses Open mode transport security, you can skip to "Updating the WebGate Web Server Configuration".

WebGate Certificate Request: Generates the request file (aaa_req.pem), which you must send to a root CA that is trusted by the OAM 11g server. The root CA returns signed certificates, which can then be installed for WebGate.

Requested certificates must be copied to the *WebGate_install_dir*\access\oblix\config directory and then the WebGate Web server should be restarted.

See Also: Appendix E

To request or install certificates for WebGate 10g

1. Indicate whether you are requesting or installing a certificate, then click Next and continue. For example:
   - Requesting a certificate, proceed with step 2.
   - Installing a certificate, skip to step 3.
2. **Request a Certificate**:
   - Enter the requested information, then click Next and issue your request for a certificate to your CA.
   - Record certificate file locations, if these are displayed.

- Click Yes if your certificates are available and continue with step 3. Otherwise, skip to "Updating the WebGate Web Server Configuration".

3. **Install a Certificate During Installation**: Specify the full paths to the following files, then click Next:

   WebGate_install_dir\access\oblix\config

   - cacert.pem the certificate request, signed by the Oracle-provided openSSL Certificate Authority
   - password.xml contains the random global passphrase that was designated during installation, in obfuscated format. This is used to prevent other customers from using the same CA. Oracle Access Manager performs an additional password check during the initial handshake between the OAM Agent and OAM Server.
   - aaa_key.pem contains your private key (generated by openSSL).
   - aaa_cert.pem signed certificates in PEM format.
   - Proceed to "Updating the WebGate Web Server Configuration".

17.4.6 Specifying WebGate Configuration Details

You perform the following task using information provided during WebGate provisioning and registration with OAM 11g.

To provide WebGate configuration details

1. Provide the information requested for the WebGate as specified in the Access System Console.
   - **WebGate ID—**Enter the agent name that you supplied during registration.
   - **WebGate password—**Enter the password supplied during registration, if any. If no password was entered, leave the field blank.
   - **Access Server ID—**Enter the name of the OAM 11g Server with which this WebGate is registered, if desired, or use any name you choose.
   - **Access Server Host Name—**Enter the DNS host name for the OAM 11g Server with which this WebGate is registered
   - **Port number—**Enter the port on which the OAM Proxy is running. If a port was not entered during provisioning, the default port is 3004.
2. Click Next to continue.

17.4.7 Updating the WebGate Web Server Configuration

Your Web server must be configured to operate with the WebGate. Oracle recommends automatically updating your Web server configuration during installation. However, procedures for both automatic and manual updates are included.

Note: To manually update your Web server configuration

1. Click No when asked if you want to proceed with the automatic update, then click Next.
2. Review the screen that appears to assist you in manually setting up your WebGate Web server, and see "Manually Configuring Your Web Server" on page 17-12.
3. Return to the WebGate installation screen, click Next, and proceed to "Provisioning a 10g WebGate with OAM 11g" on page 17-4.

To automatically update your Web server configuration

1. Click Yes to automatically update your Web server then click Next (or click No and see "Manually Configuring Your Web Server"):

   - **Most Web servers**—Specify the absolute path of the directory containing the Web server configuration file.
   - **IIS Web Servers**—The process begins immediately and may take more than a minute. For more information, see Chapter 19, "Configuring the IIS Web Server for 10g WebGates".

     You might receive special instructions to perform before you continue. Setting various permissions for the /access directory is required for IIS WebGates only when you are installing on a file system that supports NTFS. The last installation panel provides instructions for manually setting various permissions that cannot be set on the FAT32 file system. In this case, these instructions may be ignored.
   - **Sun Web Servers**—Be sure to apply the changes in the Web server Administration console before you continue.

   A screen announces that the Web server configuration has been updated.
2. Click Next and continue with "Finishing WebGate Installation".

17.4.7.1 Manually Configuring Your Web Server

If, during WebGate installation, you declined automatic Web server updates, you must perform the task manually.

Note: If the manual configuration process was launched during WebGate installation, you can skip Step 1 in the following procedure.

To manually configure your Web server for the WebGate

1. Launch your Web browser, and open the following file, if needed. For example:

   \WebGate_install_dir\access\oblix\lang\langTag\docs\config.htm

   where \WebGate_install_dir is the directory where you installed the WebGate.

Note: If you choose manual IIS configuration during 64-bit WebGate installation, you can access details in the following path

WebGate_install_dir\access\oblix\lang\en-us\docs\dotnet_isapi.htm

---

2. Select from the supported Web servers and follow all instructions, which are specific to each Web server type, as you:

   - Make a back up copy of any file that you are required to modify during WebGate set up, so it is available if you need to start over.
   - Ensure that you return to and complete all original setup instructions to enable your Web server to recognize the appropriate Oracle Access Manager files.

   Note: If you accidentally closed the window, return to step 1 and click the appropriate link again. Some setups launch a new browser window or require you to launch a Command window to input information.

3. Continue with "Finishing WebGate Installation".

17.4.8 Finishing WebGate Installation

The ReadMe information provides details about documentation and Oracle.

Note: If you are installing a 64-bit IIS WebGate, see "Finishing 64-bit WebGate Installation" in Chapter 19.

To finish the WebGate installation

1. Review the ReadMe information, then click Next to dismiss it.
2. Click Finish to conclude the installation.
3. Restart your Web server to enable configuration updates to take affect.
   - **IIS Web Servers**—Consider using `net stop iisadmin` and `net start w3svc` after installing the WebGate to help ensure that the Metabase does not become corrupted.
   - **Security-Enhanced Linux**: Run the chcon commands for the WebGate you just installed on this platform.
4. Proceed with following topics, as needed, then return to "Installing Artifacts and Certificates":
   - **Native POSIX Thread Library**: When installing Oracle Access Manager Web components for use with NPTL, there is no need to set the environment variable LD_ASSUME_KERNEL to 2.4.19.
   - **Apache2, OHS2, IHS2 Web Servers**: Chapter 18, "Configuring Apache, OHS, IHS for 10g WebGates"
   - **IIS Web Servers**: Consider using `net stop iisadmin` and `net start w3svc` after installing the WebGate to help ensure that the Metabase does not become corrupted. See also Chapter 19, "Configuring the IIS Web Server for 10g WebGates".
   - **ISA Web Servers**: Chapter 20, "Configuring the ISA Server for 10g WebGates"
   - **Lotus Domino Web Servers**: Chapter 21, "Configuring Lotus Domino Web Servers for 10g WebGates"

17.4.9 Installing Artifacts and Certificates

The ObAccessClient.xml file is one result of product of provisioning. After WebGate installation, you must copy the file to the WebGate installation directory path. If you received signed WebGate 10g certificates after installing WebGate, you can use the following procedure to install these as well.

To install artifacts (and certificates) for WebGate 10g

1. Gather WebGate 10g provisioning artifacts (and certificate files, if needed). For example:
   - ObAccessClient.xml
   - password.xml (if needed)
   - aaa_key.pem (your private key generated by openSSL).
   - aaa_cert.pem (signed certificates in PEM format)
2. Copy the files to the WebGate host: *WebGate_install_dir*\access\oblix\config.
3. Restart the WebGate Web server.

17.4.10 Confirming WebGate Installation

After WebGate installation and Web server updates, you can enable WebGate diagnostics to confirm that your WebGate is running properly.

To review WebGate diagnostics

1. Confirm OAM 11g components are running.
2. Specify the following URL for WebGate diagnostics. For example:

   Most Web Servers—http(s)://*hostname*:*port*/access/oblix/apps/webgate/bin/webgate.cgi?progid=1

   IIS Web Servers—http(s)://*hostname*:*port*/access/oblix/apps/webgate/bin/webgate.dll?progid=1

   where *hostname* refers to the name of the computer hosting the WebGate; *port* refers to the Web server instance port number.
3. The WebGate diagnostic page should appear.
   - **Successful**: If the WebGate diagnostic page appears, the WebGate is functioning properly and you can dismiss the page. Go to "Configuring Centralized Logout for 10g WebGate with OAM 11g".

     > **Note:** If this WebGate will replace the IDM Domain Agent, proceed to "Updating the WebLogic Server Plug-in" on page 17-18

   - **Unsuccessful**: WebGate should be uninstalled and reinstalled, as described in "Removing a 10g WebGate from the OAM 11g Deployment" on page 17-25.

17.5 Configuring Centralized Logout for 10g WebGate with OAM 11g

OAM 10g agents provide out of the box support for logout in a single DNS domain. To support logout across multiple DNS domains, 10g agents required customization.

With OAM 11g, session management is centralized in the OAM Server is maintained by the OAM Server and Logout support across different DNS domains is supported out of the box. OAM 11g:

- Clears the ObSSOCookie for the agent
- Clears the session in the server

With an OAM 11g Server and an OAM 10g WebGate, the application must always invoke /oamsso/logout.html, which:

- Sets the ObSSOCookie to loggedout (by invoking logout on the WebGate)
- Constructs end_url (as a URL) and redirects to the server logout URL (/oam/server/logout)

For more logout information, see "Configuring Centralized Logout for 10g WebGate with OAM 11g Servers" on page 11-7.

17.6 Replacing the IDM Domain Agent with an OAM 10g WebGate

Oracle Access Manager and Oracle Identity Manager are among the Oracle Fusion Middleware 11g components. During initial configuration with the WebLogic Server Configuration Wizard, the IDM Domain Agent is registered with OAM 11g along with the IDM domain host identifier and an application domain named for the agent.

Oracle Fusion Middleware uses OAM 11g to protected Oracle Identity Management consoles out of the box using the IDM Domain Agent.

To protect applications beyond containers, you can replace the IDM Domain Agent with a 10g WebGate (to protect the same set of applications using the same application domain and policies as the pre-registered IDM Domain Agent).

Task overview: Replacing the IDM Domain Agent with an OAM 10g WebGate

1. Provisioning a 10g WebGate to Replace the IDM Domain Agent
2. Installing a 10g WebGate to Replace the IDM Domain Agent
3. Updating the WebLogic Server Plug-in
4. Optional: Confirming the AutoLogin Host Identifier for an OAM / OIM Integration
5. Optional: Configuring OAM Security Providers for WebLogic
6. Optional: Disabling the IDM Domain Agent
7. Verification

17.6.1 Provisioning a 10g WebGate to Replace the IDM Domain Agent

Provisioning is the process of creating a WebGate registration in the OAM Administration Console. The following procedure walks through provisioning using the remote registration tool, in-band mode.

See Also:

- Chapter 6 for more information about the remote registration tool, processing, and request files
- Chapter 5 if you prefer using the OAM Administration Console

In this example, OAMRequest_short.xml is used as a template to create an agent named 10g4IDM, protecting /.../*, and declaring a public resource, /public/index.html. Your values will be different.

> **Note:** To use IDMDomainAgent policies with the replacement WebGate, ensure that the WebGate registration is configured to use the IDMDomain Host Identifier and Preferred Host.

To reuse existing IDMDomainAgent policies you can specify IDMDomain as the hostidentifier in the OAMReqRequest xml for the WebGate registration to set IDMDomain as the HostIdentifier and preferredHost. Alternatively, you can edit the Agent registration using the OAM Administration Console.

To provision a 10g WebGate to replace the IDM Domain Agent

1. Acquire the remote registration tool and set up the script for your environment. For example:

   a. Locate RREG.tar.gz file in the following path:

   ```
   WLS_home/Middleware/domain_home/oam/server/rreg/client/RREG.tar.gz
   ```

   b. Untar RREG.tar.gz file to any suitable location. For example: rreg/bin/oamreg.

   c. In the oamreg script, set the following environment variables based on your situation (client side or server side) and information in Table 6–7:

   OAM_REG_HOME = *exploded_dir_for_RREG*.tar/rreg
   JDK_HOME = *Java_location_on_the_computer*

2. Create the registration request and ensure that the autoCreatePolicy parameter is set to false:

   a. Locate OAMRequest_short.xml and copy it to a new file. For example:

   ```
   WLS_home/Middleware/domain_home/oam/server/rreg/bin/oamreg/
   ```

   Copy: OAMRequest.xml

   To: *10g4IDM*.xml

   b. Edit *10g4IDM*.xml to include details for your environment. For example, if you are changing from the IDMDomainAgent to a 10g WebGate Agent your request might look like the following:

   ```
   <OAMRegRequest>
       <serverAddress>http://sample.us.oracle.com:7001</serverAddress>
       <hostIdentifier>10g4IDM</hostIdentifier>
       <agentName>10g4IDM</agentName>
       <primaryCookieDomain>.us.example.com</primaryCookieDomain>
       <autoCreatePolicy>false</autoCreatePolicy>
       <logOutUrls><url>/oamsso/logout.html</url></logOutUrls>
       ...retain defaults for remaining elements...
       ...
       ...
   </OAMRegRequest>
   ```

   See Also: "Creating the Registration Request" on page 6-19

3. Provision the agent. For example:

   a. Locate the remote registration script.

      Linux: rreg/bin/oamreg.sh

      Windows: rreg\bin\oamreg.bat

   b. From the directory containing the script, execute the script using inband mode. For example:

      $ ./bin/oamreg.sh inband input/*10g4IDM*.xml

      ```
      Welcome to OAM Remote Registration Tool!
      Parameters passed to the registration tool are:
      Mode: inband
      Filename: ...
      ```

   c. When prompted, enter the following information using values for your environment:

      ```
      Enter your agent username: userame
         Username:  userame
      Enter agent password: ********
      Do you want to enter a Webgate password?(y/n)
          n
      iv.Do you want to import an URIs file?(y/n)
          n
      ```

   d. Review the final message to confirm that this was a successful registration:

      ```
      Inband registration process completed successfully! Output artifacts are
      created in the output folder"
      ```

4. Log in to the OAM Console and review the new registration:

   a. From the OAM 11g Console System Configuration tab, navigation tree, expand the following nodes:

      Agents
         OAM Agents
            10g Agents

   b. Double-click the agent's name to display the registration page and review the details (if you are installing a fresh WebGate, you must enter the following details during installation). For example:

      Agent Name—Enter this as the WebGate ID during WebGate installation.

      Access Client Password—Enter this as the WebGate password during WebGate installation. If no password was entered, you will leave the field blank.

      Access Server Host Name—Enter the DNS host name for the primary OAM 11g Server with which this WebGate is registered.

   c. **OAM Proxy Port**—From the OAM Console, System Configuration tab, navigation tree, double click Server Instances and locate the port on which the OAM Proxy is running.

5. Ignore the ObAccessClient.xml file that is created as a result of provisioning.

6. Proceed to "Updating the WebLogic Server Plug-in".

17.6.2 Installing a 10g WebGate to Replace the IDM Domain Agent

After provisioning you must install the 10g WebGate to replace the IDM Domain Agent. During the installation, you must provide some of the same information for the WebGate as you did when provisioning it.

Prerequisites

Provisioning a 10g WebGate to Replace the IDM Domain Agent

Task overview: Installing the WebGate includes

1. Locating and Installing the Latest OAM 10g WebGate for OAM 11g
2. Replacing the IDM Domain Agent: Proceed to "Updating the WebLogic Server Plug-in".

17.6.3 Updating the WebLogic Server Plug-in

After provisioning and installing the 10g WebGate to replace the IDM Domain Agent, the mod_wl_ohs.conf file requires specific entries to instruct the WebGate Web server to forward requests to the applications on the WebLogic Server.

Note: The generic name of the WebLogic Server plug-in for Apache is mod_weblogic. For Oracle HTTP Server 11g, the name of this plug-in is mod_wl_ohs (the actual binary name is mod_wl_ohs.so). Examples show exact syntax for implementation.

Example 17–1 illustrates the areas that must be changed using sample entries. Entries for your environment will be different.

Example 17–1 Updates for the 10g WebGate in mod_wl_ohs.conf

```
<IfModule weblogic_module>
   <Location /oamconsole>
         SetHandler weblogic-handler
         WebLogicHost hostname.us.sample.com
         WebLogicPort    6162
   </Location>
   <Location apmmconsole>
         SetHandler weblogic-handler
         WebLogicHost hostname.us.sample.com
         WebLogicPort    6162
   </Location>
...

</IfModule>
```

Note: You need similar Location entries for each of the URIs for all the applications that were earlier accessed directly on the WebLogic Server.

Prerequisites

Installing a 10g WebGate to Replace the IDM Domain Agent

To update the mod WebLogic configuration for your environment

1. Locate the mod_wl_ohs.conf file in the following path:

   ```
   <OHS-INSTANCE_HOME>/config/OHS/<INSTANCE_NAME>/mod_wl_ohs.conf
   ```

2. Edit the file to include a Location element for each application URI that was previously accessed directly on the WebLogic Server (see Example 17–1).
3. Save the file.
4. Restart the Web server.
5. Proceed to the following task, as needed:
   - Confirming the AutoLogin Host Identifier for an OAM / OIM Integration
   - Configuring OAM Security Providers for WebLogic

17.6.4 Confirming the AutoLogin Host Identifier for an OAM / OIM Integration

This topic describes how to confirm (or configure) Oracle Identity Manager (OIM) automatic login functionality when you have Oracle Access Manager integrated with OIM.

> **Note:** Skip this step if you do not have Oracle Access Manager 11g integrated with Oracle Identity Manager. 11g.

The AutoLogin functionality when OIM is integrated with OAM 11g requires the 10g WebGate Web server host name and port in the list of host identifiers for the IDM Domain Agent.

> **Note:** If you have a load balancer in front of the 10g WebGate Web server, you must also include the load balancer's host name and port during Step 3.

The agentBaseUrl parameter is used to update a given Host Identifier. However, if automatic policy creation is set to false, the remote registration utility does not create the application domain and does not honor the agentBaseUrl parameter.

The following procedure shows how to confirm (or configure) the AutoLogin host identifier for an OAM/OIM integration. You values will be different.

Prerequisites

Updating the WebLogic Server Plug-in

To configure the AutoLogin Host Identifier for an OAM / OIM Integration

1. From the Policy Configuration tab navigation tree, expand the Shared Components and Host Identifiers nodes, if needed, and select IDMDomain:

   Shared Components
     Host Identifiers
       IDMDomain

2. In the Operations panel, confirm that all host name and port combinations are listed for this Host Identifier.

3. In the Operations panel, confirm that the host and port of the Web server on which the 10g WebGate is (or will be) configured is listed. If not, add the entry:

   a. Click + button on the Operations panel.

   b. Host Name: Enter the 10g WebGate Web server host name in the Operations panel Host Name column.

   c. Port: Enter the 10g WebGate Web server port number in the Operations panel Port column.

   d. Load Balancer: If you have a load balancer in front of the 10g WebGate Web server, add the load balancer's host name and port in the Operations panel.

   e. Click Apply on the Host Identifier page.

4. Proceed to "Configuring OAM Security Providers for WebLogic".

17.6.5 Configuring OAM Security Providers for WebLogic

This section describes how to configure the WebLogic Security Providers to ensure Single Sign On using OAM 11g and the 10g WebGate.

> **Note:** Skip this step if you do not have Oracle Access Manager 11g integrated with Oracle Identity Manager 11g.

Refer to following topics for more information on setting up the security providers for the OAM 10g WebGate.

- About Security Providers
- Setting Up Security Providers for the 10g WebGate

17.6.5.1 About Security Providers

To complete the OAM 11g SSO configuration when a 10g WebGate is replacing the IDM Domain Agent requires configuring the following security providers in a WebLogic Server domain:

- OAM Identity Asserter: Uses token-based authentication and asserts the OAM SSO header and token.
- OID (or OVD) Authenticator: Creates the Subject and populates it with the correct principals.

  Depending on the store where your users are located, you configure either the Oracle Internet Directory Authenticator or the Oracle Virtual Directory Authenticator as the primary credential authenticator.
- Default Authenticator: This default WebLogic Authentication provider allows you to manage users and groups in one place: the embedded WebLogic Server LDAP server. This Authenticator is used by the Oracle WebLogic Server to login administrative users:

When you configure multiple Authentication providers, you use the JAAS Control Flag for each provider to control how the Authentication providers are used in the login sequence. You can choose the following the JAAS Control Flag settings, among others:

- REQUIRED—The Authentication provider is always called, and the user must always pass its authentication test. Regardless of whether authentication succeeds or fails, authentication still continues down the list of providers. The OAM Identity Asserter is required.

- SUFFICIENT—The user is not required to pass the authentication test of the Authentication provider. If authentication succeeds, no subsequent Authentication providers are executed. If authentication fails, authentication continues down the list of providers. Both the Oracle Internet Directory (or Oracle Virtual Directory) and the Default Authenticator are sufficient.
- OPTIONAL—When additional Authentication providers are added to an existing security realm, the Control Flag is set to OPTIONAL by default. You might need to change the setting of the Control Flag and the order of providers so that each Authentication provider works properly in the authentication sequence.

  The user is allowed to pass or fail the authentication test of this Authentication provider. However, if all Authentication providers configured in a security realm have the JAAS Control Flag set to OPTIONAL, the user must pass the authentication test of one of the configured providers.

  > **See Also:** "Configuring Authentication Providers" in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for a complete list of Authentication providers and details about configuring the Oracle Internet Directory provider to match the LDAP schema for user and group attributes.

Oracle Access Manager JAR are WAR files for authentication providers are available when you install an Oracle Fusion Middleware product (Oracle Identity Management, Oracle SOA Suite, or Oracle WebCenter). If you have a Fusion Middleware application, you already have the files you need.

- **oamAuthnProvider.jar**: Includes files for both the Oracle Access Manager Identity Asserter for single sign-on and the Authenticator for Oracle WebLogic Server 10.3.1+. A custom Oracle Access Manager AccessGate is also provided to process requests for Web and non-Web resources (non-HTTP) from users or applications.
- **oamauthenticationprovider.war**: Restricts the list of providers that you see in the Oracle WebLogic Server Console to only those needed for use with Oracle Access Manager.

  When you deploy the extension, the Administration Console creates an in-memory union of the files and directories in its WAR file with the files and directories in the extension WAR file. Once the extension is deployed, it is a full member of the Administration Console: it is secured by the WebLogic Server security realm, it can navigate to other sections of the Administration Console, and when the extension modifies WebLogic Server resources, it participates in the change control process For more information, see the *Oracle Fusion Middleware Extending the Administration Console for Oracle WebLogic Server*.

17.6.5.2 Setting Up Security Providers for the 10g WebGate

The following procedure requires the WebLogic Server Administration Console. This example illustrates setting up the Oracle Internet Directory provider with the OAM Identity Asserter and Default Authenticator. The steps are the same for OVD, should you need this.

> **Note:** If you have a Fusion Middleware application, you already have the files you need and you can skip Step 1 of the following procedure. With no Fusion Middleware application, however, you have a stand-alone Oracle WebLogic Server and must obtain the JAR and WAR files from Oracle Technology Network as described in Step 1.

Prerequisites

Updating the WebLogic Server Plug-in

To set up providers in a WebLogic Server domain for OAM 10g WebGate with OAM 11g

1. **No Oracle Fusion Middleware Application**: Obtain the Oracle Access Manager provider:

   a. Log in to Oracle Technology Network at:

   ```
   http://www.oracle.com/technology/software/products/middleware/ht
   docs/111110_fmw.html
   ```

   b. Locate the oamAuthnProvider ZIP file with Access Manager WebGates (10.1.4.3.0):

   ```
   oamAuthnProvider<version number>.zip
   ```

   c. Extract and copy oamAuthnProvider.jar to the following path on the computer hosting Oracle WebLogic Server:

   ```
   BEA_HOME/wlserver_10.x/server/lib/mbeantypes/oamAuthnProvider.jar
   ```

2. **With Oracle Fusion Middleware Application Installed**:

   a. Locate oamauthenticationprovider.war in the following path:

   ```
   ORACLE_INSTANCE/modules/oracle.oamprovider_11.1.1/oamauthenticationprovi
   der.war
   ```

   b. Copy oamauthenticationprovider.war to the following location:

   ```
   BEA_HOME/wlserver_10.x/server/lib/console-ext/autodeploy/oamauthentication
   provider.war
   ```

3. Log in to the WebLogic Server Administration Console and click **Security Realms**, ***Default Realm Name***, and click **Providers**.

4. **OAM Identity Asserter**: Perform the following steps to add this provider:

   a. Click Authentication, click New, and then enter a name and select a type:

   Name: `OAM ID Asserter`

   Type: **OAMIdentityAsserter**

   OK

   b. In the Authentication Providers table, click the newly added authenticator.

   c. Click the Common tab, set the Control Flag to **REQUIRED**, and click Save

5. **OID Authenticator:** Perform the following steps to add this provider.

   a. Click **Security Realms**, ***Default Realm Name***, and click **Providers**

   b. Click New, enter a name, and select a type:

      Name: *`OID Authenticator`*

      Type: OracleInternetDirectoryAuthenticator

      OK

   c. In the Authentication Providers table, click the newly added authenticator.

   d. On the Settings page, click the **Common** tab, set the Control Flag to **SUFFICIENT**, and then click Save.

   e. Click the **Provider Specific** tab and specify the following required settings using values for your own environment:

      Host: Your LDAP host. For example: *`localhost`*

      Port: Your LDAP host listening port. For example: *`6050`*

      Principal: LDAP administrative user. For example: *`cn=orcladmin`*

      Credential: LDAP administrative user password.

      User Base DN: Same searchbase as in Oracle Access Manager.

      All Users Filter: For example: `(&(uid=*)(objectclass=person))`

      User Name Attribute: Set as the default attribute for username in the LDAP directory. For example: `uid`

      Group Base DN: The group searchbase (same as User Base DN)

      Do not set the All Groups filter as the default works fine as is.

      Save.

6. **Default Authenticator**: Perform the following steps to set up the Default Authenticator for use with the Identity Asserter:

   a. Go to **Security Realms**, ***Default Realm Name***, and click **Providers**.

   b. Click Authentication, Click **DefaultAuthenticator** to see its configuration page.

   c. Click the Common tab and set the Control Flag to **SUFFICIENT**.

   d. Save.

7. **Reorder Providers**:

   a. Click **Security Realms**, ***Default Realm Name***, **Providers**.

   b. On the Summary page where providers are listed, click the **Reorder** button

   c. On the **Reorder Authentication Providers** page, select a provider name and use the arrows beside the list to order the providers as follows:

      OAM Identity Asserter (REQUIRED)

      OID Authenticator (SUFFICIENT)

      Default Authenticator (SUFFICIENT)

   d. Click OK to save your changes

8. **Activate** Changes: In the Change Center, click Activate Changes

9. Reboot Oracle WebLogic Server.

10. Proceed as follows:

- **Successful**: Go to "Disabling the IDM Domain Agent".
- **Not Successful**: Confirm that all providers have the proper specifications for your environment, are in the proper order, and that `oamAuthnProvider.jar` is in the correct location as described in "About Security Providers" on page 17-20.

17.6.6 Disabling the IDM Domain Agent

This step is optional, not required. The IDMDomain Agent detects when the WebGate has performed the authentication and then goes silent. However, if the agent must be disabled, then either the WLSAGENT_DISABLED system property or environment variable must be set to true for each one of the servers on which the agent should be disabled. This applies to both AdminServer and OAM Servers.

You can disable the agent in one of two ways:

- Either set the `WLSAGENT_DISABLED` environment variable to true
- Or pass `WLSAGENT_DISABLED` as a System Property

Prerequisites

Configuring OAM Security Providers for WebLogic, if needed.

To disable the IDM Domain Agent

1. On the computer hosting the IDM Domain Agent, perform one the following tasks:
   - Either set the `WLSAGENT_DISABLED` environment variable to true:

     ```
     setenv WLSAGENT_DISABLED true
     ```

   - Or or pass `DWLSAGENT_DISABLED=true` as a System Property:

     ```
     -DWLSAGENT_DISABLED=true
     ```
2. Restart the Web server.

17.6.7 Verification

Oracle recommends testing your environment using the 10g WebGate to ensure that all applications that were previously protected by the IDM Domain Agent are now protected after configuring the 10g WebGate.

See Also:

- "Validating Authentication and Authorization in an Application Domain" on page 9-47
- Chapter 10, "Validating Connectivity and Policies Using the Access Tester"

17.7 Deploying Applications in a WebLogic Container

For details about this topic, see the *Oracle Fusion Middleware Application Security Guide*.

This section provides information about deployments that currently have (or will have) applications deployed in a WebLogic container:

17.8 Removing a 10g WebGate from the OAM 11g Deployment

Use the following procedure to remove the 10g WebGate from the OAM 11g deployment, if needed.

Note: Deleting an agent registration does not remove the associated host identifier, application domain, resources, or the agent instance.

Considerations

Web Server Configuration Changes: Web server configuration changes must be manually reverted after uninstalling the WebGate). For more information about what is added, see the appropriate chapter for your Web server.

WebGate IIS Filters: To fully remove a WebGate and related filters from IIS, you must do more than simply remove the filters from the list in IIS. IIS retains all of its settings in a metabase file. On Windows 2000 and later, this is an XML file that can be modified by hand. For more information, see "Removing a 10g WebGate from the OAM 11g Deployment" on page 19-27.

Prerequisites

Evaluate the application domain, resources, and policies associated with this agent and ensure that these are configured to use another agent or that they can be removed.

To uninstall the 10g WebGate

1. Turn off the Web server for the WebGate you will remove.

Note: If you don't turn off the Web server, uninstall might fail and the backup folder will not be removed. If this happens, you need to manually remove the backup folder.

---

2. On the WebGate registration page in the OAM Administration Console, click the Disable box beside the State option to disable the WebGate.
3. **Language Packs**: Remove installed Language Packs (except the one selected as the default Administrator language (locale)) as follows:
   - Locate the appropriate Language Pack file in the component’s uninstall directory. For example:

     WebGate_install_dir\uninstIdentityLP_fr-fr
     \uninstaller.exe

   - Run the Language Pack Uninstaller program to remove the files.
   - Repeat this process to remove the same Language Pack from associated components.
   - Stop and restart WebGate Web server to re-initialize proper language support.
   - Repeat this process to remove each Language Pack (except the one selected as the default Administrator language (locale)).
4. Perform the following steps to remove 10g WebGate configuration data:
   - If you have only one instance of an Oracle Access Manager component, complete step 4 to remove it.

- If you have multiple instances of a component, see also step 5.

5. Locate and run the Uninstaller program for the specific component to remove Oracle Access Manager files. For example:

   WebGate_install_dir\access_uninstWebGate\uninstaller.exe

   Note: On UNIX systems, use uninstaller.bin

6. **Multiple Instances**: If you have multiple WebGate instances and want to remove one or all of them, you must use a specific method for your platform:

   - **Windows**: The last component can be uninstalled from Add/Remove programs. Others can be uninstalled by running the uninstall program from the \access \uninstComponent directory.
   - **UNIX**: You must always run uninstaller.bin.

7. Remove Oracle Access Manager-related updates to your Web server configuration. For details about specific Web servers, see Chapter 18, Chapter 19, Chapter 20, Chapter 21.

8. Restart the Web server.

9. Remove the *WebGate_install_dir* directory if it remains, especially if you plan to reinstall it.

18

Configuring Apache, OHS, IHS for 10g WebGates

Oracle Access Manager provides WebGates for Web servers powered by Apache v2. This includes Apache, Oracle HTTP Server, and IBM HTTP Server (IHS).

This chapter provides details about configuring the three Web server types, and includes:

18.1 Prerequisites

Ensure that your OAM 11g Administration Console is running and get familiar with:

- "Introduction to Policy Enforcement Agents" on page 5-1
- "About Installing Fresh OAM 10g WebGates to Use With OAM 11g" on page 17-2

18.2 About Oracle HTTP Server and Oracle Access Manager

Oracle Access Manager Web component package names for Oracle HTTP Server are designated with OHS, as follows:

- Oracle HTTP Server 11g is based on Apache v2.2; package names include OHS11g, for example:

Oracle_Access_Manager10_1_4_3_0_ *platform*_OHS11g_WebGate

- Oracle HTTP Server 10*g* R2 (10.1.2) and 10g (10.1.3.1.0) provide packages based on Apache v1.3 and Apache v2.0:

  Apache v2.0-based packages include OHS2, for example:
  Oracle_Access_Manager10_1_4_3_0_*platform*_OHS2_WebGate

  Apache v1.3-based packages include OHS, for example:
  Oracle_Access_Manager10_1_4_3_0_*platform*_OHS_WebGate

The following Oracle HTTP Server releases will operate with Oracle Access Manager:

Oracle HTTP Server 11g: Oracle Access Manager WebGates Oracle HTTP Server 11g can be used like WebGates for any other Web server. In addition, this WebGate for Oracle HTTP Server 11g is a key component when configuring enterprise-level single sign-on for Oracle Fusion Middleware 11g. For details, see the *Oracle Fusion Middleware Security Guide*. See also the *Oracle Fusion Middleware Administrator's Guide for HTTP Server 11g Release 1 (11.1.1).*

Oracle HTTP Server 10g (10.1.3.1.0): Provides two packages (one based on Apache v1.3 and another based on Apache v2.0). WebGates can be installed on a standalone Oracle HTTP Server. OHS2 WebGate must be installed on the Oracle Application Server to enable integration with Oracle single sign-on. During installation, the WebGate is installed as a module on OHS2.

Be sure to familiarize yourself with Oracle HTTP Server Web component requirements, as described in "Preparing Your Web Server" on page 18-7.

18.3 About Oracle Access Manager with Apache and IHS v2 WebGates

Oracle Access Manager provides components for Apache v2 Web servers and the IBM HTTP Server in addition to the Oracle HTTP Server. The IBM HTTP Server (IHS2) is a variation of Apache v2. Unless otherwise stated, the following information applies to all three:

- Apache v2.0.5.2 WebGate
- Apache v2.0.48 WebGate, including reverse proxy if you choose to activate this capability.
- Apache v2.0.47 WebGate for the IBM HTTP Server (IHS2) powered by Apache, including reverse proxy if you choose to activate this capability.

Note: For the latest Oracle Access Manager certification information, see:

`http://www.oracle.com/technology/products/id_mgmt/coreid_acc/pdf/oracle_access_manager_certification_10.1.4_r3_matrix.xls`

Each platform-specific installation package supports both plain and SSL-capable Apache modes. The number 2 in a file name indicates that this component is based on Apache v2. For example:

AIX: Oracle_Access_Manager10_1_4_3_0_power-aix_IHS2_WebGate

Linux: Oracle_Access_Manager10_1_4_3_0_ linux_Apache2_WebGate

Solaris: Oracle_Access_Manager10_1_4_3_0_sparc-s2_Apache2_WebGate

Windows: Oracle_Access_Manager10_1_4_3_0_Win32_APACHE2_WebGate

Earlier Oracle Access Manager releases included separate platform-specific installation packages for plain versus SSL-capable modes. For example, two WebGate files were provided for each platform: the APACHE_WebGate, and the APACHESSL_WebGate.

There have been no functional changes to Oracle Access Manager components to support these Web servers. Oracle Access Manager authentication occurs through the WebGate using HTTP basic, form, or SSL client certificates. Authorization for Web resources by authenticated users, and simple and multi-domain SSO with other Web servers or applications, also occurs through the WebGate.

18.3.1 About the Apache HTTP Server

The Apache HTTP Server is an open-source HTTP Web server project of the Apache Software Foundation. The project goal is to provide a secure, efficient and extensible server and HTTP services that meet current HTTP standards.

For more information, see "About Apache v2 Architecture and Oracle Access Manager" on page 18-4.

18.3.2 About the IBM HTTP Server

The IBM HTTP Server (IHS) is a variation of Apache v2. Portions of the IBM HTTP Server are based on software developed by The Apache Group. The IBM HTTP Server component also includes software developed by the OpenSSL Project and software developed by Eric Young.

Details about the Apache architecture and Oracle Access Manager, discussed in "About Apache v2 Architecture and Oracle Access Manager" on page 18-4 apply to IHS with the following exceptions:

- Previous versions of IHS required a separate IDS Client to use the mod_ibm_ldap module. With IHS powered by Apache v2.0.47, this is not a requirement.
- IHS v2.0.47 supports FIPS 140-2. FIPS support is disabled by default. To enable FIPS support, just add the SSLFIPSEnable directive to the httpd.conf file. Similarly, use SSLFIPSDisable directive to disable FIPS support.
- On AIX, ensure that the appropriate runtime library is installed before you install IHS v2.0.47.

For example on AIX 5.1, the xlC.rte 6.0 runtime library (for example: xlC.rte.6.0.0.0) must be installed before you install IHS v2.0.47. This library is required on AIX to install and use SSL with IHS v2. You can download this library from the following Web site:

```
http://www-912.ibm.com/eserver/support/fixes/fcgui.jsp
```

18.3.3 About the Apache and IBM HTTP Reverse Proxy Server

Typically, a reverse proxy is used in the following situations:

- To provide Internet users with access to a server behind a firewall
- To balance the load among several back-end servers, or to provide caching for a slower back-end server
- To bring several servers into the same URL space

The proxy_module implements a proxy/gateway for Apache and IHS powered by Apache. The client requires no special configuration; a reverse proxy appears like an ordinary Web server. The client makes requests as usual for content in the name-space of the reverse proxy. It is the reverse proxy that decides where those requests are sent. Content is returned as if the reverse proxy was the origin.

Important: The proxy_module can be used to implement a proxy capability for FTP, CONNECT (for SSL), HTTP/0.9, HTTP/1.0, and HTTP/1.1. However, only the reverse proxy capability is supported with the WebGate.

For more information, see "Requirements for Apache v2 Web Servers" on page 18-6.

18.4 About Apache v2 Architecture and Oracle Access Manager

The Apache v2 Web server provides a hybrid multi-threaded, multi-process architecture that is compatible with the thread-safe Oracle Access Manager libraries.

Important: Unless explicitly stated otherwise, all details in this discussion apply equally to Apache v2 and IHS v2 Web Servers for 10g WebGates.

In addition to the standard set of modules, the Apache v2 Web server includes Multi-Process Modules (MPMs) to bind network ports on the computer and to accept and process requests. The appropriate MPM must be compiled into the server and activated before you install a Oracle Access Manager component for Apache or IHS v2:

- **On Windows**: mpm_winnt is the default MPM on Windows platforms. mpm_winnt can use native networking features rather than the POSIX layer used in Apache 1.3.
- **On UNIX**: The prefork MPM is the default MPM for Apache v2 Web servers on UNIX platforms. The prefork MPM implements a non-threaded, pre-forking Web server that handles requests in a manner similar to Apache v1.3.

Note: If you compile Apache on UNIX with the mpm_worker_module for WebGate, you need to optimize the default pthread stacksize for WebGate to ensure optimal performance during multithreaded server implementation as described in "Apache v2 on UNIX with the mpm_worker_module for WebGate" on page H-19.

- **On AIX**: The worker MPM is the default MPM for IHS v2 on the AIX platform. The worker MPM implements a hybrid multi-process, multi-threaded server. The most important directives used to control this MPM are ThreadsPerChild and MaxClients. For details, see "Tuning Apache/IHS v2 for Oracle Access Manager WebGates" on page 18-28.

The Apache v2 Web server includes an Apache Portable Runtime (APR) library that provides an interface to platform-specific implementations, assures API developers predictable if not identical behavior regardless of platform, and eliminates the need for conditional compilation #Ifdefs. Although backward compatibility is supported with the include/apu_compat.h file, using the Apache v2 APR is recommended.

For more information, see your Apache v2 documentation. See also, "Tuning Apache/IHS v2 for Oracle Access Manager WebGates" on page 18-28.

The Apache architecture affects Oracle Access Manager components in different ways, as discussed in the following sections.

For WebGates installed with IHS and Apache v2

- There is no shared cache between processes.
- Each process maintains its own connections to the Access Server. Therefore, you should limit the number of WebGate connections. This issue is partially affected by the performance of the systems running the Web servers and Access Servers.

Note: WebGates for Apache v2 (and derivatives) can be used in installations that contain WebGates for other Web servers.

If you compile Apache on UNIX with the mpm_worker_module for WebGate, you need to optimize the default pthread stacksize for WebGate to ensure optimal performance during multithreaded server implementation as described in "Apache v2 on UNIX with the mpm_worker_module for WebGate" on page H-19.

Limitations of Apache and IHS v2 Web Servers

Due to limitations of the Apache v2 Web server, plug-ins configured for the Oracle Access Manager form-based authentication scheme do not pass variables when:

- The optional challenge parameter, passthrough:Yes, is included in the authentication scheme to pass login credentials through to a post-processing program.
- The form action is a CGI script that dumps all headers and variables passed to it and the method is called using the HTTP POST method.

For example:

```
<html>
<form name="myloginform" action="/access/...cgi" method="post">
```

18.5 Requirements for Oracle HTTP Server, IHS, Apache v2 Web Servers

Oracle Access Manager HTML pages use UTF-8 encoding. Apache-based Web servers, including Apache, Oracle HTTP Server, and IBM HTTP Server (IHS) allow administrators to specify a default character set for all HTML pages sent out using the `AddDefaultCharset` directive. This directive overrides any character specified by the application generating the HTML pages. If the `AddDefaultCharset` directive enables a character set other than UTF-8, Oracle Access Manager HTML pages are garbled.

Oracle recommends that you specify the `AddDefaultCharset` directive in the Web server configuration file (httpd.conf) as follows to ensure the correct display of Oracle Access Manager HTML pages:

```
AddDefaultCharset Off
```

See your Web server documentation for more information about this directive.

The following topics provide additional details you should be aware of:

- Requirements for IHS2 Web Servers
- Requirements for Apache and IHS v2 Reverse Proxy Servers
- Requirements for Apache v2 Web Servers

18.5.1 Requirements for IHS2 Web Servers

This discussion identifies specific requirements for IHS v2 with Oracle Access Manager. With IHS v2, you do not compile any source code to get the binaries. However, the following requirements do apply to IHS v2 Web servers:

- For an SSL capable configuration on AIX, the xLC.rte.6.0 runtime library is required.
- For an SSL capable configuration, the GSKit7 is required and can be downloaded from `https://techsupport.services.ibm.com/server/aix.fdc`.

18.5.2 Requirements for Apache and IHS v2 Reverse Proxy Servers

As discussed earlier, the proxy_module implements a proxy/gateway. The client requires no special configuration. Although the proxy_module can be used to implement a proxy capability for FTP, CONNECT (for SSL), HTTP/0.9, HTTP/1.0, and HTTP/1.1, only the reverse proxy capability is supported with certain Oracle Access Manager Apache and IHS v2 WebGates.

For Apache Web Servers: To use reverse proxy functions with Oracle Access Manager, you need to include the proxy module in the configure command. For example:

--enable-proxy: Apache proxy module

--enable-proxy-connect: Apache proxy CONNECT module

--enable-proxy-ftp: Apache proxy FTP module

--enable-proxy-http: Apache proxy HTTP module

You also need to load mod_proxy and the mod_proxy_http module into the server dynamically. A reverse proxy is activated using the ProxyPass directive or the [P] flag to the RewriteRule directive.

For IHS Web Servers: After installing the IHS Web server, reverse proxy configurations must be completed in the httpd.conf file in the following directory:

IHS_install_dir/conf directory

For more information, see "Activating Reverse Proxy for Apache v2 and IHS v2" on page 18-19.

18.5.3 Requirements for Apache v2 Web Servers

This discussion identifies specific requirements for Apache v2 with Oracle Access Manager. Additional information can be found in your Apache documentation:

PATH Variable: On UNIX systems, your PATH variable must contain the gcc location before you compile Apache v2. However, the Sun C compiler location must not be in your PATH variable. On Windows systems, Apache can be built using either command-line tools or the Visual Studio IDE Workbench. The command-line build requires that the environment reflect the PATH, INCLUDE, LIB and other variables that can be configured with the vcvars32 batch file.

Multi-Process Module (MPM): With Apache v2, a default MPM is provided for each platform to bind network ports on the computer and to accept and process requests.

Apache must have one, and only one, MPM in use at any time. If no MPM is selected during compilation, the default will be loaded into the Web server. You may activate the MPM during compilation.

mod_ssl: Oracle Access Manager supports Apache with or without SSL-capable communication. The base Apache Web server does not use SSL for browser connections and will not respond to HTTPS requests. For SSL-capable communication, Oracle Access Manager supports Apache with mod_ssl only. No SSL-specific Oracle Access Manager features operate with Apache-SSL.

mod_ssl relies on OpenSSL to provide the cryptography engine; mod_ssl provides an interface to the OpenSSL library. The OpenSSL library provides Strong Encryption using the Secure Sockets Layer and Transport Layer Security protocols.

With previous versions of Apache, the mod_ssl module had to be downloaded separately and compiled into the server. With Apache HTTP Server v2 module, mod_ssl comes as a loadable module that you can enable during configuration.

Multi-threading: Multi-threading is required for installations with Apache v1.3.27 or later.

Dynamic Shared Object (DSO): DSO support is required for WebGate. Apache modules that extend basic core server functionality may be either statically compiled for permanent inclusion in the Apache binary, or dynamically compiled and stored separately to load at runtime without recompiling. With Apache v1.3, mod_so had to be compiled. With Apache v2 on Windows systems, mod_so is a Base module and always included. With Apache v2 on UNIX, the loaded code typically comes from shared object files.

Note: Dynamically loaded Apache 1.3 modules cannot be used directly with Apache v2. Apache v1.3 modules must be modified to load dynamically or compile into Apache v2.

mod_perl: mod_perl embeds the Perl programming language in the Apache Web server. Without Perl, Apache v2 can still be built and installed; however, some support scripts written in Perl cannot be used.

Note: With Apache v.1.3.2x, some operating systems required additional options during configuration. However, to build Apache v2, there is no need to set any additional variables.

18.6 Preparing Your Web Server

The methods and steps to prepare your host computer for the Oracle Access Manager Web component installation depends upon the specific Web server and platform, as discussed in the following task overview.

To use reverse proxy functions with Oracle Access Manager, you need to include the proxy module in the configure command, as discussed in "About the Apache and IBM HTTP Reverse Proxy Server" on page 18-3. See also "Activating Reverse Proxy for Apache v2 and IHS v2" on page 18-19.

Task overview: Preparing your Web server and installing Oracle Access Manager

1. Install the IHS v2 Web server or compile and install the Apache v2 Web server as discussed in:
   - Preparing the IHS v2 Web Server
   - Preparing Apache and Oracle HTTP Server Web Servers on Linux
   - Preparing Oracle HTTP Server Web Servers on Linux and Windows Platforms
   - Setting Oracle HTTP Server Client Certificates
   - Preparing the Apache v2 Web Server on UNIX
   - Preparing the Apache v2 SSL Web Server on AIX
   - Preparing the Apache v2 Web Server on Windows
2. Activate reverse proxy capability if desired, as described in "Activating Reverse Proxy for Apache v2 and IHS v2" on page 18-19.
3. Install Oracle Access Manager components, as described elsewhere in this guide.
4. Finish Web server configuration, as described in "Verifying httpd.conf Updates for Oracle Access Manager WebGates" on page 18-22.
5. Refer to the following topics as needed:
   - "Tuning Oracle HTTP Server for Oracle Access Manager WebGates" on page 18-25
   - "Tuning OHS /Apache Prefork and MPM Modules for OAM" on page 18-26
   - "Tuning Apache/IHS v2 for Oracle Access Manager WebGates" on page 18-28

> **Note:** In all the procedures that follow, path name variables, modules, and options are examples provided only to illustrate the steps. Your environment will vary. Refer to your Web server documentation for additional details.

18.6.1 Preparing the IHS v2 Web Server

To prepare your IHS v2 Web server to accept and use the WebGate for IHS v2, you need to complete one or more of the following procedures, depending on your environment and requirements:

- Preparing the Host for IHS v2 Installation
- Installing the IBM HTTP Server v2
- Setting Up SSL-Capability
- Starting a Secure Virtual Host
- Activating Reverse Proxy for Apache v2 and IHS v2

When you have completed the appropriate procedures, you are ready to install the WebGate for IHS v2.

18.6.1.1 Preparing the Host for IHS v2 Installation

You need to complete this procedure to set up the host computer before you install the IHS Web server. For additional information, see "Requirements for IHS2 Web Servers" on page 18-6 and "Requirements for Apache v2 Web Servers" on page 18-6.

This example illustrates installation on AIX 5.1. Your environment may vary.

To prepare for IHS v2 installation

1. On the host computer, download and install the IBM Developer Kit, Java Technology Edition version 1.4 from the following site:

   http://www.ibm.com/java/jdk

   The IBM Developer Kit ships with the WebSphere Application Server or can be downloaded from this site.

2. On the host computer, download and install the xlC.rte 6.0 runtime for AIX 5.1, which is required by the GSKit7 runtime executable from the following site:

   https://techsupport.services.ibm.com/server/aix.fdc

3. On the host computer, create a new directory in which you will uncompress the IBM HTTP Server install image.

4. On the host computer, download the IBM HTTP Server install image from the following Web site:

   http://www-306.ibm.com/software/webservers/httpservers/

5. On the host computer, uncompress the install image in your new directory.

   For example:

   ```
   tar -xf IHS.tar
   ```

   A listing of the following files appears, based on your operating system:

   ```
   gskit.sh
   setup.jar
   gskta.rte (a GSKit runtime executable for AIX)
   ```

   You are ready to begin the installation, as described next.

6. Proceed to "Installing the IBM HTTP Server v2" on page 18-9.

18.6.1.2 Installing the IBM HTTP Server v2

The procedure that follows walks you through a typical IBM HTTP Web server installation. Alternatively, you may choose to perform a silent installation. In this case, you use silent.res file with the `java -jar setup.jar` -silent -options silent.res command. You can customize silent install options by editing the silent.res text file. All options are set to true by default. To disable an option, set its value to false.

To install the IBM HTTP Web server powered by Apache v2

1. Set your path to point to the Java Technology Edition version 1.4 installed on your computer in the previous example. For example:

   ```
   export PATH=$PATH:/usr/java14/java/bin
   ```

2. From to the directory where you uncompress the install image, type the following command:

   ```
   java -jar setup.jar
   ```

3. Choose the language in which to run the installation.

The Welcome to the InstallShield Wizard for the IBM HTTP Server appears.

4. Click Next to dismiss the Welcome screen.
5. Specify the directory name. For example:

   ```
   AIX: /usr/IBMIHS/
   ```

6. Click Next to continue.

   Options appear for a typical, custom, or developer installation. When you choose a typical installation, a list will appear with everything included and the size of the image. If you choose a custom installation, a list of components appears and you can clear the box next to the any components you do not want to install.

7. Select the type of installation you would like to perform, then click Next. For example:

   ```
   Typical
   ```

   The following message appears. You can click Cancel to stop the installation.

   ```
   Installing IBM HTTP Server. Please wait.
   ```

   The next message also appears. You can click Cancel to stop the inventory update.

   ```
   Updating the inventory.
   ```

8. Click Finish to complete your installation.
9. Stop then start the IHS server using the apachectl commands, as follows:

   For example:

   ```
   IHS2_install_dir/bin
   ./apachectl stop
   ./apachectl start
   ```

   where *IHS2_install_dir* is the directory where you installed the IHS v2 Web server.

You may configure the IHS v2 Web server in several modes either before or after installing the WebGate for IHS v2:

- Setting Up SSL-Capability
- Starting a Secure Virtual Host
- Activating Reverse Proxy for Apache v2 and IHS v2

18.6.1.3 Setting Up SSL-Capability

If you need to setup SSL-capability, use the following procedure either before or after installing the WebGate for IHS v2.

To setup SSL for IHS v2 using the default configuration file

1. Locate and open the following file:

   IHS2_install_dir/conf/httpd.conf

2. Specify the SSLEnable directive to enable SSL.
3. Specify a Keyfile directive and any SSL directives you want to enable.
4. Stop then start the IHS server, as follows. For example:

   ```
   IHS2_install_dir/bin
   ```

```
./apachectl stop
./apachectl start
```

where *IHS2_install_dir* is the directory where you installed the IHS v2 Web server.

5. Continue with the following procedures:

   - Starting a Secure Virtual Host
   - Activating Reverse Proxy for Apache v2 and IHS v2

18.6.1.4 Starting a Secure Virtual Host

If you need to start a secure virtual host, use the following procedure either before or after installing the WebGate for IHS v2.

To start an IHS v2 secure virtual host

1. Locate and open the following file:

   ```
   IHS2_install_dir/conf/httpd.conf
   ```

   where *IHS2_install_dir* is the directory where you installed the IHS v2 Web server.

2. Specify the SSLEnable directive in the virtual host stanza of the configuration file, to enable SSL for a virtual host.

   You can specify any directive, with the exception of the cache directives, inside a virtual host.

3. Specify a Keyfile directive and any SSL directives you want to enable for that particular virtual host.

4. Load the mod_ibm_ssl.so using the LoadModule directive in the conf file.

5. Stop then start the IHS virtual host, as follows. For example:

   ```
   IHS2_install_dir/bin
   ./apachectl stop
   ./apachectl start
   ```

   > **Note:** The start and stop instructions for an SSL implementation are the same as non-SSL-capable implementations.

6. Continue with Activating Reverse Proxy for Apache v2 and IHS v2.

18.6.2 Preparing Apache and Oracle HTTP Server Web Servers on Linux

When installing Oracle Access Manager WebGates for Apache or Oracle HTTP Server on Linux, you are prompted to install as the same user under which the Web server is running. See the User and Group directive entries in the httpd.conf file.

When installing Oracle Access Manager WebGates for vendor-bundled Apache v2 on Red Hat Enterprise Linux 4, ensure that all Oracle Access Manager WebGates are installed for Web server user & group (default: apache). See also "Tuning Apache/IHS v2 for Oracle Access Manager WebGates" on page 18-28.

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

18.6.3 Preparing Oracle HTTP Server Web Servers on Linux and Windows Platforms

When using Oracle Access Manager WebGates for Oracle HTTP Server v2 on Windows and Linux platforms, both the Perl module and the PHP module must be commented out in the httpd.conf.

Note: With Oracle HTTP Server 11g, there is no need to comment out any module for Oracle Access Manager WebGates on any platform.

18.6.4 Setting Oracle HTTP Server Client Certificates

When using cert_decode and credential_mapping authentication modules, you must ensure that the Client Certificate authentication scheme works properly with SSL-enabled Oracle HTTP Server by adding `+EarlierEnvVars` and `+ExportCertData` to the existing SSL options in the Oracle HTTP Server Web server configuration file. For example:

credential_mapping:

```
obMappingBase="o=company,c=us",obMappingFilter=
"(&(objectclass=InetOrgPerson)(mail=%certSubject.E%))"
```

ssl.conf must include:

```
SSLOptions +StdEnvVars +ExportCertData +EarlierEnvVars
```

To add ssl options to Oracle HTTP Server

1. Locate and open the Oracle HTTP Server Web server configuration file with a text editor. For example:

   $ORACLE_INSTANCE/ohs/conf/ssl.conf

2. In the ssl.conf file, add the following information to existing SSL options. For example:

   ```
   SSLOptions +StdEnvVars +ExportCertData +EarlierEnvVars
   ```

3. Save the file and restart the Web server.

18.6.5 Preparing the Apache v2 Web Server on UNIX

This discussion provides an overview and steps to prepare the Apache v2 HTTP Web server for Oracle Access Manager on UNIX platforms, including Solaris, UNIX, Linux, and AIX. See also "Preparing the Apache v2 SSL Web Server on AIX" on page 18-16

Apache v2 can be configured, built, and installed plain or as SSL-capable. After downloading and extracting Apache source files, you use a script (configure script on UNIX and the makefile.win make script for Windows) to compile the source tree for your environment.

Note: Basic requirements are the same regardless of your platform. However, the remainder of this discussion and the procedures that follow focus on UNIX platforms. For more information, see also "Preparing the Apache v2 SSL Web Server on AIX" on page 18-16.

When you configure Apache v2 on UNIX platforms, you specify the installation directory path name using the `-prefix=` option with the `./configure` command. During configuration you enable the modules that are appropriate for your environment. For example, mod_so is included in the server automatically when dynamic modules are included in the compilation. However, you can ensure the server is capable of loading DSOs by including the `-enable-so` option with the configure command. If you have multiple Perl interpreters installed, you can include the -with-perl option to ensure the correct interpreter is selected during configuration.

In the configure command, you can also include the options to enable mod_ssl, and to activate an MPM. After configuration, you can verify which MPM was chosen using `./httpd -l` to list every module that is compiled into the server.

When you finish configuring Apache, you build the various parts that form the Apache package using the make command then install the package under the installation directory you specified with the `-prefix=` option during configuration.

For steps and examples, see the following procedures and your Apache documentation:

- To prepare plain Apache v2 for UNIX
- To prepare SSL-capable Apache v2 on UNIX
- To prepare Apache v2 for Windows
- Activating Reverse Proxy for Apache v2 and IHS v2

In the procedures that follow, path name variables, modules, and options are examples provided only to illustrate the steps. Your environment will vary. Refer to your Web server documentation for additional details. There is no difference in the build procedure between Apache v2.0.48 and v2.0.52.

To prepare plain Apache v2 for UNIX

1. Confirm that your environment meets Apache requirements for the appropriate compiler and build tools, as described in Apache documentation located at:

   `http://httpd.apache.org/docs-2.0/install.html#requirements`

   Note: There are no known restrictions with regard to supported compiler versions for Apache v2 and Oracle Access Manager plug-ins. See the Apache documentation.

2. Download a complete, unmodified version of the Apache HTTP Server v2, as described in the Apache documentation. For example:

   `http://httpd.apache.org/download.cgi`

   Note: Be sure to download Perl, if needed.

3. Extract (uncompress, then untar) source files from the tarball, as described in the Apache documentation. For example:

   ```
   gzip -d httpd-2_0_48.tar.gz
   tar -xvf httpd-2_0_48.tar
   ```

   You can use the following step as an example of configuring the Apache source tree. If you compile Apache on UNIX with the mpm_worker_module for WebGate, see "Apache v2 on UNIX with the mpm_worker_module for WebGate" on page H-19.

   > **Note:** To use reverse proxy functions with Oracle Access Manager, you need to include the proxy module in the configure command, as discussed in "About the Apache and IBM HTTP Reverse Proxy Server" on page 18-3.

4. Ensure that you have the correct version of GNU gcc libraries in the proper path to build the Apache source; gcc libraries should be in the PATH:

   ```
   export PATH=/usr/local/packages/gcc-3.4.6/bin:$PATH
   ```

5. Configure the Apache source tree and enable or activate the desired modules using details in the Apache documentation. For example:

   ```
   cd apache_source_dir
   ./configure --with-mpm=prefork --prefix=apache_install_dir --with-included-apr
   ./configure --with-mpm=worker --prefix=apache_install_dir --with-included-apr
   ```

   where *apache_source_dir* refers to the directory where you extracted Apache and *apache_install_dir* refers to the directory where you want to install Apache.

6. Compile the Apache package you configured using the make command. For example:

   ```
   make
   ```

7. Install the Apache package in the configured directory path that you specified earlier using the --prefix= option. For example:

   ```
   make install
   ```

8. Customize the installation using instructions in the Apache documentation.

   For example, you may need to tune the httpd.conf to set basic values for:

   ```
   ServerName
   User/owner of the WebServer
   Group
   ```

   > **Note:** To view the complete list of values, use the command: ./configure --help.

9. Stop then restart the Apache Web server to test the installation using commands in the *apache_install_dir*/bin directory. For example:

   ```
   ./apachectl stop
   ./apachectl start
   ```

10. Continue with appropriate tasks for your environment, as follows:

    - To prepare SSL-capable Apache v2 on UNIX
    - Preparing the Apache v2 Web Server on UNIX
    - Activating Reverse Proxy for Apache v2 and IHS v2

The following procedure outlines how to prepare an SSL-capable Apache v2 Web server on UNIX. The Apache mod_ssl is loadable; however, this installation requires the Open Source toolkit for SSL/TLS. Again, be sure to download Perl, if needed. If AIX is the platform you are using, be sure to see "Preparing the Apache v2 SSL Web Server on AIX" on page 18-16 for additional information.

To prepare SSL-capable Apache v2 on UNIX

1. Confirm that your environment meets Apache requirements for the appropriate compiler and build tools, as described in Apache documentation located at:

   ```
   http://httpd.apache.org/docs-2.0/install.html
   ```

2. Download a complete, unmodified version of the Apache HTTP Server v2 and Open Source, as described in the Apache documentation.

   ```
   http://httpd.apache.org/download.cgi
   http://www.openssl.org/
   ```

3. Extract (uncompress, then untar) source files from the tarballs, as described in the Apache documentation. For example:

   ```
   gzip -d httpd-2_0_48.tar.gz
   tar -xvf httpd-2_0_48.tar
   gzip -d openssl-0_9_6f.tar.gz
   tar -xvf openssl-0_9_6f.tar
   ```

4. Configure the OpenSSL source tree, as described in Apache documentation. For example:

   ```
   cd openssl_source_dir
   ./config -fPIC --prefix=openssl_install_dir
   ```

   where *openssl_source_dir* refers to the directory where you extracted OpenSSL and *openssl_install_dir* refers to the directory where you want to install the configured OpenSSL package.

5. Compile the OpenSSL package in the installation directory you configured using the make command with the --prefix= option. For example:

   ```
   make
   ```

6. Issue the make test command to complete any sanity testing of OpenSSL and check the correct version of the tools required. For example:

   ```
   make test
   ```

7. Install the OpenSSL package in the configured directory path that you specified earlier using the --prefix= option. For example:

   ```
   make install
   ```

8. Configure the Apache source tree and enable or activate desired modules, as described in your Apache documentation. For example:

   ```
   cd apache_source_dir ./configure --prefix=apache_install_dir
   ```

```
 --enable-so \ --with-mpm='prefork' --with-perl=perl_interpreter_path \
--with-port=non_ssl_port --enable-ssl \ --with-ssl=openssl_install_dir
```

where *apache_source_dir* refers to the directory where you extracted Apache; *apache_install_dir* refers to the directory where you want to install Apache; and *openssl_install_dir* refers to the directory where you installed the configured OpenSSL package.

9. Compile using the make command to build the Apache SSL-capable package in the installation directory you configured using the --prefix= option. For example:

   ```
   make install
   ```

10. Install the Apache SSL-capable package in the configured directory path that you specified earlier using the --prefix= option. For example:

    ```
    make install
    ```

    You must explicitly make certificates for the Apache v2 server to enable SSL using the openssl tool located at *openssl_install_dir*/bin/. The make certificate command does not work with Apache v2.

11. Make certificates using the OpenSSL tool in the *openssl_install_dir*/bin directory, as described in your OpenSSL documentation and remember that "Common Name" is the fully qualified host name.

12. Customize the installation using instructions in the Apache documentation:

    - Tune the httpd.conf to set basic values for:

      ```
      ServerName
      User/owner of the WebServer
      `Group
      ```

    - Tune the ssl.conf to set basic values for:

      ```
       Listen 7000
       <VirtualHost _default_:7000>
       ServerName ps0733.persistent.co.in:7000
       SSLCertificateFile /home/qa/software/ws/apache/
       apache-2.0.48_ssl_7000/conf/ssl.crt/server.crt
       SSLCertificateKeyFile /home/qa/software/ws/apache/
       apache-2.0.48_ssl_7000/conf/ssl.key/server.key
      ```

13. Stop then restart the Apache Web server to test the installation using commands in the *apache_install_dir*/bin directory. For example:

    ```
    ./apachectl stop
    ./apachectl startssl
    ```

14. Continue with Activating Reverse Proxy for Apache v2 and IHS v2, if needed.

18.6.6 Preparing the Apache v2 SSL Web Server on AIX

While building the Apache v2 SSL Web server, the symbols from the OpenSSL Library libssl.a are exported into the httpd executable in Apache. The symbols needed by Oracle Access Manager from the OpenSSL library are:

- SSL_get_peer_certificate( )
- i2d_X509( )

During linking and binding on the AIX platform, any unused or unreferenced symbols are deleted. Therefore, the two symbols required by Oracle Access Manager are missing from the httpd executable.

You need to use openssl-0.9.7d to compile on AIX (openssl-0.9.7e does not compile on AIX). The rest of the steps are the same as on UNIXopenssl-0.9.7d.

Client Cert Authentication: If you are using Client Cert Authentication on the AIX platform, be sure to use AIX 5.2 Maintenance Level 4 with the following hot fix applied for dlsym problem on AIX:

`http://www-1.ibm.com/support/docview.wss?uid=isg1IY63366`

To prepare the AIX platform for Apache v2

1. Ensure that your AIX platform meets the system requirements for Oracle Access Manager.
2. See details in "Preparing the Apache v2 Web Server on UNIX" on page 18-12 and when building the Apache v2 Web server:
   - Use openssl-0.9.7d to compile the Web server for AIX.
   - Use the make command in the following manner:

     ```
     make MFLAGS=EXTRA_LDFLAGS='-Wl,-bE:OpenSSL_Symbols.exp'
     ```

where OpenSSL_Symbols.exp is the file containing the two required symbols. The symbol must be exported using the export file only, as shown.

> **Note:** Do not export the symbol on AIX with the following methods: -bnog: To suppress garbage collection of symbols -bexpal: To export all symbols -uSymbolName: To export a particular symbol.

18.6.7 Preparing the Apache v2 Web Server on Windows

Following are some details about how installing and configuring Apache v2 on Windows differs from Apache v2 on UNIX. For more information, see your Apache documentation.

During Installation: Apache will configure files in the \conf subdirectory to reflect the chosen installation directory. If any configuration files in this directory already exist, a new copy of the corresponding file will be written with the extension .ORIG. For example, \conf\httpd.conf.ORIG.

After Installation: Apache is configured using the files in the \conf subdirectory. These are the same files used to configure the UNIX version. However, there are a few differences.

You must edit the configuration files in the \conf subdirectory to customize Apache for your environment. These files will be configured during the installation; Apache is ready to run from the installation directory, with the documents server from the subdirectory htdocs. There are many options you should set before starting to use Apache. For example, Apache listens on port 80 unless you change the Listen directive in the configuration files or install Apache only for the current user.

Multi-Threading: Apache for Windows is multi-threaded, which means that it does not use a separate process for each request as Apache does on UNIX. Instead there are usually only two Apache processes running: a parent process, and a child which handles the requests. Within the child process each request is handled by a separate thread.

UNIX-Style Names: Apache uses UNIX-style names internally. The directives that accept filenames as arguments must use Windows filenames instead of UNIX filenames. However, you must use forward slashes, not back slashes. Drive letters may be used. However, if a drive letter is omitted, the drive with the Apache executable is assumed.

LoadModule Directive: Apache for Windows includes the ability to load modules at runtime without recompiling the server. If Apache is compiled normally, it will install a number of optional modules in the \Apache2\modules directory. To activate these or other modules, you must use the LoadModule directive. For example, to activate the status module, use the following (in addition to the status-activating directives in access.conf):

```
LoadModule status_module modules/mod_status.so
```

On UNIX, the loaded code typically comes from shared object files (.so extension), on Windows this may be either the .so or .dll extension.

Process Management Directives: These directives are also different for Apache on Windows.

Error Logging: During Apache startup, any errors are logged into the Windows event log, which provides a backup to the error.log file. For more information, see your Apache documentation.

Apache Service Monitor: Apache comes with an Apache Service Monitor utility. With it you can see and manage the state of all installed Apache services on any computer on your network. To manage an Apache service with the monitor, you must first install the service. Apache may be run as a service on Windows. For details, see your Apache documentation.

Starting, Restarting, Shutting Down: Running Apache as a service is the recommended method. An Apache service is typically started, restarted, and shut down using the Apache Service Monitor and commands like NET START Apache2 and NET STOP Apache2. You may also use standard Windows service management.

You may work with Apache from the command line using the apache command. Apache will execute and remain running until it is stopped by pressing Control-C. You may also run Apache from the Start Menu during installation.

> **Note:** Pressing Control-C may not allow Apache to end any current operations and clean up gracefully.

Apache Services Accounts: By default, all Apache services are registered to run as the system user (the LocalSystem account). The LocalSystem account has no network privileges through any Windows-secured mechanism. However, the LocalSystem account has wide privileges locally. For details about creating a separate account to run one or more Apache services, see your Apache documentation.

To prepare Apache v2 for Windows

1. Confirm that your environment meets Apache requirements, as described in Apache documentation located at:

   `http://httpd.apache.org/docs-2.0/install.html`

   For Windows installations a list of HTTP and FTP mirrors from which you can download Apache v2 is provided online.

When you complete the next step, be sure to download the version of Apache for Windows with the .msi extension.

2. Download a complete, unmodified version of the Apache HTTP Server v2 (and OpenSSL), as described in the Apache documentation. For example:

   ```
   http://httpd.apache.org/download.cgi
   http://www.openssl.org/
   ```

3. Install Apache v2 (run the .msi file you downloaded and supply requested information), using your Apache documentation as a guide.
4. Locate the .default.conf file, verify new settings, then update your existing configuration file if needed.
5. Start Apache, either in a console window or as a service.
6. Launch a browser and enter the following URL to connect to the server and access the default page. For example:

   ```
   http://localhost/
   ```

   A welcome page and a link to the Apache manual should appear. If not, look in the error.log file in the logs subdirectory.

   Once your basic installation is working, you need to configure it properly by editing the files in the \conf subdirectory.
7. Configure the Apache installation for your environment, using the Apache documentation as a guide.
8. Test your customized environment.
9. Continue with Activating Reverse Proxy for Apache v2 and IHS v2, if needed.

18.7 Activating Reverse Proxy for Apache v2 and IHS v2

The WebGates for Apache v2 and IHS v2 powered by Apache support reverse proxy capability, if you choose to activate this capability. The procedures to implement reverse proxy capability differ, depending on your environment:

- To activate reverse proxy capability for Apache v2 Web servers
- To activate reverse proxy capability for IHS v2 Web servers

18.7.1 Activating Reverse Proxy For Apache v2 Web Servers

For reverse proxy functions with Oracle Access Manager, you need to include the Apache proxy module in the configure command for the Web server. You also need to load mod_proxy and the mod_proxy_http module into the server dynamically. A reverse proxy is activated using the ProxyPass directive or the [P] flag to the RewriteRule directive.

Reverse proxy capability is activated using the ProxyPass directive or the [P] flag to the RewriteRule directive. It is not necessary to turn ProxyRequests on to configure a reverse proxy. Access control is less critical when using a reverse proxy (ProxyPass directive with ProxyRequests Off), because clients can contact only the hosts that you have specifically configured. You can control access to your proxy using the <Proxy> control block.

To activate reverse proxy capability for Apache v2 Web servers

1. Review "About the Apache and IBM HTTP Reverse Proxy Server" on page 18-3.
2. Include the Apache proxy module in the configure command for the Web server, if needed.

   For example:

   ```
   --enable-proxy
   --enable-proxy-connect
   --enable-proxy-ftp
   --enable-proxy-http
   ```

   See the Apache documentation for more information.
3. Use the ProxyPass directive or the [P] flag to the RewriteRule directive to activate a reverse proxy, as follows:

   ```
   Reverse Proxy
   ProxyRequests Off
   <Proxy *>
    Order deny,allow
    Allow from all
   </Proxy>
   ProxyPass /foo http://foo.example.com/bar
   ProxyPassReverse /foo http://foo.example.com/bar
   ```
4. Control access to your proxy using the <Proxy> control block as follows:

   ```
   <Proxy *>
    Order Deny,Allow
    Deny from all
    Allow from 192.168.0
   </Proxy>
   ```
5. Perform steps in Chapter 17, "Managing OAM 10g WebGates with OAM 11g", if you haven't yet done so.

18.7.2 Activating Reverse Proxy For IHS v2 Web Servers

Use the following procedure after installing the Web server.

To activate reverse proxy capability for IHS v2 Web servers

1. Review "About the Apache and IBM HTTP Reverse Proxy Server" on page 18-3
2. Install the IHS v2 Web server, as described in "Preparing the IHS v2 Web Server" on page 18-8.
3. Load the modules by including these lines (uncommented) in the Dynamic Shared Object section of the httpd.conf file in:

   IHS_install_dir/conf/httpd.conf

   ```
   LoadModule access_module modules/mod_access.so
   LoadModule auth_module modules/mod_auth.so
   LoadModule auth_dbm_module modules/mod_auth_dbm.so
   LoadModule include_module modules/mod_include.so
   LoadModule log_config_module modules/mod_log_config.so
   LoadModule env_module modules/mod_env.so
   LoadModule unique_id_module modules/mod_unique_id.so
   LoadModule setenvif_module modules/mod_setenvif.so
   LoadModule proxy_module modules/mod_proxy.so
   ```

```
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule mime_module modules/mod_mime.so
LoadModule dav_module modules/mod_dav.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule asis_module modules/mod_asis.so
LoadModule info_module modules/mod_info.so
LoadModule cgid_module modules/mod_cgid.so
LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule dir_module modules/mod_dir.so
LoadModule imap_module modules/mod_imap.so
LoadModule actions_module modules/mod_actions.so
LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
```

4. Directives Under the IfModule mod_proxy.c Tag--Use the information and the following examples to ensure that:

   - Allow or Deny conditions are appropriately commented.

     For example:

     ```
        <Proxy *>
           Order deny, allow
     #    Deny from all
           Allow from all
     #   Allow from .domain.com
     </Proxy>
     ```

   - URLs to be protected are mentioned in both the ProxyPass and the ProxyPassReverse directives.

     For example:

     ```
     <IfModule mod_proxy.c>
     ProxyRequests Off
     ProxyPass /testproxy http://bedford: 8809/testrev/
     ProxyPassReverse /testproxy http://bedford: 8809/testrev/
     ProxyPass /test2 http://bedford: 8809/testrev/
     ProxyPassReverse /test2 http://bedford: 8809/testrev/
     ```

5. Restart the Web server after any modifications to the httpd.conf file.

6. **Testing**: To access the proxy URL, access `http://<proxy_host>:80/testproxy/`

   Note:

   While testing, make sure the URLs have a trailing forward slash. Sometimes resources cannot be accessed without the forward slash at the end.

7. Enabling SSL on Reverse Proxy Server: Use the documentation on the IHS default page.

   For example, sample SSL settings in the DSO section of the httpd.conf file load the ibm_ssl_module as:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
```

8. Include the following directives in your httpd.conf file:

   ```
   SSLEnable
      Keyfile /opt/IBMIHS/bin/key.kdb
      SSLClientAuth none
      SSLProxyEngine on
   ```

9. Restart server.
10. Access the Web server URL and confirm that the browser is presented with a certificate.

    Note: You can switch back to open mode for the Web server simply by commenting out the preceding directives and restarting the server.

11. **key.kdb**: To generate the key.kdb, use the ikeyman utility (preferably in GUI mode) provided in the *IHS_install_dir*/bin directory.

    Note: The ikeyman utility uses the gsk7bas utility. However, you need to apply fix pack PQ83048 on gsk7bas.

12. Perform the following steps:
    - Complete 10g WebGate installation with OAM 11g as described in Chapter 17, "Managing OAM 10g WebGates with OAM 11g", if you haven't yet done so
    - Return to this chapter to perform remaining tasks in this chapter as needed.

18.8 Verifying httpd.conf Updates for Oracle Access Manager WebGates

It is a good idea to complete the following procedures to ensure that the Apache or IHS v2 httpd.conf file includes Web server configuration updates for Oracle Access Manager. For details, see:

- Verifying WebGate Details
- Verifying Language Encoding

To update httpd.conf for reverse proxy on IHS Web servers, see "Activating Reverse Proxy For IHS v2 Web Servers" on page 18-20. To customize httpd.conf for your Web server, see your Web server documentation.

18.8.1 Verifying WebGate Details

The example that follows shows the WebGate section in the httpd.conf file. The details will vary, depending on your environment. This example is provided only to illustrate the type of changes you will see in httpd.conf.

To verify the WebGate section in httpd.conf

1. Locate the updated httpd.conf file on the computer hosting the WebGate.
2. Open the httpd.conf file and ensure that the section that loads the WebGate in your platform is present.

   For example:

On Windows

```
#*** BEGIN Oblix NetPoint WebGate Specific ****
<IfModule mod_ssl.c>
LoadModule obWebgateModule "WebGate_install_
dir\access\oblix\apps\webgate\bin\webgatessl.d ll"
      WebGateInstalldir "WebGate_install_dir"
      WebGateMode PEER
      </IfModule>
<IfModule !mod_ssl.c>
LoadModule obWebgateModule "WebGate_install_
dir\access\oblix\apps\webgate\bin\webgate.dll"
      WebGateInstalldir "WebGate_install_dir"
      WebGateMode PEER
      </IfModule>
<Location "\oberr.cgi">
SetHandler obwebgateerr
      </Location>
      <LocationMatch "/*">
      AuthType Oblix
      require valid-user
      </LocationMatch>
#*** END Oblix NetPoint WebGate Specific ****
```

On UNIX

```
#*** BEGIN Oblix NetPoint WebGate Specific ****
LoadFile "/home/qa/netpoint/703/c1-copy/wg/access/oblix/lib/libgcc_s.so.1"
LoadFile "/home/qa/netpoint/703/c1-copy/wg/access/oblix/lib/libstdc++.so.5"
<IfModule mod_ssl.c>
   LoadModule obWebgateModule "/home/qa/netpoint/703/c1-copy/wg/access/oblix/
apps/webgate/bin/webgatessl.so"
</IfModule>
<IfModule !mod_ssl.c>
   LoadModule obWebgateModule "/home/qa/netpoint/703/c1-copy/wg/access/oblix/
apps/webgate/bin/webgate.so"
</IfModule>
WebGateInstalldir "/home/qa/netpoint/703/c1-copy/wg/access"
WebGateMode PEER
<Location /access/oblix/apps/webgate/bin/webgate.cgi>
SetHandler obwebgateerr
</Location>
<Location "/oberr.cgi">
SetHandler obwebgateerr
</Location>
<LocationMatch "/*">
AuthType Oblix
require valid-user
</LocationMatch>
#*** END Oblix NetPoint WebGate Specific ****
```

Notes for UNIX

When running Apache v2 on HP-UX, do not use nobody for User or Group, because shared memory may not work. Instead, use your login name as User Name with a group Group as "Oblix" (or "www" as User Name and "others" as Group Name). On HP-UX, "www" is equivalent to "nobody" on Solaris.

When running Apache v2 on HPUX 11.11, ensure that the AcceptMutex directive in the Apache httpd.conf file is set to "fcntl". If the directive is not present, add it to the httpd.conf file (AcceptMutex fcntl). For more information, see http://issues.apache.org/bugzilla/show_bug.cgi?id=22484).

Notes for IHS on AIX

```
#*** BEGIN Oblix NetPoint WebGate Specific ****
   LoadModule obWebgateModule DR/oblix/apps/webgate/bin/webgate.so
   WebGateInstalldir DR
   WebGateMode PEER
   <Location "/oberr.cgi">
      SetHandler obwebgateerr
   </Location>
   <LocationMatch "/*">
      AuthType Oblix
         require valid-user
   </LocationMatch>
#*** END Oblix NetPoint WebGate Specific ****
```

1. Use the chmod -r username:groupname directory/file to change the User Name and Group Name of a directory or a file.

   When you do this, you need to change the User and Group parameters in the httpd.conf file accordingly.

2. See "Tuning Apache/IHS v2 for Oracle Access Manager WebGates" on page 18-28 for more information and complete any additional steps needed to finish the Oracle Access Manager implementation for Apache v2.

> **Important:** You use the following procedure only if you need to clear the httpd.conf file of WebGate-related changes, then complete the Apache v2 Web server configuration for the WebGate anew.

To start httpd.conf updates anew

1. Restore the original httpd.conf file to remove any Oracle Access Manager entries that are present.

2. Update the httpd.conf file for Oracle Access Manager using one of the following methods:

   - **Either** open the file *component_install_dir*/access/oblix/lang/LangTag/docs/config.htm and perform a manual configuration, as described in Chapter 17, "Managing OAM 10g WebGates with OAM 11g".

   - **Or** launch the ManageHttpConf program in *component_install_dir*/access/oblix/tools/setup/InstallTools/ManageHttpConf without any options to print instructions on its use.

   > **Note:** If the ManageHttpConf program is run with WebGate entries already present in the httpd.conf file, an error message will be printed and the httpd.conf file will not be updated.

3. Complete activities in "Tuning Apache/IHS v2 for Oracle Access Manager WebGates" on page 18-28.

18.8.2 Verifying Language Encoding

As mentioned earlier, Oracle Access Manager HTML pages use UTF-8 encoding. Apache-based Web servers allow administrators to specify a default character set for all HTML pages sent out using the `AddDefaultCharset` directive, which overrides any character specified by the application generating the HTML pages. If the `AddDefaultCharset` directive enables a character set other than UTF-8, Oracle Access Manager HTML pages are garbled.

To ensure proper language encoding

1. Open the httpd.conf file.
2. Locate the `AddDefaultCharset` directive.
3. Complete one of the following activities to ensure that proper encoding of Oracle Access Manager HTML pages:
   - Either set the `AddDefaultCharset` directive to Off.
   - Or Comment out the `AddDefaultCharset` directive.
4. Save the httpd.conf file and restart the Web server.

18.9 Tuning Oracle HTTP Server for Oracle Access Manager WebGates

After installing the Oracle Access Manager Web component for Oracle HTTP Server, you need to complete the steps that follow.

As mentioned earlier, before installing Oracle Access Manager WebGates for Oracle HTTP Server, in the httpd.conf file you must change the user and group to match the user that is installing the component.

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

To tune Oracle HTTP Server for Oracle Access Manager WebGates

1. Shut down opmn, as you usually do.
2. Locate and open the opmn.xml file for editing. For example:

   $oracle_home/opmn/bin/opmn.xml

3. In the opmn.xml file, adjust items as follows:

```
<ias-component id="HTTP_Server">
<process-type id="HTTP_Server" module-id="OHS2">
         <environment>
                <variable id="TMP" value="/tmp"/>
                <variable id="LD_ASSUME_KERNEL" value="2.4.19"/>
         </environment>
          <module-data>
            <category id="start-parameters">
              <data id="start-mode" value="ssl-disabled"/>
            </category>
          </module-data>
          <process-set id="HTTP_Server" numprocs="1"/>
        </process-type>
      </ias-component>
```

4. Refresh the OPMN configuration by executing the following script:

   ```
   #oracle_home/opmn/bin/opmnctl reload
   ```

5. Start the Oracle HTTP Server Web server, as described in "Starting and Stopping Oracle HTTP Server Web Servers"

18.10 Tuning OHS /Apache Prefork and MPM Modules for OAM

Oracle recommends specific tuning parameters with Oracle Access Manager WebGates for these Web servers.

The tuning parameters described in this section are configured in the httpd.conf file with Apache v2.0 and OHS11g.

For Apache v2.2, however, tuning is configured in the following files:

apache_install_dir/conf/extra/httpd-mpm.conf

apache_install_dir/conf/extra/httpd-default.conf

Also for Apache v2.2, the entries for httpd-mpm.conf and httpd-default.conf should be uncommented, as follows:

From:

```
#Include conf/extra/httpd-mpm.conf
#Include conf/extra/httpd-default.conf
```

To:

```
Include conf/extra/httpd-mpm.conf
Include conf/extra/httpd-default.conf
```

Use the following topics as needed for your environment:

- Tuning Oracle HTTP Server /Apache Prefork Module
- Tuning Oracle HTTP Server /Apache MPM Module
- Kernal Parameters Tuning

18.10.1 Tuning Oracle HTTP Server /Apache Prefork Module

Oracle recommends the following as broad guidelines when using Oracle Access Manager with either the Oracle HTTP Server or Apache Prefork module:

Timeout 300

KeepAlive On

MaxKeepAliveRequests 500

KeepAliveTimeout 10

StartServers: 5 (Initial number of processes to start; used only on startup.)

MaxClients: 500 (Total number of processes to handle load at peak time. Determines how many child processes will be created to handle requests at peak period.

ServerLimit: 500 (The maximum configured value for MaxClients for the lifetime of the process. If MaxClients is set to a value higher than the default, ServerLimit value should be specified above the rest of the parameters.

MinSpareServers, MaxSpareServers: Default values should suffice requirements to handle a heavy load. During operation, these values regulate how the parent process creates children to serve requests.

MaxRequestsPerChild: 0 - Number of requests sent to each child process. 0 indicates the process never expires/dies

18.10.2 Tuning Oracle HTTP Server /Apache MPM Module

Oracle recommends the following as broad guidelines when using Oracle Access Manager with either the Oracle HTTP Server or Apache Prefork module:

Timeout 300

KeepAlive On

MaxKeepAliveRequests 500

KeepAliveTimeout 10

StartServers: 2 (Initial number of processes to start; used only on startup.)

MaxClients: 500 (Total number of processes to handle load at peak time. Determines how many child processes will be created to handle requests at peak period.

ServerLimit: 25 (The maximum configured value for MaxClients for the lifetime of the process. If MaxClients is set to a value higher than the default, ServerLimit value should be specified above the rest of the parameters.

MinSpareServers, MaxSpareServers: 25, 75. During operation, these values regulate how the parent process creates children to serve requests.

ThreadsPerChild: 25 (The number of worker threads in single httpd process.)

MaxRequestsPerChild: 0 (This directive sets the limit on the number of requests that an individual child server process will handle. The value 0 will ensure that the process never expires.)

18.10.3 Kernal Parameters Tuning

Oracle Recommends that you ensure that the kernal parameters for the soft and hard limit on the file descriptors are set to a high value. For example:

Hard limit (rlim_fd_max): 65535

Soft limit (rlim_fd_cur): 65535

The high value of the file descriptor is a strong recommendation for the Apache server that will open and close sockets for requests.

18.11 Starting and Stopping Oracle HTTP Server Web Servers

Starting and stopping an Oracle HTTP Server Web server is the same procedure for both v1.3 and v2, on all platforms.

To start the Oracle HTTP Server Web server

1. Locate and change to the following directory:

   $ORACLE_HOME\opmn\bin\

2. From the command line, enter the following command:

   ```
   opmnctl/startproc process-type=HTTP_Server
   ```

To stop the Oracle HTTP Server Web server

1. Locate and change to the following directory:

   $ORACLE_HOME\opmn\bin\

2. From the command line, enter the following command:

   ```
   opmnctl/stopproc process-type=HTTP_Server
   ```

18.12 Tuning Apache/IHS v2 for Oracle Access Manager WebGates

Unless explicitly stated, information here applies to both Apache and IHS v2 WebGate components (also known as plug-ins). For details about Oracle HTTP Server, see the *Oracle HTTP Server Administrator's Guide 10 g R2 (10.1.2)*.

Apache v2 bundled with Security-Enhanced Linux: With SELinux, errors could be reported in WebServer logs/console when starting a Web server on Linux distributions that have more strict SELinux policies in place after installing an Oracle Access Manager Web component. You can avoid these errors by running appropriate chcon commands for the installed Web component before restarting the Web server.

> **See Also:** "SELinux Issues" on page H-14

Apache v2 bundled SELinux-enabled Linux Distribution: Security-enhanced Linux (SELinux) is an automatically enabled implementation of a mandatory access-control mechanism. As described in your Linux documentation, SELinux policies provide access to certain pre-defined system directories such as /etc/httpd/conf, /usr/sbin/apachect, and /var/log/ (to name a few) for system daemons.

When Oracle Access Manager WebGates are installed with the bundled Apache Web server, certain policies must be added to allow Apache processes to access Oracle Access Manager installation files.

The bundled Apache Web server runs as user "apache" with a security context defined as context=user_u:system_r:unconfined_t. As a result, when Oracle Access Manager WebGates are installed in any of the user folders, the Apache Web server will not start.

The $SELINUX_SRC variable represents the SELinux policy source directory. The default value is /etc/selinux/targeted/src/policy. However, your environment may vary. Be sure to consult your system administrator for the actual value for your system.

To add Oracle Access Manager policies to Apache bundled with Red Hat Enterprise Linux 4

1. After installing each Oracle Access Manager Web component, log in as the 'root' user.

2. Ensure that all Oracle Access Manager WebGates are installed for Web server user & group (default: apache).

3. Create an oracle_access_manager.te policy file in the *$SELINUX_SRC*/domains/programs/directory and add the following rules:

   ```
   type oracle_access_manager_t, file_type, sysadmfile;
   allow httpd_t oracle_access_manager_t:file { rw_file_perms create rename
   link unlink setattr execute };
   allow httpd_t oracle_access_manager_t:dir  { rw_dir_perms create append
   rename link unlink setattr };
   ```

4. Create an oracle_access_manager.fc file context in the directory *$SELINUX_SRC*/file_contexts/program, then register the Oracle Access Manager Web component installation directory (without identity or access suffix). For example:

```
Oracle_Access_Manager_install_dir(/.*)? system_u:object_r:oracle_access_
manager_t
```

> **Note:** When the WebGate is installed in a separate directory from the Access Manager, be sure to register the WebGate installation directory separately.

5. Compile and deploy the policy files as follows:

```
cd $SELINUX_SRC
make load
Label Oracle Access Manager files
run restorecon -R Oracle_Access_Manager_install_dir (without the identity or
access suffix)
```

Apache v2 Directives: Apache 1.3 uses a process model for serving multiple HTTP requests at once. This differs from the single process (thread) model employed by other Web servers, which manage several requests simultaneously in one process.

> **Note:** Only the prefork MPM in Apache v2 uses the same process model for serving HTTP requests as Apache v1.3. For all other MPMs, Apache v2 uses a hybrid process-thread model.

Several directives in the Apache v2 Web server configuration file (httpd.conf) affect how the Apache Web server decides to create or destroy worker processes. The following parameters affect the performance of the Apache v2 Web server:

- **ThreadsPerChild**: This directive sets the number of threads created by each child process. The child creates these threads at startup and never creates more.
  - If you are using an MPM like mpm_winnt, where there is only one child process, this number should be high enough to handle the entire load of the server.
  - If you are using an MPM like mpm_worker, where there are multiple child processes, the total number of threads should be high enough to handle the common load on the server.
- **MinSpareThreads**: This value is only used with mpm_worker. Since Oracle Access Manager plug-in initialization is deferred until the first request, there is minimal advantage of keeping high value for this directive. However, it is useful to keep this parameter as high as possible.
- **MaxSpareThreads**: This value is only used with mpm_worker. The value for MaxSpareThreads must be greater than or equal to the sum of MinSpareThreads and ThreadsPerChild or the Apache HTTP Server automatically corrects it.

  Recommendation: Keep the value high. For a dedicated server this will not be a problem.
- **MaxSpareServers**: With Apache v2, this is used only with the prefork MPM model. To preserve as much state as possible in the server, set the MaxSpareServers to a high value. Setting this value to the maximum of 255 keeps

all Apache worker-processes available indefinitely, but it does not provide an opportunity for worker-process recycling during low-load periods.

- **MinSpareServers**: With Apache v2, this is used only with the prefork MPM model. Since Oracle Access Manager plug-in initialization is deferred until the first request, using a high value for the MinSpareServers parameter provides minimal advantage. However, it is useful to keep this parameter as high as possible. For dedicated Web server systems, this should pose no great burden.
- **MaxClients**: With IHS v2 and the worker MPM, MaxClients restricts the total number of threads that will be available to serve clients. For hybrid MPMs, the default value is 16 (ServerLimit) multiplied by a value of 25 (ThreadsPerChild). To increase MaxClients to a value that requires more than 16 processes, you must also raise ServerLimit.

Appropriate values for the preceding parameters depend on the expected load and the performance class of the systems involved, including the Access Server and LDAP server.

Apache servers on very high performance systems with high expected loads may be recompiled with a larger limit on the number of worker processes. These systems may see a greater performance impact on the StartServers and MinSpareServers parameters for dealing with sudden load spikes.

You may need to adjust operating system limits for the Access Server for proper operation. In particular, the maximum number of file descriptors available for any one Access Server may need to be increased beyond the default value. Configuring more than one connection between each Apache-based WebGate and an Access Server may quickly exceed this limit.

For additional information, see your Apache documentation.

18.13 Removing Web Server Configuration Changes After Uninstall

Web server configuration changes that occur during installation must be manually removed after uninstalling the WebGate). This type of information must be removed manually.

Further, you must remove any changes that you manually made to your Web server configuration file for the WebGate) should be removed. For more information about what is added for each component, look elsewhere in this chapter.

18.14 Helpful Information

Consult the following manual for more information about the Oracle HTTP Server:

Oracle HTTP Server Administrator's Guide 10 g R2 (10.1.2)

The following URLs provide information about building an Apache release and source code:

Apache v2 documentation:

`http://httpd.apache.org/docs-2.0/`

Apache v2 source code:

`http://httpd.apache.org/download.cgi`

Mod-SSL documentation:

`http://httpd.apache.org/docs-2.0/mod/mod_ssl.html`

OpenSSL documentation:

http://www.openssl.org/docs/

OpenSSL source code:

http://www.openssl.org/source/

Compiling and Installing Apache v2:

http://httpd.apache.org/docs-2.0/install.html#test

IHS:

http://www-306.ibm.com/software/webservers/httpservers/doc/v2047/manual/readme.html

19 Configuring the IIS Web Server for 10g WebGates

This chapter summarizes activities that you need to perform to configure 10.1.4 WebGate with a Microsoft Internet Information Server (IIS Web server for Windows environments). Unless explicitly stated, information and steps in this chapter apply equally to 32-bit and 64-bit WebGate installations. Topics include:

- Prerequisites
- WebGate Guidelines for IIS Web Servers
- Prerequisite for Installing WebGate for IIS 7
- Updating IIS 7 Web Server Configuration on Windows 2008
- Completing WebGate Installation with IIS
- Installing and Configuring Multiple 10g WebGates for a Single IIS 7 Instance
- Installing and Configuring Multiple WebGates for a Single IIS 6 Instance
- Finishing 64-bit WebGate Installation
- Confirming WebGate Installation on IIS
- Starting, Stopping, and Restarting the IIS Web Server
- Removing Web Server Configuration Changes Before Uninstall

19.1 Prerequisites

Ensure that your OAM 11g Administration Console is running and get familiar with:

- Introduction to Policy Enforcement Agents on page 5-1
- About Installing Fresh OAM 10g WebGates to Use With OAM 11g on page 17-2

19.2 WebGate Guidelines for IIS Web Servers

ISAPI is an Internet Web server extension that the WebGate that communicates with the IIS Web server. For example, you will need the following package to install the Oracle Access Manager WebGates for IIS:

Oracle_Access_Manager10_1_4_3_0_Win32_*ISAPI*_WebGate

64-bit WebGate: Oracle_Access_Manager10_1_4_3_0_Win64_ISAPI_WebGate.exe

Updating the IIS Web server configuration file is required when installing Oracle Access Manager WebGates. With IIS Web servers, a configuration update involves updating the Web server directly by adding the ISAPI filter and creating extensions required by Oracle Access Manager. A filter listens to all requests to the site on which it is installed. Filters can examine and modify both incoming and outgoing streams of data to enhance IIS functionality. ISAPI extensions are implemented as DLLs that are loaded into a process that is controlled by IIS. Like ASP and HTML pages, IIS uses the virtual location of the DLL file in the file system to map the ISAPI extension into the URL namespace that is served by IIS.

Oracle recommends that you update the IIS Web server configuration file automatically during Oracle Access Manager Web component installation. Automatic updates may take more than a minute. However, updating the IIS Web server configuration file manually takes longer and could introduce unintended errors.

For more specific guidelines, see:

- Guidelines for ISAPI WebGates
- Prerequisite for Installing Any 10g WebGate for IIS 7
- Prerequisite for Installing a 32-bit WebGate for IIS 7

19.2.1 Guidelines for ISAPI WebGates

General WebGate preparation and installation details apply to ISAPI WebGates. Additionally, this topic provides specific guidelines for ISAPI WebGates installed with an IIS Web server. You can install multiple WebGates with a single IIS Web server instance or you might have a 64-bit WebGate.

Note: Unless explicitly stated, details apply equally to 32-bit and 64-bit WebGates.

lockdown Mode: Before installing the WebGate, ensure that your IIS Web server is *not* in lockdown mode. Otherwise things will appear to be working until the server is rebooted and the metabase re-initialized, at which time IIS will disregard activity that occurred after the lockdown.

Permissions: Setting various permissions for the /access directory is required for IIS WebGates only when you are installing on a file system that supports NTFS. For example, suppose you install the ISAPI WebGate in Simple or Cert mode on a Windows 2000 computer running the FAT32 file system. The last installation panel provides instructions for manually setting various permissions that cannot be set on the FAT32 file system. In this case, these instructions may be ignored.

Virtual Hosts: Each IIS Virtual Web server can have it's own WebGate.dll file installed at the virtual level, or can have one WebGate affecting all sites installed at the site level. Either install the WebGate.dll at the site level to control all virtual hosts or install the WebGate.dll for one or all virtual hosts.

postgate.dll: You may also need to install the postgate.dll file at the computer level. The postgate.dll is located in the *WebGate_install_dir*, as described in "Installing the Postgate ISAPI Filter". If you perform multiple installations, multiple versions of this file may be created which may cause unusual Oracle Access Manager behavior. In this case, you should verify that only one webgate.dll and one postgate.dll exist.

> **Note:** The postgate.dll is always installed at the site level. If for some reason the WebGate is reinstalled, the postgate.dll is also reinstalled. In this case, ensure that only one copy of the postgate.dll exists at the site level.

Updating Web Server Configuration for WebGate: As with other Oracle Access Manager WebGates, your Web server must be configured to operate with the WebGate. Oracle recommends automatically updating your Web server configuration during installation. However, you can decline the automatic update and instead manually configure your Web server as described in "Provisioning a 10g WebGate with OAM 11g" on page 17-4.

FAT32 file system: You may receive special instructions to perform during WebGate installation. For example: Setting various permissions for the /access directory is required for IIS WebGates only when you are installing on a file system that supports NTFS. The last installation panel provides instructions for manually setting various permissions that cannot be set on the FAT32 file system. In this case, these instructions can be ignored.

SSL and Client Certificate Authentication: On IIS, if you are using client certificate authentication you must enable SSL on the IIS Web server hosting the WebGate before enabling client certificates for WebGate. You must also ensure that various filters are installed in a particular order. In addition, you may need to install the postgate.dll as an ISAPI filter.

Web Server Releases: Web server details in this chapter apply to the stated release. If the release is not stated, you can presume it is IIS v5. Details specific to IIS v6 or IIS v7 are identified.

See Also:

- WebGates for IIS v7 on page 19-4
- WebGates for IIS v6 on page 19-4

32-bit versus 64-bit WebGates: Unless explicitly stated, all information applies equally to both 32-bit and 64-bit WebGates.

See Also:

- WebGates for IIS v6 on page 19-4
- Finishing 64-bit WebGate Installation on page 19-24

General WebGate Preparation and Installation Details: Refer to this chapter for IIS-specific guidelines. Refer to Chapter 17 for general preparation and installation details.

Completing and Confirming WebGate Installation: Perform tasks relevant to your ISAPI WebGate and IIS version:

See Also:

- Completing WebGate Installation with IIS
- Finishing 64-bit WebGate Installation
- Confirming WebGate Installation on IIS

19.2.1.1 WebGates for IIS v7

General guidelines and WebGate installation are usually the same regardless of the IIS release for which you are installing a WebGate. However, there are several specific topics to review when you are installing one or more WebGates for IIS v7:

- Prerequisite for Installing WebGate for IIS 7 on page 19-5
- Updating IIS 7 Web Server Configuration on Windows 2008 on page 19-6
- Installing and Configuring Multiple 10g WebGates for a Single IIS 7 Instance on page 19-14

19.2.1.2 WebGates for IIS v6

General guidelines and WebGate installation are usually the same regardless of the IIS release for which you are installing a WebGate. However, there are several specific topics of interest.

Multiple WebGates with a Single IIS 6 Instance: IIS v6.0 supports hosting multiple Web sites on a single Web server instance and Oracle Access Manager ISAPI WebGate allows you to protect each Web site with a different WebGate.

> **See Also:** Multiple WebGates with a Single IIS 6 Instance

64-bit IIS v6 WebGate: Perform installation as you do for all others, using instructions available in Chapter 17. If you choose manual Web server configuration during WebGate installation, you can access details in the following path:

WebGate_install_dir\access\oblix\lang\en-us\docs\dotnet_isapi.htm

Following WebGate installation and IIS configuration, perform tasks in "Finishing 64-bit WebGate Installation" on page 19-24.

Earlier Release WebGate Installations: Previously Oracle recommended that WebGate be installed in the same physical directory location as Policy Manager. This required a virtual directory named “access” for both Policy Manager and WebGate, which is mapped to the physical location of both Policy Manager and WebGate.

> **Note:** You can install WebGate 10*g* (10.1.4.3) for IIS in any location, separate from that of Policy Manager.

If you have an earlier, combined WebGate and Policy Manager installation, you can de-couple the components using the following steps.

To de-couple an earlier WebGate/Policy Manager installation

1. Uninstall any patches applied to the earlier WebGate and Policy Manager, if any.
2. Uninstall the earlier Policy Manager and WebGate combination.
3. Install Policy Manager 10*g* (10.1.4.3).
4. In a separate directory location, install WebGate 10*g* (10.1.4.3)

19.2.1.3 Multiple WebGates with a Single IIS 6 Instance

Unless explicitly stated, details in this topic apply equally to 32-bit and 64-bit WebGates.

IIS v6.0 supports hosting multiple Web sites on a single Web server and Oracle Access Manager ISAPI WebGate allows you to protect each Web site with a different WebGate.

Note: Previous ISAPI WebGate releases did not support multiple WebGates with a single IIS Web server instance. You either had to install one WebGate for all Web sites at the top level, or protect a single Web site by configuring WebGate at the Web site level.

IIS 6 provides application pools that are used to run virtual servers. You can think of an application pool as a group of one or more URLs that are served by a worker process or a set of worker processes. An application pool is a configuration that links one or more applications to a set of one or more worker processes. Because applications in this pool are separated from other applications by worker process boundaries, an application in one application pool is not affected by problems caused by applications in other application pools. Today, WebGate instances can run in different process spaces.

When you have multiple Web sites on a single IIS v6.0 Web server instance, you need to ensure that user requests reach the correct Web site. To do this, you need to configure a unique identity for each site on the server using at least one of three unique identifiers:

- Host header name
- IP address
- TCP port number

Note: If you have multiple Web sites on a single server and these are distinguished by IP address and port, multiple WebGates are not required. Starting with release 10.1.4.2.0 virtual hosts on Apache and IIS 6.0 are supported. As a result, a single WebGate on the top level can protect all the Web sites even if the IP addresses are different. This is handled by using different Host Identifiers for each Web site.

You can install multiple WebGates on different Web sites of the same IIS Web server instance. However, several manual steps are required.

See Also: "Installing and Configuring Multiple WebGates for a Single IIS 6 Instance" on page 19-19

19.3 Prerequisite for Installing WebGate for IIS 7

This section provides prerequisites for installing WebGates with IIS v7 Web servers. It includes the following topics:

- Prerequisite for Installing Any 10g WebGate for IIS 7
- Prerequisite for Installing a 32-bit WebGate for IIS 7

19.3.1 Prerequisite for Installing Any 10g WebGate for IIS 7

The following procedure applies to 32-bit and 64-bit WebGates equally.

With WebGate for IIS v7 Web Server, you can use Form-based authentication without enabling pass through functionality only when the **`<add segment="bin"/>`** entry is not present in the applicationHost.config file. For example, if you have access/oblix/apps/webgate/bin/webgate.dll as an action in the Form-based authentication scheme, ensure that the **`<add segment="bin"/>`** entry is not present in the applicationHost.config file. If the entry is present, you must remove it, as described next

To locate and remove the `<add segment="bin"/>` entry

1. Go to Windows\System32\inetsrv\config and open the applicationHost.config file.
2. Search for the `<hiddenSegments>` module.
3. Remove the entry **`<add segment="bin"/>`** if it is present.
4. Save the file.

19.3.2 Prerequisite for Installing a 32-bit WebGate for IIS 7

The following procedure applies to 32-bit WebGates only.

The following procedure provides steps to configure a 32-bit WebGate for IIS 7 Web Server to use either Simple or Cert transport security mode. This configuration requires that the IIS 6 Management Compatibility module be installed.

To add the IIS 6 Management Compatibility module for a 32-bit WebGate for IIS 7 and Simple or Cert security

1. From the State menu, click Administrative Tools, and then click Server Manager.
2. In the Server Manager tree, expand Roles, and then click Web Server (IIS).
3. In the Web Server (IIS) pane, Role Services section, click Add Role Services.
4. On the Select Role Services page of the Add Role Services Wizard, click IIS6 Management Compatibility under Management Tools.
5. On the Confirm Installation Selections page, click Install.
6. On the Results page, click Close.

19.4 Updating IIS 7 Web Server Configuration on Windows 2008

You can display these steps when you decline automatic Web server updates during Oracle Access Manager WebGate installation.

To display steps to configure IIS 7 Web server on Windows 2008 for ISAPI WebGates

1. When installing WebGate, click No when asked if you want the automatic Web server update and:
   a. Read information on a new screen to assist in manually setting up your Web server for the WebGate.
   b. Click the following item in the table that appears perform the steps that are displayed.

Table 19–1 IIS 7 WebGate Windows Server 2008

Supported Server OS	Microsoft IIS
Windows Server 2008	ISAPI
...	...

2. After performing steps to update the IIS 7 Web server on Windows 2008, return to the WebGate installation screen and click Next, as described in the chapter on WebGate installation.
3. Proceed with "Completing WebGate Installation with IIS".

19.5 Completing WebGate Installation with IIS

Unless explicitly stated, details in this topic apply equally to 32-bit and 64-bit WebGates.

> **See Also:**
>
> As needed, see:
>
> - Finishing 64-bit WebGate Installation on page 19-24
> - Installing and Configuring Multiple WebGates for a Single IIS 6 Instance on page 19-19
>
> If you have IIS v7, Oracle recommends the following topics:
>
> - Updating IIS 7 Web Server Configuration on Windows 2008 on page 19-6
> - Installing and Configuring Multiple 10g WebGates for a Single IIS 7 Instance on page 19-14

Completing WebGate installation with an IIS Web server, includes the following activities after the installation is complete.

Task overview: Completing IIS WebGate installations includes

1. Enabling Client Certificate Authentication on the IIS Web Server on page 19-7
2. Ordering the ISAPI Filters on page 19-8
3. Enabling Pass-Through Functionality for POST Data on page 19-9
4. Protecting a Web Site When the Default Site is Not Setup on page 19-13

19.5.1 Enabling Client Certificate Authentication on the IIS Web Server

Unless explicitly stated, details in this topic apply equally to 32-bit and 64-bit WebGates.

If you are using client certificate authentication, you must enable SSL on the IIS Web server. If you select client certificate authentication during setup, you must also add the cert_authn.dll as one of the ISAPI filters.

> **Note:** The procedures here reflect the sequence for IIS v5. Your environment might be different.

To enable SSL on the IIS Web server

1. Start the Internet Information Services console, if needed: Click Start, Programs, Administrative Tools, Internet Information Services.
2. Expand the local computer to display your Web Sites.

3. Expand the Default Web Site (or the appropriate Web site), then expand \access\oblix\apps\webgate\bin.
4. Right click cert_authn.dll and select Properties.
5. In the Properties panel, select the File Security tab.
6. In the Secure Communications sub-panel, click Edit.
7. In the Client Certificate Authentication sub-panel, click Accept Certificates and click OK.
8. Click OK in the cert_authn.dll Properties panel.
9. Proceed to the next procedure: "To add cert_authn.dll as an ISAPI filter".

To add cert_authn.dll as an ISAPI filter

1. Start the Internet Information Services console, if needed: Click Start, Programs, Administrative Tools, Internet Information Services.
2. Expand the local computer to display your Web Sites.
3. Right click the appropriate Web Site to display the Properties panel.
4. Click the ISAPI Filters tab, then click the Add button to display the Filter Properties panel.
5. Enter filter name "cert_authn".
6. Click the Browse button and navigate to the following directory:

   \WebGate_install_dir\access\oblix\apps\webgate\bin
7. Select cert_authn.dll as the executable.
8. Click OK on the Filter Properties panel.
9. Click Apply on the ISAPI Filters panel.
10. Click OK.
11. Ensure the filters are listed in the correct order.

19.5.2 Ordering the ISAPI Filters

Unless explicitly stated, details in this topic apply equally to 32-bit and 64-bit WebGates.

It is important to ensure that the WebGate ISAPI filters are included in the right order.

> **Note:** This task is the same whether you are installing one or more WebGates per IIS Web server instance.

To order the WebGate ISAPI filters

1. Start the Internet Information Services console, if needed: Click Start, Programs, Administrative Tools, Internet Information Services.
2. Expand the local computer to display your Web Sites.
3. Right-click the Web Site and select Properties.
4. Click Properties, select ISAPI filters.
5. Confirm the following .dll files appear.

For example:

cert_authn.dll
webgate.dll

6. Add any missing filters, if needed, then select a filter name and use the up and down arrows to arrange the filter order as shown in step 5.

> **WARNING: Confirm that there is only one webgate.dll and one postgate.dll filter. If you perform multiple WebGate installations on one computer, multiple versions of the postgate.dll file might be created and cause unusual Oracle Access Manager behavior.**

19.5.3 Enabling Pass-Through Functionality for POST Data

This section describes how the WebGate can be set up in conjunction with IIS 6.0 Worker Process Isolation Mode. It also covers configuration steps required for IIS 6.0 running in IIS 5.0 Isolation Mode.

> **Note:** This section supersedes information in "Installing Postgate.dll on IIS Web Servers" in the *Oracle Access Manager Installation Guide*. For the IIS 5.0 Web server, the existing functionality using postgate.dll continues to be supported.

Topics here include:

- About ISAPI WebGate 10.1.4.2.3
- About Pass-Through Functionality for POST Data
- Implementing Pass-Through: IIS 6.0 in Worker Process Isolation Mode
- Implementing Pass-Through with IIS 6.0 Web Server in IIS 5.0 Isolation Mode

19.5.3.1 About ISAPI WebGate 10.1.4.2.3

Starting with ISAPI WebGate release 10.1.4.2.3, Oracle Access Manager pass-through functionality is supported with IIS 6.0 running in a Worker Process Isolation Mode. ISAPI WebGate 10.1.4.2.3 also operates with IIS 6.0 running in IIS 5.0 Isolation Mode using postgate.dll.

> **Note:** Oracle recommends using Worker Process Isolation Mode for new or existing implementations. Worker Process Isolation Mode is a default setting for the IIS 6.0 Web server. For the IIS 5.0 Web server, the existing functionality (using postgate.dll) continues to be supported.

This section describes how to set up ISAPI WebGate release 10.1.4.2.3 in conjunction with IIS 6.0 Worker Process Isolation Mode. It also provides configuration steps required for IIS 6.0 running in IIS 5.0 Isolation Mode. This section supersedes information in Section 19-6 (Installing Postgate.dll on IIS Web Servers) of the *Oracle Access Manager Installation Guide*.

19.5.3.2 About Pass-Through Functionality for POST Data

POST data is required for pass through during a form login on the IIS Web server when using the WebGate extension method (where the WebGate is the action of the form). In other words, if a form authentication scheme on the IIS Web server is configured with the pass-through option, and the target of the login form requires the data posted by the form, the WebGate extension method (where the WebGate DLL is the action of the form) cannot be used. The WebGate filter method (where the action of the form is a protected URL that is not the WebGate DLL) must be used instead, and based on IIS version, the postgate.dll must be installed or configure webgate.dll as ISAPI extension.

IIS 6.0 in Worker Process Isolation Mode: webgate.dll must be configured as an ISAPI filter and also as an ISAPI extension to achieve pass-through functionality. (This does not apply to ISA server integration.) Pass-through functionality is supported with 10.1.4.2.3 and higher ISAPI WebGates. However, you must also set a new user-defined parameter `"UseWebGateExtForPassthrough"` to `true` in the WebGate configuration profile in the Access System Console.

IIS 5.0 or IIS6.0 running in IIS 5.0 Isolation Mode: postgate.dll must be configured as an ISAPI filter to achieve the pass-through functionality.

19.5.3.3 Implementing Pass-Through: IIS 6.0 in Worker Process Isolation Mode

The following steps outline this task.

Task overview: Implementing Pass-Through Functionality with IIS 6.0 Web Server in Worker Process Isolation Mode

1. Install WebGate as described in "Locating and Installing the Latest OAM 10g WebGate for OAM 11g" on page 17-6.
2. Set the pass-through parameter as described in "Setting the UseWebGateExtForPassthrough Parameter in the WebGate Profile".
3. Configure webgate.dll as described in "Configuring webgate.dll as an ISAPI Extension".

19.5.3.3.1 Setting the UseWebGateExtForPassthrough Parameter in the WebGate Profile You must set the new user-defined parameter, `UseWebGateExtForPassthrough`, in the WebGate profile to implement pass-through functionality with the IIS 6.0 Web server in Worker Process Isolation Mode. You must set `UseWebGateExtForPassthrough` to `true`. If this parameter is set to `false`, pass-through functionality will not work.

> **See Also:** "IIS Web Server Issues" on page H-10

To set the UseWebGateExtForPassthrough Parameter in the WebGate Profile

1. Launch the Access System Console and click Access System Configuration.
2. Click AccessGate Configuration.
3. Enter your search criteria for the WebGate, and then click Go.
4. In the Search Results table, click a WebGate name.
5. At the bottom of the Details for AccessGate page, click Modify.
6. On the Modify AccessGate page, locate the User Defined Parameters section of the page, enter the following parameter, and value, and then click the Add button:

   Parameter: `UseWebGateExtForPassthrough`

   Value: `true`
7. Click the Add button if you want to add more user-defined parameters.
8. Save to save this new information.

9. Repeat for each WebGate in your deployment.

10. Proceed to "Configuring webgate.dll as an ISAPI Extension".

19.5.3.3.2 Configuring webgate.dll as an ISAPI Extension

The webgate.dll is part of the WebGate installation. The following procedure describes how to configure webgate.dll as an ISAPI extension. This task must also be performed to implement pass-through functionality with IIS 6.0 Web Server in Worker Process Isolation Mode.

> **Note:** You can have multiple webgate.dlls configured at different website levels from the top level Web Sites. In this case, you also need to configure webgate.dll as an ISAPI extension for each website protected by WebGate.

To configure webgate.dll as an ISAPI extension

1. Go to websites, right click, and select Properties.

2. In the Properties dialog box, select the Home Directory tab.

3. Click the Configurations button to open the Application Configurations dialog box.

4. In Wild Card Application Maps, click the Inset button.

5. Provide the path to webgate.dll. For example:

   ```
   WebGate_install_dir/access/oblix/apps/webgate/bin/webgate.dll
   ```

6. Uncheck the "verify that file exists" box.

7. Confirm and finalize the changes: click OK, then click OK again; click Apply, and then click OK.

8. Stop the IIS Administration Server from Services and restart the IIS Web server.

19.5.3.4 Implementing Pass-Through with IIS 6.0 Web Server in IIS 5.0 Isolation Mode

The following steps outline this task.

> **Note:** Skip this task if you are using IIS 6.0 Web server in Worker Process Isolation Mode.

Task overview: Implementing Pass-Through Functionality with IIS 6.0 Web Server in IIS 5.0 Isolation Mode

1. Install WebGate as described in the *Oracle Access Manager Installation Guide*.

2. Set up IIS 6.0 as described in "Setting Up IIS 6.0 Web Server in IIS 5.0 Isolation Mode" on page 19-11.

3. Install postgate.dll as described in "Installing the Postgate ISAPI Filter""Installing the Postgate ISAPI Filter".

19.5.3.4.1 Setting Up IIS 6.0 Web Server in IIS 5.0 Isolation Mode The following information is updated for the 10.1.4.2.3 WebGate.

When IIS 6.0 Web server is used, the following steps outline how to set up the WWW Service to run in IIS 5.0 Isolation Mode. This is required by the ISAPI postgate filter.

To set IIS 5.0 isolation on IIS 6 Web servers

1. Start the Internet Information Services console, if needed: Click Start, Programs, Administrative Tools, Internet Information Services.
2. Expand the local computer to display your Web Sites.
3. Right-click the Web Site and select Properties.
4. Select the Service tab in the Web Site Properties window.
5. Check the box beside Run WWW service in IIS 5.0 Isolation Mode.
6. Click OK.
7. Proceed with "Installing the Postgate ISAPI Filter".

19.5.3.4.2 Installing the Postgate ISAPI Filter

The following information is updated for the 10.1.4.2.3 WebGate.

For single WebGate installations, you should install the filters in the following order:

- The ISAPI WebGate filter should be installed after the sspifilt filter and before any others.
- The postgate filter should be installed before the WebGate filter, only if needed.
- All other Oracle Access Manager filters can be installed at the end.

> **Note:** Before installation (or after uninstallation) the filters must be removed manually. If multiple copies of a filter are installed, this means that they were not manually removed before installing the new filters.

You can have multiple webgate.dlls configured at different levels from the top level Web Sites. However, they share the same postgate.dll. If you perform multiple WebGate installations on one computer, multiple versions of the postgate.dll file can be created which might cause unusual Oracle Access Manager behavior. There can only be one postgate.dll configured at the (top) Web Sites level of a computer

> **Note:** postgate.dll is not supported when you have more than one WebGate installed and configured for a single IIS Web server instance.

The following procedures guide as you install and position the postgate ISAPI filter when you have a single WebGate installed with a single IIS Web server instance.

To install the postgate ISAPI filter

1. Start the Internet Information Services console, if needed: Click Start, Programs, Administrative Tools, Internet Information Services.
2. Expand the local computer to display your Web Sites.
3. Right-click the Web Site and select Properties.
4. Select the ISAPI Filters tab in the Web Site Properties window.

5. Click the Add button to display the Filter Properties panel.
6. Enter the filter name "postgate".
7. Click the Browse button and navigate to the following directory:

   \WebGate_install_dir\access\oblix\apps\webgate\bin
8. Select postgate.dll as the executable.
9. Click OK on the Filter Properties panel.
10. Click Apply on the ISAPI Filters panel.
11. Reposition the postgate ISAPI filter, as follows:

    a. Start the Internet Information Services console, if needed.

    b. Right-click your local computer, then select All Tasks, select Restart IIS.

    c. Select the ISAPI Filters tab on the Properties panel.

    d. Select the postgate filter and move it before WebGate, using the up arrow.

    For example:

    postgate.dll
    webgate.dll

    e. Restart IIS.

> **Note:** Consider using `net stop iisadmin` and `net start w3svc` to help ensure that the Metabase does not become corrupted.

19.5.4 Protecting a Web Site When the Default Site is Not Setup

Unless explicitly stated, this topic applies equally to 32-bit and 64-bit WebGates.

> **See Also:** "Setting Access Permissions, ISAPI filters, and Directory Security Authentication" on page 19-25

When you install a WebGate on an IIS Web server that does not have the "Default Web Site" configured, the installer does not create "Virtual Directory access", which must be done manually using the following procedure.

To protect a Web site (not the default site)

1. Start the Internet Information Services console, if needed
2. Select the name of the Web site to protect.
3. Right-click the name of the Web site to protect and select New, and then select Virtual Directory in the menu.
4. Click Next.
5. Select Alias: access, then click Next.
6. Directory: Enter the full path to the /access directory, then click Next.

   WebGate_install_dir\access
7. Select Read, Run Scripts, and Execute, then click Next.
8. Click Finish.
9. Restart IIS. For example:

Select Start, then Run.
Type `net start w3svc.`
Click OK.

19.6 Installing and Configuring Multiple 10g WebGates for a Single IIS 7 Instance

This section describes how to install and configure multiple WebGates for different Web sites on the same IIS 7 Web server instance. Several steps are manual and will differ from those that are performed when you install a single WebGate with a single IIS instance. When installing multiple WebGates for a single IIS instance:

- The webgate.dll must be configured as an ISAPI filter at the individual Web site level, not the default (top) Web server level
- The /access virtual directory is mapped at the Web site level to the respective /access directory in the WebGate installation.

When configuring the impersonation DLL for multiple WebGates, you need to configure a user to act as the operating system.

Task overview: Installing and configuring multiple WebGates for a single IIS 7 instance

1. Installing Each IIS 7 WebGate in a Multiple WebGate Scenario
2. Setting the Impersonation DLL for Multiple IIS 7 WebGates
3. Enabling Client Certification for Multiple IIS 7 WebGates
4. Configuring IIS 7 WebGates for Pass Through Functionality
5. Confirming IIS 7 WebGate Installation
6. Perform the following tasks, which are the same whether you install one or more WebGates per IIS Web server instance:
    - "Ordering the ISAPI Filters" on page 19-8
    - "Confirming WebGate Installation on IIS" on page 19-26

    See Also: "Confirming Multiple WebGate Installation" on page 19-24

19.6.1 Installing Each IIS 7 WebGate in a Multiple WebGate Scenario

After installing the ISAPI WebGate, there are several manual steps to perform as described here.

By default, webgate.dll is configured as an ISAPI filter at the host name (top) level. When installing multiple WebGates with a single IIS 7 instance, you need to remove the respective webgate.dll from the top level and configure it for the appropriate individual Web site after each WebGate installation.

To install each WebGate when you will have several with one IIS 7 instance

1. Install the ISAPI 7 WebGate as described in Chapter 17.
2. Go to the Web site to protect, and configure webgate.dll as the ISAPI filter using these steps:
    a. Start the Internet Information Services (IIS) Manager: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager.

b. Select the *hostname* from the Connections pane.

c. From the hostname Home pane, double-click ISAPI Filters, look for any WebGate.dll; if it is present, select it and click **Remove** from the Action pane.

d. In the Connection pane, under Sites, click the name of the Web Site for which you want to configure a WebGate filter.

e. In the Home pane, double-click ISAPI Filters.

f. In the Actions pane, click Add…

g. In the Filter name text box of the Add ISAPI Filter dialog box, type "WebGate" as name for the ISAPI filter.

h. In the Executable box, type the file system path of the WebGate ISAPI filter file or click the ellipsis button (...) to go to the folder that contains the WebGate.dll ISAPI filter file, and then click OK.

```
WebGate_install_dir\access\oblix\apps\webgate\bin\webgate.dll
```

3. Creating a Virtual Directory:

a. Expand the Sites pane and select the Web Site for which you just configured the ISAPI filter (WebGate.dll).

b. On the Action pane, click View Virtual Directories and then select **Add Virtual Directory.**

c. Specify **access** in the Alias text box and the physical path to the WebGate **access** folder of WebGate or click the ellipsis button (...) to go to the "access" folder, and then click OK.

```
WebGate_install_dir\access\
```

d. Save and apply these changes.

4. Setting permissions to the Virtual Directory:

a. Select the "access" virtual directory created in Step 3.

b. From the access Home pane, double click Handler Mappings; from the Action pane, select Edit Feature Permissions….

c. Check boxes beside Read, Script, and Execute, then click OK.

5. Setting Directory Permissions for WebGate:

a. In Explorer, right click the WebGate installation directory `WebGate_install_dir\access` and select Properties.

b. Click the Security tab and click the Edit button.

c. Add user "IUSR", select "Allow" for "Modify".

d. Add user "IIS_IUSRS", select "Allow" for "Modify".

e. Add user "NETWORK", select "Allow" for "Modify".

f. Add user "NETWORK SERVICE", select "Allow" for "Modify".

g. For group "Administrators" select "Allow" for "Modify".

6. WebGate in Simple or Cert Mode:

a. In the file system, locate and right-click the "password.xml" file in WebGate_install_dir\access\oblix\config\password.xml, and select Properties.

b. Click the Security tab.

  c. Give "Allow" for "Read" rights to users "IUSR", "NETWORK SERVICE", "IIS_WPG", "IIS_IUSRS".

7. Ensure that there is no webgate.dll in the top level (the hostname level).
8. Perform the next set of tasks using instructions in the following topics:
   a. "Setting the Impersonation DLL for Multiple IIS 7 WebGates" on page 19-16
   b. "Enabling Client Certification for Multiple IIS 7 WebGates" on page 19-17
9. Repeat these steps when you install the next WebGate for the IIS instance.

19.6.2 Setting the Impersonation DLL for Multiple IIS 7 WebGates

The client's access token is known as an impersonation token. The impersonation token identifies the client, the client's groups, and the client's privileges. The information in the token is used during access checks when the thread requests access to resources on the client's behalf.

The Access System authenticates and authorizes the user. IISImpersonationExtension.dll of Oracle Access Manager in the wildcard extension behaves like a filter for each request to the Web server. The Access System designates a special user that does have the right to impersonate another user by configuring it using the impersonation username/password on the AccessGate Configuration page. That designated user must have "act as operating system" rights. DLL impersonates the user authenticated and authorized by Oracle Access Manager and generates the impersonation token.

You perform the following steps to set the impersonation DLL for each WebGate that protects a Web site for a single IIS 7 Web server instance. You can do this either immediately after the installation task in the previous topic or all at one time.

> **Note:** This task must be performed for each WebGate that protects an individual Web site for a single IIS Web server instance.

To add the impersonation DLL to IIS 7 configuration for individual Web sites

1. Start the Internet Information Services (IIS) Manager, if needed: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager.
2. Add "IISImpersonationExtension.dll" as a Wildcard Script Map to the required Web Site:
   a. Expand Sites in the connection pane.
   b. Click the Web Site name to which you want to add IISImpersonationExtension.dll.
   c. Double click Handler Mappings from the selected Web Site's "home" pane.
   d. From the Action pane, click Add Wildcard Script Map.
   e. In the Name text box of the Add Wildcard Script Map dialog box, type "Oracle Impersonation Plugin" as name for the dll.
   f. In the Executable box, type the file system path of the WebGate IISImpersonationExtension.dll or click the ellipsis button (...) to go to the folder that contains IISImpersonationExtension.dll, and then click OK.

```
WebGate_install_dir/access/oblix/apps/WebGate/bin/
IISImpersonationExtension.dll
```

This example shows the default path, where *WebGate_install_dir* is the file system directory where you have installed this particular WebGate.

3. Proceed as follows:
   - **Client Certificate Authentication**: "Enabling Client Certification for Multiple IIS 7 WebGates"
   - "Confirming IIS 7 WebGate Installation" on page 19-19.

19.6.3 Enabling Client Certification for Multiple IIS 7 WebGates

You perform this task to set the enable client certification for each WebGate that protects a Web site for a single IIS 7 Web server instance. You can do this either immediately after the adding the impersonation DLL to an individual Web site or all at one time.

> **Note:** SSL should be enabled on the Web Site before configuring the client certification for WebGate. Follow these steps after the Web Site is SSL enabled.

If you select client certificate authentication during setup, you must also enable and then add the cert_authn.dll as one of the ISAPI filters in the respective Web site.

To enable cert_authn.dll on the IIS 7 Web server

1. Start the Internet Information Services (IIS) Manager, if needed: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager.
2. Expand Sites in the connection pane.
3. Expand the Web Site to \access\oblix\apps\webgate\bin.
4. Right click the "bin" directory and select Switch To Content View.
5. Right click the "cert_authn.dll".and from the drop down menu, select Switch To Feature View.
6. From the cert_authn.dll Home pane, double click SSL Settings.
7. From SSL Settings pane, select Require SSL check-box and select Accept from Client Certificates.
8. Select Apply from Action pane.
9. Repeat for each WebGate installed on this host, for which you want to enable client certification.
10. Restart the IIS 7 Web server.
11. Proceed to the next task: "To add cert_authn.dll as an ISAPI v7 filter".

To add cert_authn.dll as an ISAPI v7 filter

1. Start the Internet Information Services (IIS) Manager, if needed: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager.
2. Expand Sites in the connection pane.
3. Click on the Web Site name for which you want to add "cert_authn.dll".
4. In the Home pane, double-click ISAPI Filters.
5. In the Actions pane, click Add.

6. In the Filter name box of the Add ISAPI Filter dialog box, type *Oracle Certification Authentication Plugin* as name for the ISAPI filter.
7. In the Executable box, type the file system path of the WebGate cert_authn.dll or click the ellipsis button (...) to go to the folder that contains cert_authn.dll, and then click OK.

   ```
   WebGate_install_dir/access/oblix/apps/WebGate/bin/cert_authn.dll
   ```

   This example shows the default path, where *WebGate_install_dir* is the file system directory where you have installed this particular WebGate.
8. Click View Ordered List from the Action pane and arrange the filters as shown here by using "Move Up" or "Move Down":

   cert_authn.dll
   webgate.dll
9. Select Apply from Action pane.
10. Repeat for each WebGate installed on this host, for which you want to enable client certification.
11. Restart the IIS 7 Web server.
12. Proceed as needed for your deployment:
    - "Configuring IIS 7 WebGates for Pass Through Functionality"

19.6.4 Configuring IIS 7 WebGates for Pass Through Functionality

Here you will add WebGate.dll as a Wildcard Script Map to the required Web Site. While configuring WebGate to work with pass through functionality, you must ensure that "Physical Path" of the Web sites on which you are installing WebGates differ. Otherwise, the changes in "Handler Mappings" are reflected in all the Web Sites sharing the same physical path.

> **Note:** "Physical Path" is the path that is provided at the time of creating the Web Site. To check this path after the creation of the Web Site, , In Action pane click on Basic Settings..., you will be presented with a window showing the physical path of the Web Site.
>
> - Click the Web Site name.
> - In the Action pane, click Basic Settings.

To configure WebGate for pass through functionality

1. Start the Internet Information Services (IIS) Manager, if needed: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager.
2. Expand Sites in the connection pane.
3. Click the Web Site name for which you want to enable pass through.
4. Double click Handler Mappings from the selected Web Site's "home" pane.
5. From the Action pane, click Add Wildcard Script Map.
6. In the Name text box of the Add Wildcard Script Map dialog box, type WebGate as name for the ISAPI filter.

7. In the Executable box, type the file system path of the WebGate ISAPI filter file (WebGate.dll) or click the ellipsis button (...) to go to the folder that contains the WebGate.dll ISAPI filter file, and then click OK.

   ```
   WebGate_install_dir/access/oblix/apps/WebGate/bin/WebGate.dll
   ```

8. In the Access System Console:

   a. Locate the Web Gate profile and click Modify.

   b. Under User Defined Parameters, enter the following parameter and value:

      UseWebGateExtForPassthrough

      true

   c. Save the profile.

9. Repeat for each WebGate installed on this host, for which you want to enable pass through.

10. Restart the IIS 7 Web server.

11. Proceed to the next task: "Confirming IIS 7 WebGate Installation".

19.6.5 Confirming IIS 7 WebGate Installation

You can use the following procedure to confirm IIS 7 WebGate installation.

To verify IIS 7 WebGate installation

1. Go to the URL:

   ```
   http(s)://hostname:port/access/oblix/apps/webgate/bin/webgate.dll?progid=1
   ```

   where *hostname* refers to the name of the computer hosting the WebGate; *port* refers to the Web server instance port number.

2. The WebGate diagnostic page should appear.

   - **Successful**: If the WebGate diagnostic page appears, the WebGate is functioning properly and you can dismiss the page.
   - **Unsuccessful**: If the WebGate diagnostic page does not open, the WebGate is not functioning properly. In this case, the WebGate should be uninstalled and reinstalled. For more information about removing Oracle Access Manager see the *OAM Installation Guide* Chapter 22, then return to the chapter on installing a WebGate.

19.7 Installing and Configuring Multiple WebGates for a Single IIS 6 Instance

Unless explicitly stated, this topic applies equally to 32-bit and 64-bit WebGates.

> **See Also:** "Installing and Configuring Multiple 10g WebGates for a Single IIS 7 Instance" on page 19-14

This section describes how to install and configure multiple WebGates for different Web sites on same IIS Web server instance. Several steps are manual and will differ from those that are performed when you install a single WebGate with a single IIS instance. When installing multiple WebGates for a single IIS instance:

- The webgate.dll must be configured as an ISAPI filter at the individual Web site level, not the default (top) Web server level
- The /access virtual directory is mapped at the Web site level to the respective /access directory in the WebGate installation.

When configuring the impersonation DLL for multiple WebGates, you need to configure a user to act as the operating system.

There can only be one postgate.dll configured at the (top) Web Sites level of a machine. However, you might have multiple webgate.dlls configured at different levels below the top level Web Sites. If you perform multiple WebGate installations on one machine, multiple versions of the postgate.dll file might be created that can cause unusual Oracle Access Manager behavior.

Task overview: Installing and configuring multiple WebGates for a single IIS instance

1. Installing Each WebGate in a Multiple WebGate Scenario
2. Setting the Impersonation DLL for Multiple WebGates
3. Enabling SSL and Client Certification for Multiple WebGates
4. Perform the following tasks, which are the same whether you install one or more WebGates per IIS Web server instance:
   - "Ordering the ISAPI Filters" on page 19-8
   - "Confirming WebGate Installation on IIS" on page 19-26

   See Also: "Confirming Multiple WebGate Installation" on page 19-24

19.7.1 Installing Each WebGate in a Multiple WebGate Scenario

Unless explicitly stated, this topic applies equally to 32-bit and 64-bit WebGates.

After installing the ISAPI WebGate, there are several manual steps to perform as described here.

By default, webgate.dll is configured as an ISAPI filter at the Web sites (top) level. When installing multiple WebGates with a single IIS instance, you need to remove the respective webgate.dll from the top level and configure it for the appropriate individual Web site after each WebGate installation.

> **Note:** If you perform multiple WebGate installations on one machine, multiple versions of the postgate.dll file might be created which can cause unusual Oracle Access Manager behavior. The postgate.dll is not supported in environments where you have multiple WebGates configured with a single IIS v6 web server instance.

To install each WebGate when you will have several with one IIS instance

1. Install the ISAPI WebGate as described in Chapter 17.
2. Go to the Web site to protect, and configure webgate.dll as the ISAPI filter using these steps:
   a. Start the Internet Information Services (IIS) Manager: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager

   b. Right click **Web Sites**, and then click the **Properties** option.

   c. Click the ISAPI filter tab, look for the path to webgate.dll; if it is present in the filter, then select it and click the **Remove** button.

   d. Under Web Sites, right-click the name of the Web site to protect, and select the **Properties** option.

   e. Click the ISAPI filter tab to add the filter DLLs.

   f. Add the following filter to identify the path to the webgate.dll file, and name it "webgate".

   ```
   WebGate_install_dir/access/oblix/apps/webgate/bin/webgate.dll
   ```

   g. Save and apply these changes.

   h. Go to the **Directory Security** tab.

   i. Confirm that "anonymous access" and "basic authentication" are selected so that Oracle Access Manager provides authentication for this Web server.

   j. Save and apply these changes.

3. Go to Web sites level to protect and create an /access virtual directory that points to the newly installed *WebGate_install_dir*:

   a. Under **Web Sites**, right-click the name of the Web site to be protected.

   b. Select **New** and create a new virtual directory named `access` that points to the appropriate *WebGate_install_dir*/access.

   c. Under **Access Permissions**, check **Read**, **Run Scripts**, and **Execute**.

   d. Save and apply these changes.

4. In the file system, set directory permissions for Oracle Access Manager:

   a. In the file system, locate and right-click *WebGate_install_dir*\access, and the select **Properties**.

   b. Click the **Security** tab.

   c. Add user "IUSR_*machine_nam*e" and then select "**Allow**" for "**Modify**".

      For example, for a *machine_name* of Oracle, select IUSR_ORACLE.

   d. Add user "IWAM_*machine_nam*e" and then select "**Allow**" for "**Modify**"

      For example, for a *machine_name* Oracle, select IWAM_ORACLE.

   e. Add user "IIS_WPG" and then select "**Allow**" for "**Modify**".

   f. Add user "NETWORK SERVICE" and then select "**Allow**" for "**Modify**".

   g. For the group "Administrators", select "**Allow**" for "**Modify**".

5. If Webgate has been set up in Simple or Cert mode, perform the follow steps:

   a. In the file system, locate and right-click the "password.xml" file in *WebGate_install_dir*\access\oblix\config\password.xml.

   b. Click the Security tab.

   c. Give "Allow" for "Read" rights to users "IUSR_*machine_nam*e", IWAM_*machine_name*, "IIS_WPG", and "NETWORK SERVICE".

6. Add a new Web service extension using the following steps:

a. Right click **Web Service Extensions**, and then select **Add a new Web service extension**....

b. Add the Extension name **`Oracle WebGate`**.

c. Click **Add** to add the path to the extension file, and then enter the path to the appropriate webgate.dll.

```
WebGate_install_dir\access\access\oblix\apps\webgate\bin\webgate.dll
```

d. Click **OK** to save the changes.

e. Check box beside **Set extension status to allowed**.

f. Click **OK** to save the changes.

7. Ensure that there is no webgate.dll in the ISAPI filter at the top Web site level ("web sites").

8. Perform the next set of tasks using instructions in the following topics:

   a. "Setting the Impersonation DLL for Multiple WebGates" on page 19-22

   b. "Enabling SSL and Client Certification for Multiple WebGates" on page 19-23

9. Repeat these steps when you install the next WebGate for the IIS instance.

19.7.2 Setting the Impersonation DLL for Multiple WebGates

Unless explicitly stated, this topic applies equally to 32-bit and 64-bit WebGates and IIS v6.

The client's access token is known as an impersonation token. The impersonation token identifies the client, the client's groups, and the client's privileges. The information in the token is used during access checks when the thread requests access to resources on the client's behalf.

The Access System authenticates and authorizes the user. IISImpersonationExtension.dll of Oracle Access Manager in the wildcard extension behaves like a filter for each request to the Web server. The Access System designates a special user that does have the right to impersonate another user by configuring it using the impersonation username/password on the AccessGate Configuration page. That designated user must have "act as operating system" rights. DLL impersonates the user authenticated and authorized by Oracle Access Manager and generates the impersonation token.

You perform the following steps to set the impersonation DLL for each WebGate that protects a Web site for a single IIS Web server instance. You can do this either immediately after the installation task in the previous topic or all at one time.

Note: This task must be performed for each WebGate that protects an individual Web site for a single IIS Web server instance.

To add the impersonation DLL to IIS configuration for individual Web sites

1. Start the Internet Information Services (IIS) Manager, if needed: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager.

2. Click the plus icon (+) beside the Local Computer icon in the left pane to display your Web Sites.

3. Click **Web Service Extensions** in the left pane.

4. Double-click **WebGate** in the right pane to open the Properties panel.
5. Click the **Required Files** tab.
6. Click **Add**.
7. In the Path to file text box, type the full path to IISImpersonationExtension.dll, and then click OK. For example:

   ```
   WebGate_install_dir\access\oblix\apps\webgate\bin\IISImpersonationExtension.dll
   ```

   This example shows the default path, where *WebGate_install_dir* is the file system directory where you have installed this particular WebGate.
8. Verify that the Allow button beside the WebGate icon is grayed out, which indicates that the dll is allowed to run as a Web service extension.
9. Right click the Web site name, and then click **Properties**.
10. Click the **Home Directory** tab, and then click the **Configuration** button.
11. In the list box for Wildcard application maps, click the entry for `IISImpersonationExtension.dll` to highlight it, then click **Edit**.
12. Ensure that the box is unchecked, and then click **OK**.
13. Repeat these steps for each WebGate and Web site pair for the IIS Web server instance.
14. Proceed as follows:
    - **Client Certificate Authentication**: "Enabling SSL and Client Certification for Multiple WebGates"
    - "Confirming Multiple WebGate Installation" on page 19-24.

19.7.3 Enabling SSL and Client Certification for Multiple WebGates

You perform this task to set the enable client certification for each WebGate that protects a Web site for a single IIS Web server instance. You can do this either immediately after the adding the impersonation DLL to an individual Web site or all at one time.

> **Note:** Procedures in this topic apply equally to 32-bit and 64-bit WebGates, and IIS 6, unless stated otherwise.

If you select client certificate authentication during setup, you must also add the cert_authn.dll as one of the ISAPI filters in the respective Web site.

To enable SSL on the IIS v6 Web server

1. Start the Internet Information Services (IIS) Manager, if needed: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager.
2. Expand the local computer icon to display your Web Sites.
3. Expand the appropriate individual Web Site, then expand \access\oblix\apps\webgate\bin.
4. Right click `cert_authn.dll` and select **Properties**.
5. In the Properties panel, select the **File Security** tab.
6. In the Secure Communications sub-panel, click **Edit**.

7. In the Client Certificate Authentication sub-panel, click **Accept Certificates** and click **OK**.
8. Click **OK** in the cert_authn.dll Properties panel.
9. Repeat for each WebGate installed on this host.
10. Proceed to the next task: "To add cert_authn.dll as an ISAPI filter".

To add cert_authn.dll as an ISAPI filter

1. Start the Internet Information Services console, if needed.
2. Expand the local computer to display your Web Sites.
3. Right click the appropriate Web Site to display the Properties panel.
4. Click the **ISAPI Filters** tab, then click the **Add** button to display the Filter Properties panel.
5. Enter filter name "`cert_authn`".
6. Click the **Browse** button and navigate to the following directory:

   \WebGate_install_dir\\access\\oblix\\apps\\webgate\\bin
7. Select `cert_authn.dll` as the executable.
8. Click **OK** on the **Filter Properties** panel.
9. Click **Apply** on the **ISAPI Filters** panel.
10. Click **OK**.
11. Repeat for each WebGate installed on this host.
12. Ensure the filters are listed in the correct order.
13. Proceed to "Confirming Multiple WebGate Installation".

19.7.4 Confirming Multiple WebGate Installation

This task applies equally to 32-bit and 64-bit WebGates, and IIS v6 Web servers.

If you perform multiple WebGate installations on one machine, multiple versions of the postgate.dll file might be created which can cause unusual Oracle Access Manager behavior. the postgate.dll is not supported in environments where you have multiple WebGates configured with a single IIS v6 web server instance.

See Also:

- "Finishing 64-bit WebGate Installation" on page 19-24
- "Confirming WebGate Installation on IIS" on page 19-26

19.8 Finishing 64-bit WebGate Installation

This section describes how to complete installation of a 64-bit WebGate. You can skip this section if you are installing a 32-bit WebGate. In this case, see instead, "Completing WebGate Installation with IIS" on page 19-7.

Before you start tasks here, be sure that you have completed WebGate installation according to information in Chapter 17. You must also have completed Web server configuration updates for this WebGate either automatically during WebGate installation or manually, as described in "WebGates for IIS v6" on page 19-4.

Task overview: Finishing installation of a 64-bit WebGate

1. Perform steps in "Setting Access Permissions, ISAPI filters, and Directory Security Authentication" on page 19-25.
2. Enable client certificates, if desired. See "Setting Client Certificate Authentication" on page 19-26.
3. When finished, you can:
   - Confirm operations as described in "Confirming WebGate Installation on IIS" on page 19-26
   - Create a policy domain to protect this domain as described in the *Oracle Access Manager Access Administration Guide*.
   - Implement Windows Impersonation, as described in the *Oracle Fusion Middleware Integration Guide for Oracle Access Manager*.

19.8.1 Setting Access Permissions, ISAPI filters, and Directory Security Authentication

Unless explicitly stated, this topic applies equally to 32-bit and 64-bit WebGates. It describes setting access permissions for the Web site that you are using as a default.

To set or confirm access Permissions, ISAPI filters, and Directory Security Authentication

1. Start the Internet Service Manager. For example, from the Start menu click Programs then click Administrative Tools, and click Internet Service Manager.
2. Expand the local computer by clicking +, in the left panel.
3. Click to expand the Web Sites tab.
4. Right-click Default Web Site (or the site you are using as a default), and create a virtual directory as described in "Protecting a Web Site When the Default Site is Not Setup" on page 19-13.
5. Right-click **Web Sites** in the Internet Information Services tab, click **Properties**, and perform the following steps:

   a. From the Internet Information Services tab, click the **Edit** button.

   b. Locate the ISAPI filter tab to confirm (or add) the filter DLLs, as follows:

      Filter: If you updated the IIS Web server configuration file, webgate.dll should be properly located.

      No Filter: Add the webgate.dll filter from *WebGate_install_dir*\oblix\access\apps\webgate\bin\webgate.dll

   c. Save and apply any changes.

   d. Click the Directory Security tab and confirm that both **Anonymous Access** and **Basic Authentication** are selected.

      Selected: Proceed to Step 6.

      Not Selected: Select **Anonymous Access** and **Basic Authentication**, then save and apply these changes.
6. Proceed as follows:
   - "Setting Client Certificate Authentication", if desired
   - **No Client Certificate Authentication**: Restart the IIS Web server.

- **Filter Positions**: Perform instructions in "Ordering the ISAPI Filters" on page 19-8 to ensure that all filters have been added and are in the proper order.

19.8.2 Setting Client Certificate Authentication

This task is optional and should be performed only if you want to use client certificate authentication. In this case, IIS and WebGate must be SSL-enabled.

Information in this topic is a sub set of details in "Enabling Client Certificate Authentication on the IIS Web Server" on page 19-7.

To add cert_authn.dll as an ISAPI filter

1. Start the Internet Information Services console, if needed: Click Start, Programs, Administrative Tools, Internet Service Manager.
2. Expand the local computer to display your Web Sites.
3. Right-click the Default Web Site (or the Web site that you use as a default), then expand \access\oblix\apps\webgate\bin.
4. Right click cert_authn.dll and select Properties, then:
    a. In the Properties panel, select the File Security tab.
    b. In the Secure Communications sub-panel, click Edit.
    c. In the Client Certificate Authentication sub-panel, click Accept Certificates and click OK.
    d. Click OK in the Secure Communications panel.
    e. Click OK in the cert_authn.dll Properties panel.
5. Click the **ISAPI Filters** tab, click the **Add** button to display the Filter Properties panel, and then:
6. Ensure the filters are listed in the correct order, as described in "Ordering the ISAPI Filters" on page 19-8.
7. Proceed to "Confirming WebGate Installation on IIS" on page 19-26.

19.9 Confirming WebGate Installation on IIS

After installing WebGate and updating the IIS Web server configuration file, you can use the WebGate diagnostics to verify the WebGate is properly installed.

Note: This task is the same for both 32-bit and 64-bit WebGates. It is the same whether you are installing one or more WebGates per IIS Web server instance.

To verify WebGate installation

1. Go to the URL:

   ```
   http(s)://hostname:port/access/oblix/apps/webgate/bin/webgate.dll?progid=1
   ```

   where *hostname* refers to the name of the computer hosting the WebGate; *port* refers to the Web server instance port number.
2. The WebGate diagnostic page should appear.

- **Successful**: If the WebGate diagnostic page appears, the WebGate is functioning properly and you can dismiss the page.
- **Unsuccessful**: If the WebGate diagnostic page does not open, the WebGate is not functioning properly. In this case, the WebGate should be uninstalled and reinstalled. For more information about removing Oracle Access Manager see "Removing a 10g WebGate from the OAM 11g Deployment" on page 17-25, in the chapter on installing a 10g WebGate Chapter 17.

19.10 Starting, Stopping, and Restarting the IIS Web Server

When instructed to restart your IIS Web server during Oracle Access Manager Web component installation or setup, be sure to follow any instructions that appear on the screen. Also, consider using `net stop iisadmin` and `net start w3svc` are good ways to stop and start the Web server. The `net` commands help to ensure that the Metabase does not become corrupted following an installation.

19.11 Removing Web Server Configuration Changes Before Uninstall

The information in this section applies equally to 32-bit and 64-bit WebGates.

Web server configuration changes that occur during installation must be manually reverted after uninstalling the WebGate. For example, the ISAPI transfilter will be installed for IIS WebGate. However, if you uninstall WebGate this is not removed automatically. Also, the created Web service extension and the link to the identity directory will not be removed. This type of information must be removed manually. These are examples of information to remove, not a complete list.

Further, you must remove any changes that you manually made to your Web server configuration file for the WebGate should be removed. For more information about what is added for each component, look elsewhere in this chapter.

To fully remove a WebGate and related filters from IIS, you must do more than simply remove the filters from the list in IIS. IIS retains all of its settings in a metabase file. On Windows 2000 and later, this is an XML file that can be modified by hand. There is also a tool available, MetaEdit, to edit the metabase. MetaEdit looks like Regedit and has a consistency checker and a browser/editor. To fully remove a WebGate from IIS, use MetaEdit to edit the metabase.

20

Configuring the ISA Server for 10g WebGates

This chapter describes how to configure the Oracle Access Manager ISAPI WebGate and Microsoft Internet Security and Acceleration Server (ISA Server) to operate together. Topics include:

- Prerequisites
- About Oracle Access Manager and the ISA Server
- Compatibility and Platform Support
- Installing and Configuring WebGate for the ISA Server
- Configuring the ISA Server for the ISAPI WebGate
- Starting, Stopping, and Restarting the ISA Server
- Removing Oracle Access Manager Filters Before WebGate Uninstall on ISA Server

20.1 Prerequisites

Ensure that your OAM 11g Administration Console is running and get familiar with:

- "Introduction to Policy Enforcement Agents" on page 5-1
- "About Oracle Access Manager and the ISA Server" on page 20-1

20.2 About Oracle Access Manager and the ISA Server

The ISA Server is Microsoft's "integrated edge security gateway". It is designed to protect IT environments from Internet-based threats and to give users secure remote access to applications and data.

WebGate is the Oracle Access Manager Web server plug-in access client that intercepts HTTP requests for Web resources and forwards them to the Access Server for authentication and authorization. ISAPI is the Internet Web server extension that Oracle Access Manager uses to identify WebGates that communicate with the ISA Server (and the IIS Web Server).

This WebGate has been tested to operate with the ISA Server in scenarios that use both Oracle Access Manager Basic and Form (form-based) authentication schemes. You develop Basic and Form authentication schemes and policy domains using Oracle Access Manager as usual.

Note: Oracle Access Manager Client Certificate authentication is not supported for the ISA Server.

See Also: *Oracle Access Manager Access Administration Guide* for more information about authentication management and policy domains.

Using ISA Server with Oracle Access Manager is similar to using the IIS Web server. However, the ISA Server provides firewall and Virtual Private Network (VPN) functions.

ISA Server can be configured for third-party security filters. To enforce Oracle Access Manager security during authentication and authorization when you use ISA Server, both webgate.dll and postgate.dll must be registered as ISA Server Web filters. Every request to the Access Server that passes through ISA Server requires webgate.dll and postgate.dll.

The following overview outlines the tasks that you must perform and the topics where you will find the steps to set up the ISAPI WebGate with the ISA Server.

Task overview: Installing and configuring the ISAPI WebGate on ISA Server

1. Confirming "Compatibility and Platform Support" on page 20-2
2. "Installing and Configuring WebGate for the ISA Server" on page 20-2.
3. "Configuring the ISA Server for the ISAPI WebGate" on page 20-3.
4. Perform the following tasks, as described in:
    a. "Ordering the ISAPI Filters" on page 20-6
    b. "Removing Oracle Access Manager Filters Before WebGate Uninstall on ISA Server" on page 20-7

20.3 Compatibility and Platform Support

Get the latest certification matrix from Oracle Technology Network at the following URL:

http://www.oracle.com/technology/products/id_mgmt/coreid_acc/pdf/oracle_access_manager_certification_10.1.4_r3_matrix.xls

20.4 Installing and Configuring WebGate for the ISA Server

After ISA Server installation, you perform the following tasks to install WebGate for use with ISA Server.

See Also: "Compatibility and Platform Support" on page 20-2

Task overview: Performing WebGate configuration for ISA Server includes

1. "Installing WebGate with ISA Server" on page 20-3
2. "Changing /access Directory Permissions" on page 20-3
3. "Registering Oracle Access Manager Plug-ins as ISA Server Web Filters" on page 20-4

20.4.1 Installing WebGate with ISA Server

When you install WebGate with the ISA Server, the destination for the ISAPI WebGate installation (also known as the *WebGate_install_dir*) should be same as that of the Microsoft ISA Server. For example, if ISA Server is installed on C:\Program Files\Microsoft ISA Server, the ISAPI WebGate should also be installed there.

> **Note:** During WebGate installation, do not automatically update the ISA Server configuration. Instead, choose "No" when asked about automatic updates to the ISA Server configuration.

Task overview: Installing the ISAPI WebGate for the ISA Server

1. See Chapter 17 for details on the following topic, as these apply to your environment:
   - Provisioning a 10g WebGate with OAM 11g
   - Locating and Installing the Latest OAM 10g WebGate for OAM 11g
   - Deploying Applications in a WebLogic Container
   - Configuring Centralized Logout for 10g WebGate with OAM 11g
2. Changing /access Directory Permissions on page 20-3

20.4.2 Changing /access Directory Permissions

After finishing ISAPI WebGate installation and configuration for the ISA Server, you need to change permissions to the \access subdirectory. This subdirectory was created in the ISA Server (also WebGate) installation directory. You need to add the user NETWORK SERVICE and grant full control to NETWORK ADMINISTRATOR.

This enables the ISA Server to establish a connection between the WebGate and Access Server. Certain configuration files should be readable by network administrators, which is why you grant NETWORK ADMINISTRATOR full control.

To change permissions for the \access subdirectory

1. In the file system, right-click *WebGate_install_dir*\access, and select **Properties**.
2. In the Properties window, click the **Security** tab.
3. Add user "NETWORK SERVICE" and then select "Allow" to give "**Full Control**".
4. For the "NETWORK ADMINISTRATOR", select "**Full Control**".

20.5 Configuring the ISA Server for the ISAPI WebGate

The following topics describe how to configure the ISA Server to operate with the Oracle Access Manager ISAPI WebGate.

Task overview: Performing WebGate configuration for ISA Server includes

1. "Registering Oracle Access Manager Plug-ins as ISA Server Web Filters" on page 20-4
2. "Configuring ISA Firewall Policies for ISA Web Filters" on page 20-4

20.5.1 Registering Oracle Access Manager Plug-ins as ISA Server Web Filters

After resetting ISAPI WebGate permissions, you need to register Oracle Access Manager webgate.dll and postgate.dll plug-ins as Web Filters within ISA Server. Web filters screen all HTTP traffic that passes through the ISA Server host. Only compliant requests are allowed to pass through.

Oracle Access Manager authentication schemes define how the user is challenged for credentials, maps user-supplied information, verifies it, and so forth. With the ISA Server, you must choose either Form or Basic authentication as the challenge method. You must also specify a Challenge Parameter to map the credentials provided by the user to the corresponding user profile stored in the directory server.

Note: If Oracle Access Manager libraries are not registered as ISA Web filters, Oracle Access Manager authentication could fail. Do not point to webgate.dll in the action path for form-based login in the authentication scheme. Instead, specify the path to a dummy file in the /access directory as shown here:

```
action= "/access/dummy"
```

For form based authentication, postgate.dll must be installed and should be at a higher level than webgate.dll.

The following procedure describes how to register Oracle Access Manager plug-ins in the ISA Server.

Note: If you need to undo the filter registration, you can use the following procedure with the `/u` option in the `regsvr32` command. For example: `regsvr32 /u ISA_install_dir\access\oblix\apps\webgate\bin\webgate.dll`

To register Oracle Access Manager plug-ins as ISA Server Web filters

1. Locate the ISA Server installation directory, from which you will perform the following tasks.
2. Run `net stop fwsrv` to stop the ISA Server.
3. Register the webgate.dll as an ISAPI Web filter by running `regsvr32 ISA_install_dir\access\oblix\apps\webgate\bin\webgate.dll`.
4. Register the postgate.dll as an ISAPI Web filter by running `regsvr32 ISA_install_dir\access\oblix\apps\webgate\bin\postgate.dll`.
5. Restart the ISA Server by running `net start fwsrv` to restart the ISA Server.
6. Proceed to "Configuring ISA Firewall Policies for ISA Web Filters".

20.5.2 Configuring ISA Firewall Policies for ISA Web Filters

To authenticate users, ISA Server must be able to communicate with the authentication servers. After registering Oracle Access Manager webgate.dll and postgate.dll as ISA Web filters, you must configure the ISA Firewall Policy rule to protect resources using these Web filters.

Web publishing rules essentially map incoming requests to the appropriate Web servers. Access rules determine how clients on a source network access resources on a destination network. ISA Firewall Policy rules require client membership in a user set:

either Firewall clients, authenticated Web clients, or virtual private network (VPN) clients. The ISA Server attempts to match authenticated users based upon ISA Firewall Policy rules.

> **See Also:** Your ISA Server documentation for details about ISA Firewall Policies and rules

The following procedure describes how to configure an ISA Firewall Policy rule to use with ISA Web filters for Oracle Access Manager webgate.dll and postgate.dll.

> **Note:** After you perform the following procedure, when you create a listener in the authentication click Allow client authentication over HTTP in Advanced Properties.

To configure ISA policies to enable Oracle Access Manager authentication and authorization

1. From the Start menu, click **All Programs**, click **Microsoft ISA Server**, and then click **ISA Server Management**.
2. From the tree of the ISA Server Management console, locate the name of this server, and then click **Firewall Policy**.
3. From the Tasks tab, click **Publish Web Sites**.
4. In the **Web publishing rule** name field, type a descriptive name for the rule, and then click **Next**.
5. On the Select Rule Action page, confirm that the Allow option is selected, and then click **Next**.
6. In the **Publishing type**, confirm that the **Publish a single Web site or load balancer** option is selected, and then click **Next**.
7. On the Server Connection Security page, click **Use non-secured connections to connect the published Web server or server farm**, and then click **Next**.

   > **Note:** If you are using secured connections, see the server connection security settings provided by ISA Server.

8. Perform the following steps to set internal publishing details:

   a. In the **Internal site name** box, type the internally-accessible name of the Web server.

   b. Check the **Use a computer name or IP address to connect to the published server** check box.

   c. Type the internally-accessible and fully qualified domain name, or type the IP address of the Web server computer, in the **Computer name or IP address** box

   d. Click **Next**.

9. In the **Public name** box, type the publicly-accessible domain name of the Web server computer, and then click **Next**.
10. To publish a particular folder in the Web site:

    a. Type the folder name in the **Path (optional)** box to display the full path of the published Web site in the Web site box.

   b. Click **Next**.

11. In the **Accept requests for** list:

    a. Click **This domain name (type below)**.

    b. In the Public name box, type the publicly-accessible fully qualified domain name of the Web site.

    c. Click **Next**.

12. In the **Web listener** list, either click the **Web listener** to use for this Web publishing rule; otherwise or create a new Web listener, as follows:

    a. Click **New**, type a descriptive name for the new Web listener, and then click Next.

    b. Click **Do not require SSL secured connections with clients**, and then click **Next**.

    c. In the **Listen for requests from these networks** list, click the required networks and click to check the **External** box, then click **Next**.

    d. In the **Select how clients will provide credentials to ISA Server** list, click **No Authentication**, and then click **Next**.

    e. On the Single Sign On Settings page, click **Next**, and then click **Finish**.

13. **Authentication Delegation**: Perform the following steps in the **Select the method used by ISA Server to authenticate to the published Web server** list:

    a. Click **No Delegation**.

    b. Click **Client Cannot Authenticate Directly**.

    c. Click **Next**.

    This is used by ISA Server to authenticate to the published Web server.

14. On the User Sets page:

    a. Choose **All** (the default user setting) to set the rule that applies to requests from the user sets box.

    b. Click **Next** and then click **Finish**.

15. Click **Apply** to update the firewall policy, and then click **OK**.

16. Validate that only applicable ports are open and that the traffic that you would like to pass through is allowed.

20.5.3 Ordering the ISAPI Filters

It is important to ensure that the WebGate ISAPI filters are included in the right order. postgate.dll should be loaded before webgate.dll.

To order the WebGate ISAPI filters for ISA Server

1. From the Start menu, click All Programs, click Microsoft ISA Server, and then click ISA Server Management.

2. Expand Configuration, then check Add-ins to display your Web-filters.

3. Right-click the Web-filters and select Properties.

4. Confirm the following .dll files appear.

   For example:

postgate.dll
webgate.dll

5. Add any missing filters, if needed, then select a filter name and use the up and down arrows to arrange the filter order as shown in step 5.

WARNING: Confirm that there is only one webgate.dll and one postgate.dll filter and ensure that these are in an enabled state. Also, ensure that postgate.dll is installed at higher priority level than webgate.dll.

20.6 Starting, Stopping, and Restarting the ISA Server

When instructed to restart your ISA Server during Oracle Access Manager Web component installation or setup, be sure to follow any instructions that appear on the screen. Also, consider using `net stop fwsrv` and `net start fwsrv` are good ways to stop and start the ISA Server. The `net` commands help to ensure that the Metabase does not become corrupted following an installation.

For more information, see your ISA Server documentation.

20.7 Removing Oracle Access Manager Filters Before WebGate Uninstall on ISA Server

If you plan to uninstall the WebGate that is configured to operate with the ISA Server, you must first unregister the Oracle Access Manager filters manually, and then uninstall WebGate.

See Also: Chapter 17 for details about uninstalling Oracle Access Manager 10g WebGates

To unregister filters before WebGate uninstall

1. Stop the ISA Server.
2. Run the following command to unregister webgate.dll. For example:

```
regsvr32 /u ISA_install_dir\access\oblix\apps\webgate\bin\webgate.dll
```

3. Run the following command to unregister postgate.dll. For example:

```
regsvr32 /u ISA_install_dir\access\oblix\apps\webgate\bin\postgate.dll
```

21

Configuring Lotus Domino Web Servers for 10g WebGates

This chapter provides tips about installing and configuring Lotus Domino to operate with the WebGate. Topics include:

- Prerequisites
- Installing the Domino Web Server
- Setting Up the First Domino Web Server
- Starting the Domino Web Server
- Enabling SSL (Optional)
- Installing a Domino Security (DSAPI) Filter

> **Note:** The information here presumes that you are familiar with your operating system commands, Lotus Notes, and the Domino Web server.

21.1 Prerequisites

Ensure that your OAM 11g Administration Console is running and get familiar with:

- "Introduction to Policy Enforcement Agents" on page 5-1
- "About Installing Fresh OAM 10g WebGates to Use With OAM 11g" on page 17-2

21.2 Installing the Domino Web Server

Before you install the WebGate with a Domino Web server, you need a properly installed and set up Domino Enterprise Server R5. The following information focuses on Solaris. However, with some modifications, these steps can be used as a guide for other UNIX systems.

> **Note:** You need to register if this is the first time you download from lotus.com.

To download the Domino Web server on UNIX

1. Download Lotus Domino from the following URL:

```
http://www-10.lotus.com/ldd/down.nsf
```

2. Untar the downloaded file to your staging area. For example:

   ```
   gct@planetearth[/export/users2/gct/temp] 433 : ls C37UUNA.tar
   gct@planetearth[/export/users2/gct/temp] 434 : tar xf C37UUNA.tar
   gct@planetearth[/export/users2/gct/temp] 435 : ls C37UUNA.tar sol/
   ```

You need to install Domino as user "root". The installation script creates soft link, /opt/lotus, to link to your Lotus Domino installation directory.

To install the Domino Web server on UNIX

1. Run the install script for the Domino Web server. For example:

   ```
   gct@planetearth[/export/users2/gct/temp/sol] 441 : su root
   Password:
   root@planetearth[/export/users2/gct/temp/sol] 1 : ls
   install* license.txt script.dat sets/ tools/
   root@planetearth[/export/users2/gct/temp/sol] 2 :
   root@planetearth[/export/users2/gct/temp/sol] 2 : ./install
   ========================================================
   Domino Server Installation
   ========================================================
   Welcome to the Domino Server Install Program.
   Type h for help on how to use this program.
   Press TAB to begin the installation.
   --------------------------------------------------------
   Type h for help
   Type e to exit installation
   Press TAB to continue to the next screen.
   --------------------------------------------------------
   ```

   You are asked to select the setup type.

2. Select Setup type. For example:

   ```
   Select Setup type: [Domino Enterprise Server]
   ```

3. Complete the installation with the following considerations in mind. For example:

   - The default program directory is set to /opt/lotus. You may over write it to another directory. For example, /export/home/WWW/lotus.
   - The default data directory is set to /local/notesdata1. You may also over write this to something else. For example, /export/home/WWW/lotus/data1.
   - Over write Domino UNIX user to own data directory. The default user is set to notes. You may change it to a valid UNIX user. For example, gct or root.
   - Over write "The UNIX user for this directory must be a member of this group". The default group is set to notes. You may change it to a valid UNIX group name. For example: oblix.

   Note: Be sure to put Domino data directory in your $PATH before you proceed from here.

21.3 Setting Up the First Domino Web Server

After successfully installing, you must set up the first Domino server.

To set up first Domino server

1. Run /opt/lotus/bin/http httpsetup.

   By default, Domino will use port 8081.

2. Ensure that port 8081 is not already in use.
3. Launch your browser and enter the URL that follows. For example:

   ```
   http://hostname:8081
   ```

4. Follow instructions on the screen and keep the following in mind.
   - Check HTTP to get the Web server.
   - Ensure the designated administrator has a first and last name.
   - Keep passwords simple, and record them in a safe location. For example, oracleoracle.
5. Run all commands as the UNIX user that you've configured for this Domino Web server.

   > **WARNING: Do not run as root.**

21.4 Starting the Domino Web Server

After successfully setting up the first Domino Web server, you must start it.

To start Domino server

1. Run /opt/lotus/bin/server.
2. Launch your browser and enter the following URL.

   For example:

   http://*hostname*:*80*/names.nsf

   You will be prompted for login name and password.

3. Select Server-Server.
4. Select your intended server.
5. Select Edit Server.
6. Select Ports, select Internet Ports, then click Web.
7. Change the value for TCP/IP port number to your desired port number.
8. Click Save and Close to save all your changes.
9. Restart server /opt/lotus/bin/server.

21.5 Enabling SSL (Optional)

Enabling SSL is not mandatory for the WebGate. However, if you need to generate a keyring file (.kyr) and its corresponding stash file (.sth) from the Lotus Notes client on a Windows system to the UNIX system, use the steps that follow.

To generate the keyring and stash files

1. Launch the Lotus Notes Client on your Windows system.

For example:

File, select Databases, then click Open

2. Select Server Certificate Admin.
3. Create the key ring file.
4. Create the certificate request.
5. Install the trusted root certificate into the key ring file.
6. Install the certificate into the key ring file.
7. Copy or ftp the newly created keyring file and stash file from the Windows system to your UNIX computer.
8. Store both files in your Domino data directory.

To enable SSL

1. Launch your browser and enter the following URL.

   For example:

   http://*hostname*:*port*/names.nsf

   You will be prompted for login name and password
2. Select Server-Server.
3. Select your intended server.
4. Select Edit Server.
5. Select Ports, select Internet Ports, then click Web.
6. In the SSL Key file name field, enter the absolute path to the keyring file.
7. Change the SSL Port number value to your desired port number.
8. Enable SSL port status.
9. Select Client Certificate "Yes" for Client Certificate authentication.
10. Click Save and Close to save all your changes.
11. Restart the Web server.

    For example:

    /opt/lotus/bin/server

21.6 Installing a Domino Security (DSAPI) Filter

The Domino security API filter, DSAPI, is an authentication method that enables you to register a DLL with the Domino Web server. In this case, the Web server calls the WebGate DLL to authenticate the user when a request for authentication occurs rather than using SSL or basic authentication.

Authentication within Domino is optional with the Oracle Access Manager DSAPI filter. You can implement certain aspects of authentication that the default Web server does not support.

Task overview: Completing WebGate and filter installation

1. Before you install the WebGate on a Domino Web server, complete all steps described earlier.

2. Complete the WebGate installation and Web server update as described in "Locating and Installing the Latest OAM 10g WebGate for OAM 11g" on page 17-6.
3. See "Completing the WebGate Installation" on page 21-5 and choose one of the two options discussed there.

21.6.1 Completing the WebGate Installation

To ensure the Domino Web Server can use the WebGate DLL, you need to edit the enter the name or names of the DLL/DLLs (DSAPI libraries) to be called for authentication in the DSAPI filter file names field of the HTTP tab under the Internet Protocols tab in the Server document.

> **Note:** Relative paths will be based on the Domino executable directory. DSAPI filter libraries will be called to handle events in the order they appear in this list.

There are two ways to install the filter:

- Through a Web browser and names.nsf (option 1)
- Through a Lotus Notes workstation and the Address Book (option 2)

Option 1: To setup the DSAPI filter to access names.nsf

1. Go to the names.nsf URL and log in. For example:

   ```
   http://hostname:port/names.nsf
   ```

2. Click the Server-Servers link.

   A Java applet will be loaded.
3. Select a server from those listed.
4. Click the Edit Server link to go to Edit mode.
5. Click the Internet Protocols link.

   By default, the HTTP tab is selected and information is displayed in Edit mode.
6. Look for DSAPI where it says "DSAPI filter file names:", then type in the absolute path to the libwebgate.so file.
7. Save your changes.
8. Restart the Domino http server task.

Option 2: To access the Address Book through Lotus Notes

1. Open Domino Name and Address book. For example, select:

   File, Database, Open, then click Address Book
2. Switch to server view and open the server document.
3. Edit the server document.
4. Click the Internet Protocols tab.

   By default, the HTTP tab is selected and information is displayed in Edit mode.
5. Look for DSAPI where it says "DSAPI filter file names:", then type in the absolute path to the libwebgate.so file.

6. Save your changes.
7. Restart the Domino http server task.

Part VIII

Appendixes

Part VIII provides information that is outside the scope of day-to-day administration tasks with Oracle Access Manager 11g.

Part VIII contains the following appendixes:

- Appendix A, "Transitioning OAM 11g from a Test to a Production Environment"
- Appendix B, "Co-existence Overview: OAM 11g and OSSO 10g"
- Appendix C, "Integrating Oracle ADF Applications with Oracle Access Manager 11g SSO"
- Appendix D, "Internationalization and Multibyte Data Support for OAM 10g WebGates"
- Appendix E, "Securing Communication with OAM 11g"
- Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"
- Appendix G, "Configuring OAM 11g for IPv6 Clients"
- Appendix H, "Troubleshooting"

A

Transitioning OAM 11g from a Test to a Production Environment

This chapter describes how to move Oracle Access Manager 11g data from a test environment to a production environment. You can also use this approach for testing and rolling out upgrades.

This chapter includes the following topics:

- Prerequisites
- Introduction to Deployment Scenarios and Data Types
- Introduction to Methods and Tools
- Planning an OAM 11g Move from Test to Production
- Backup and Recovery Strategies
- Moving OAM 11g From Test to Production

> **See Also:** *Oracle Fusion Middleware Administrator's Guide* for details about moving Oracle Access Manager 10g to a new (or an existing) production environment.

A.1 Prerequisites

Install and configure target components, as described in the Oracle Fusion Middleware Installation Guide for Oracle Identity Management

A.2 Introduction to Deployment Scenarios and Data Types

It is fairly common to develop and test applications in a smaller restricted environment, and then eventually roll out the test applications (and, optionally, test data) to a larger shared environment. You can use the information in this chapter to move Oracle Access Manager 11g data from a test deployment (the source) to a production deployment (the target). You can also use this approach for testing and rolling out upgrades.

Table A–1 describes the types of deployments that customers might have within their enterprise. Deployment types might be named differently in your enterprise.

Table A–1 Deployment Types

Deployment Type	Description
Development Deployment	Ideally a *sandbox*-type setting where the dependency on the overall deployment is minimal
QA Deployment	Typically a smaller shared deployment used for testing
Pre-production Deployment	Typically a shared deployment used for testing with a wider audience
Production Deployment	Fully shared and available within the enterprise on a daily basis

Within each deployment, OAM 11g configuration data is stored in files while OAM 11g policy data is stored in a database.

On the policy side, each application domain is constructed using the following shared components:

- Authentication Module
- Authentication Scheme (containing one authentication module)
- Host-Identifiers
- Resources

On the system configuration side, agents host resources or partner applications that must be protected. An agent can be an OAM Agent (WebGate or AccessGate) or OSSO agent. Each agent must be registered with OAM 11g to protect hosted resources. Registering an agent:

- Defines the agent and its specific configuration parameters
- Creates an application domain for the specified resources
- Creates an authentication policy with the default authentication scheme for the partner application
- Creates an authorization policy for the specified resources
- Generates the symmetric key for the partner application

Transitioning policy and partner information from the test (source) environment to the production (target) environment is accomplished with MBean registered on the AdminServer of the OAM Server. WLST commands fetch the partner and policy information from the source server and applies this on the production server.

The following overview presents the general scope of tasks that must be performed to move OAM 11g policy and partner information from a test (source) environment to a production (target) environment.

Task overview: Moving OAM 11g from test to production

1. Perform planning activities, as described in Planning an OAM 11g Move from Test to Production.
2. Export data such as users and groups, the identity and policy stores, and credentials from the source and then Import data to the target as described in Moving OAM 11g From Test to Production.

   To reuse source data in only the target system, you can re-register the agents within the target deployment after importing the source data to the target
3. Modify any information that is specific to the new environment such as host name or ports.
4. Deploy applications.

A.3 Introduction to Methods and Tools

This section provides a high-level overview of data migration approaches, methods, and tools for OAM 11g:

- About New versus Existing Production Environments
- About Methods to Move from Test to Production
- About the WebLogic Scripting Tool Commands
- About Conflict Resolution
- About Building a Dependency Tree for Each Application Domain

A.3.1 About New versus Existing Production Environments

Oracle Access Manager and Oracle Identity Manager are components of Oracle Fusion Middleware 11g. All existing access technologies in the Oracle Identity Management stack converge in Oracle Access Manager 11g. The differences in the scope of tasks required to move an entire Identity Management environment from a test source to a production target are described in Table A–2.

Table A–2 Differences when Transferring Data to New versus Existing Target Environments

New Target Environment	Existing Target Environment
In this scenario you want to move existing Identity Management components in a test environment to a new environment that does not yet exist. This requires the following tasks. Task 8 is the subject of this chapter. All other tasks are described in the *Oracle Fusion Middleware Administrator's Guide*	In this scenario you want to move one or more applications from the source to a target in an existing environment, while retaining the source security-related configuration. This requires migrating application-specific data and incremental changes from the source to the target. Task 2 is the subject of this chapter. All other tasks are described in the *Oracle Fusion Middleware Administrator's Guide*
1. Copy the Database to a New Production Environment 2. Move Oracle Internet Directory to a New Production Environment 3. Move Oracle Virtual Directory to a New Production Environment 4. Move Oracle Directory Integration Platform to a New Production System 5. Move Oracle Identity Federation to a New Production Environment 6. Move Oracle Identity Manager to a New Production Environment 7. Move Oracle Identity Navigator to a New Production Environment 8. Move Oracle Access Manager 11g to a New Production Environment, as described in "Moving OAM 11g From Test to Production" on page A-10 9. Move Oracle Access Manager 10g to a New Production Environment 10. Move Oracle Adaptive Access Manager to a New Production Environment 11. Move Audit Policies to a New Production Environment 12. Move Oracle Platform Security to a New Production Environment 13. Move Oracle Web Services Manager to a New Production Environment	1. Move Oracle Internet Directory to an Existing Production Environment 2. Move Oracle Access Manager 11g to a New Production Environment, as described in "Moving OAM 11g From Test to Production" on page A-10 3. Move Oracle Access Manager 10g to a New Production Environment 4. Move Oracle Adaptive Access Manager to a New Production Environment 5. Move Oracle Identity Manager to a New Production Environment 6. Move Oracle Identity Navigator to a New Production Environment 7. Move Oracle Platform Security to a New Production Environment 8. Move Oracle Web Services Manager to a New Production Environment

A.3.2 About Methods to Move from Test to Production

When moving Oracle Access Manager 11g from test to production, you can use one of the methods described in:

- Table A–3, " Full Replication"
- Table A–4, " Golden Template"
- Table A–5, " Delta-Replication"
- Table A–6, " Application Re-association"

Table A–3 describes full replication. By performing manual and automated tasks, you can replicate the OAM test source setup to an OAM production target.

Table A–3 Full Replication

Requirements	Automated and Manual Tasks	Not Required or Processed
■ The user store is already configured for the target OAM Server. ■ The same clients and partners that communicate with the source OAM Server also communicate with the target OAM Server.	The following WLST commands are used: ■ Export: Replicate and export the source application domains and policies. ■ Import: Imports the source configuration and conflict resolution profile to the target Replaces the schema in the target with the schema in the source. Removes application domains in the target system that are not present in the source. An Administrator must also: ■ Replace the schema in the target system with the schema in the source system.	■ Partners ■ Clients ■ OAM Servers

Table A–4 describes golden template processing. By performing manual and automated tasks, you can create a target OAM Server with the identical topology as the source.

Table A–4 Golden Template

Requirements	Manual and Automated Tasks	Not Required or Processed
■ The same clients and partners that communicate with the source also communicate with the target OAM Server. ■ The target must be a clean environment; all configuration in the target is overwritten by the source.	The Administrator must manually: ■ Set up the required target topology to match the existing source topology. ■ Replace the schema in the target system with the schema in the source system, if there are changes in the user configuration. ■ Register target OAM Servers using either the OAM Administration Console or remote registration commands for OAM. ■ Use t2pClient commands to replicate the source on the target. The WLST commands clone the existing OAM test setup and: ■ Configures the target user identity store to match the source user identity store (configureUserStore) ■ Replicates and migrates partner data from the source to the target ■ Exports the source configuration ■ Imports the source configuration with a conflict resolution profile	Any change in user configuration is an independent, manual step.
If a source WebGate is configured with a list of primary/secondary OAM Server hosts and OAP ports that are not included in the target (production) environment, after the transition you might see empty fields or a subset of source primary or secondary servers listed in the target environment.	After running WLST commands, above, the Administrator must also edit target (production) agent registration pages to match actual hosts and ports:	

Table A–5 describes requirements and processing for incremental transfer (known as delta replication). All incremental changes in the source are transferred to the target. Selective transfer is not required.

Table A–5 Delta-Replication

Requirements	Tasks	Not Required or Processed
The source OAM Server contains the "truth". Any conflicts between the source and the target are resolved based on the source.	The Administrator runs the WLST command with the "MigrateAll" flag set to "false" to move only the changes from the source to the target system.	Policy configuration that has not changed is not processed.

Table A–6 outlines requirements and tasks. By performing manual and automated tasks, you can re-associate source client applications to the target environment.

Table A–6 Application Re-association

Requirements	Automated and Manual Tasks	Not Required or Processes
Re-associates partners and clients from the source environment to the target.	The Administrator runs WLST commands to: ■ Replicate and transfer OAM policies from the source to the target system. ■ Create place holder host-identifiers and use these in target application domains.	
	The Administrator uses the remote registration tool to: ■ Register source partners with the target system. ■ Add the agents to target host-identifiers created by WLST commands. ■ Associate an agent with an existing application domain. ■ Create a new policy in the target with the default authentication scheme for the partner application. ■ Associate the partner application with the specified application domain and policy.	

A.3.3 About the WebLogic Scripting Tool Commands

Whether you are moving to a new target, or to an existing target, Oracle provides the WLST commands that use an MBean on the OAM 11g AdminServer and enable administrators to:

- Configure the target user identity store to match the source user identity store, when needed.
- Replicate and move application domain and policy data (for all or for only selected domains and policies).
- Provide a conflict resolution profile that describes how ID conflicts between the source and target systems must be resolved.

Exporting replicates and exports application domains and partner information to a temporary dump file. To protect this sensitive information, a Keystore is generated with the dump file. The key in this Keystore is used to encrypt the dump file.

Table A–7 provides information on export mode commands, which you run on the test source OAM Server that is hosting the partner to be exported.

Table A–7 Export Commands

Command	Description	Example
exportPartners()	Exporting a partner creates an object with all partner information, along with the key for each of the partners. This command takes the path to the temporary oam-partners file as a parameter.	exportPartners(pathTempOAMPartnerFile=', <pathTempOAMPartnerFile>)
exportPolicy-	Exports application domain and policy data from the source. OAM application domains are exported with all dependencies. This command takes the path to the temporary oam-policy file as a parameter.	exportPolicy(pathTempOAMPolicyFile=', <pathTempOAMPolicyFile >')

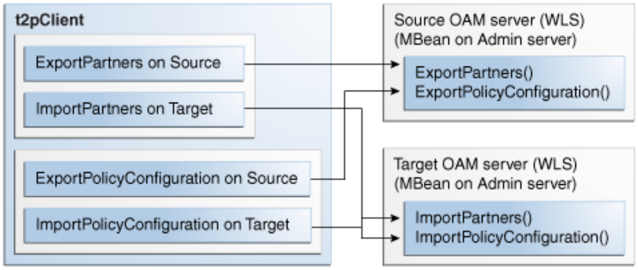
Importing decrypts the generated dump file using the key in the Keystore and imports the dump file contents to the target OAM Server. You can import partners, policies, or policy differences, as described in Table A–8. Import commands are run on the target OAM Server.

Table A–8 Import Commands

Command	Description	Example
importPartners()	Decrypts and imports partner data using the key in the Keystore. This command takes as input the path the temporary oam-partners file as a parameter that was created during the export operation.	importPartners (pathTempOAMPartnerFile=', <pathTempOAMPartnerFile>')
importPolicy-	Decrypts and imports application domain and policy data. **Caution**: This command overwrites all policy data on the target. This command takes as input the path the temporary oam-policy file that was created during the export operation.	importPolicy(pathTempOAMPolicyFile=', <pathTempOAMPolicyFile >')
importPolicyDelta-	Decrypts and only the changes from the source to the target OAM Server without overwriting unchanged policy data on the target. **Note**: This command writes only changed policy data to the target. This command takes as input the path the temporary oam-policy file that was created during the export operation.	importPolicyDelta(pathTempOAMPolicyFile=', <pathTempOAMPolicyFile >'

Figure A–1 illustrates the processing that occurs between the source and target systems.

Figure A–1 Source and Target processing



A.3.4 About Conflict Resolution

The source system is presumed to be the single source of truth during data migration. Any conflicts that are detected between the source system and the target system must be resolved during processing. A sample of the conflict resolution template is shown here and described in Table A–9.

```
<Conflict-Resolution>
   <Authentication-Modules value="Use Existing/Error"/>
   <Authentication-Schemes value="Reuse/Replace/Create New/Error"/>
   <Host-Identifiers value="Add to Existing/Create New"/>
   <Policy value="Reuse/Replace/Create New/Error"/>
   <Application-Domains value="Reuse/Replace/CreateNew/Error"/ >
<Conflict-Resolution>/
```

Table A–9 Conflict Resolution

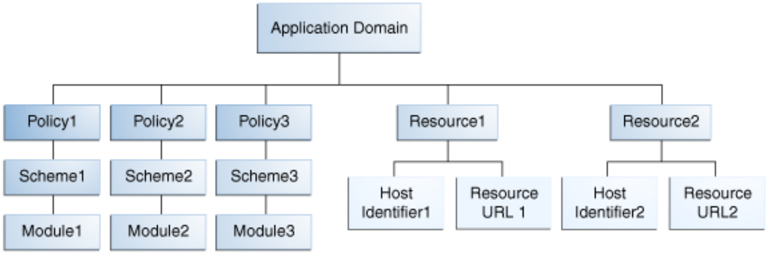
Element	Description
Authentication Modules Authentication-Modules value="Use Existing/Error"/>	The authentication modules are defined for each OAM system. These modules are configured during system configuration. So the migration will not create authentication modules if they are not present in the production system. If the authentication module in the production system does not match the test system, the administrator can choose to use existing authentication module or throw an exception and exit the migration process. The administrator records this choice in the conflict resolution profile.
Authentication Schemes <Authentication-Schemes value="Reuse/Replace/Creat e New/Error"/>	If authentication scheme with the same ID exists in test and production system, the conflict needs to be resolved before migrating. The Authentication scheme has the attributes - Name, Authentication level, Challenge method and Authentication module. All these attributes are used to match the authentication schemes in the test and production systems. If all these attributes match, the migration process will use the existing scheme. However, if any of these attributes don't match, the administrator can choose to create a new authentication scheme with a new unique ID. This choice recorded in the resolution profile.
Host-Identifiers Host-Identifiers value="Add to Existing/Create New"/>	A host can be addressed in multiple ways: host name with the domain name, host name without the domain name and the IP address. A host-identifier is used to group all these addressing methods so that requests matching any of these addressing methods are protected by the OAM Server. For instance, suppose that you need to resolve one of the following types of conflicts between the source system (the single source of truth) and the target system: ■ Authentication Modules ■ Authentication Schemes ■ Host-Identifiers and hosts Host-identifiers are used as a part of the resources in application domains. While exporting the application domains from the test to the production system, the host-identifier conflicts will have to be resolved. If the host-identifier and all the hostnames in that host-identifier exactly match in the test and production systems match, then migration of the application domain continue. However, if the host-identifiers and hostnames do not match then conflict has to be resolved. Here are the situations where the host-identifiers do not match: Host identifier with the same name has a completely different set of hostnames in test and production system Host identifier with the same name in test system has a subset of hostnames in the production system Host identifier with the same name in the production system has a subset of hostnames in the test system Host identifier with the same hostnames has different names.

A.3.5 About Building a Dependency Tree for Each Application Domain

Before migrating an OAM 11g application domain, a dependency tree must be constructed for each of the application domains to be migrated.

The dependency tree can be represented as shown in Figure A–2.

Figure A–2 Dependency Tree for Each Application Domain



In the sample dependency tree shown in Figure A–2, the application domain consists of three authentication policies and two resources. Each authentication policy is configured with an authentication scheme and each of authentication scheme has an authentication module configured. This sample application domain applies to two resources (each resource is defined as a host identifier and a resource URL).

To migrate data for an application domain, the shared components (Modules, Schemes and Host-identifiers) must be migrated first, if they are not already migrated. Shared component data migration is followed by application domain data migration.

A.4 Planning an OAM 11g Move from Test to Production

Planning and preparation are key components of any successful data transfer strategy. This section discusses the planning considerations and inventory items that you and your team need to create to ensure your success:

- Noting Differences Between Source and Target Environments
- Developing Deployment Inventories
- Developing Tests
- Understanding Change Propagation
- Scheduling and Notifications

A.4.1 Choose the Method

Review details in "About Methods to Move from Test to Production" on page A-4 and choose the method that best suits your needs as described in:

- Table A–3, " Full Replication"
- Table A–4, " Golden Template"
- Table A–5, " Delta-Replication"
- Table A–6, " Application Re-association"

A.4.2 Noting Differences Between Source and Target Environments

When transferring Oracle Access Manager configuration data from a source to a target, be sure to note the following types of differences between the two environments:

- Names and implementation details of OAM Server instances
- Names and implementation details of OAM Agents (WebGates, and AccessGates) including changing the OAM Server to which the Agent points.
- Names and implementation details of OSSO Agents (mod_osso) including changing the OAM Server to which the Agent points
- Definitions for Host Identifiers
- Definitions for authentication schemes, including Challenge Redirect parameters.
- Definitions for authorization policies, constraints, responses, and resources
- Definitions for application domains, including all redirect URLs defined in authentication and authorization policies

A.4.3 Developing Deployment Inventories

Before starting any transfer activities, Oracle recommends that you take inventory of your existing Oracle Access Manager 11g Release 1 (11.1.1) deployment. You can gather details from existing installation records or you can gather fresh information directly from the deployment.

A.4.4 Developing Tests

To help ensure data correctness before transfer, Oracle recommends that you develop specific tests that evaluate configuration in the source deployment.

After transfer, you can use these same tests in the target deployment to ensure that everything is working as expected.

A.4.5 Understanding Change Propagation

All changes are reflected in the OAM Administration Console and are automatically propagated to every OAM Server in the cluster.

When you have a single OAM Server and a single OAM Administration Console running on different computers, changes are propagated to the managed run-time OAM Server.

A.4.6 Scheduling and Notifications

Before starting any move, Oracle strongly recommends that you and your team schedule specific transfer windows and that you notify other administrators about planned activities in any deployment for which they are responsible.

A.5 Backup and Recovery Strategies

Oracle recommends that you back up data before transfer, and restore the backup after transfer if needed.

A.6 Moving OAM 11g From Test to Production

This section is divided into the following based on your needs:

- Exporting OAM 11g Data from Test (Source)
- Importing OAM 11g to Production (Target)

A.6.1 Exporting OAM 11g Data from Test (Source)

Use steps in the following procedure as needed to export partner and policy data from the test source environment.

Prerequisites

Planning an OAM 11g Move from Test to Production

See:

- "Introduction to Methods and Tools" on page A-3
- Appendix F, "Introduction to Custom WLST Commands for OAM Administrators"
- *Oracle Fusion Middleware Administrator's Guide* for details about moving Oracle Access Manager 10g to a new (or an existing) production environment

To export data from the test source

1. **Export Partner Data**: On the source OAM Server hosting the OAM 11g partner run the following t2pClient command using the path to your own temporary OAM partners file. For example:

   ```
   exportPartners(pathTempOAMPartnerFile=', <pathTempOAMPartnerFile>>')
   ```

2. **Export Policy Data**: On the source OAM Server hosting the OAM 11g policy data, run the following t2pClient command using the path to your own temporary OAM policy file. For example:

   ```
   exportPolicy(pathTempOAMPolicyFile=', <pathTempOAMPolicyFile >')
   ```

3. Repeat on each source OAM Server hosting partner and policy data.

A.6.2 Importing OAM 11g to Production (Target)

Use steps in the following procedure as needed to import partner and policy data from the test source environment.

See Also: "Introduction to Methods and Tools" on page A-3

Prerequisites

Exporting OAM 11g Data from Test (Source)

To import data to the production target

1. **Import Partner Data**: On the target OAM Server, run the following t2pClient command using the path to the temporary source partners file. For example:

   ```
   importPartners(pathTempOAMPartnerFile=', <pathTempOAMPartnerFile>>')
   ```

2. **Import Full Policy Data**: On the target OAM Server, run the following t2pClient command using the path to the temporary source policy file. For example:

   ```
   importPolicy(pathTempOAMPolicyFile=', <pathTempOAMPolicyFile >')
   ```

3. **Import Only the Policy Delta**: On the target OAM Server, run the following t2pClient command using the path to the temporary source policy file. For example:

   ```
   importPolicyDelta(pathTempOAMPolicyFile=', <pathTempOAMPolicyFile >')
   ```

4. Repeat on each source OAM Server hosting partner and policy data.

B

Co-existence Overview: OAM 11g and OSSO 10g

You can upgrade an existing OracleAS SSO 10g Release (10.1.2.0.2) through OracleAS SSO 10g Release (10.1.4.3.0) to Oracle Access Manager 11g. This chapter explains the co-existence that is provided when upgrading OracleAS 10g SSO (OSSO) to use Oracle Access Manager 11g SSO. It includes the following sections:

- Prerequisites
- Introduction to Upgrading and Co-existence with OracleAS 10g SSO
- Pre- and Post-Upgrade Topology and Authentication Examples
- Introduction to Validating Post-Upgrade Co-Existence with OAM 11g
- Validating Post-Upgrade Co-existence

B.1 Prerequisites

See *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management* for upgrade steps.

B.2 Introduction to Upgrading and Co-existence with OracleAS 10g SSO

Oracle uses the term "upgrade" when referring to moves between Oracle product versions and technologies. For instance, a move from OC4J to Oracle WebLogic Server is an upgrade; moving from OracleAS 10g SSO to OAM 11g SSO is an upgrade.

> **Note:** Oracle uses the term "migration" for moves from a non-Oracle technology stack to an Oracle technology stack.

The Oracle-provided Upgrade Assistant scans the existing OracleAS 10g SSO server configuration, accepts as input the 10g OSSO policy properties file and schema information, and transfers configured partner applications into the destination Oracle Access Manager 11g SSO.

> **See Also:** *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management*

A typical OSSO deployment includes a number of OSSO servers with a front-end load balancer. All OSSO servers in the cluster have the same back-end store. The load balancer routes authentication requests to any OSSO servers in the cluster.

Co-existence requires OAM 11g and OSSO 10g to use the same back-end user identity store: Oracle Internet Directory (OID). OAM 11g co-exists with the single OSSO Server or cluster of OSSO servers. Dynamic redirection must work as expected for applications protected by OSSO 10g and OAM 11g.

During the upgrade process, partner applications registered with OSSO 10g are transferred to OAM 11g along with associated Oracle HTTP Server Agents, corresponding host identifiers, and other details. OAM 11g is added to the front-end load balancer of existing OSSO 10g Servers.

> **Note:** If an existing OSSO 10g configuration has out-of-the box configurations that cannot be mapped directly, an Administrator must manually transfer these after automated upgrade processes finish.

After upgrading, OAM 11g co-exists with either a single OSSO Server or cluster of OSSO servers. Existing partner applications (including Portal, Forms, Reports, and Discoverer) start using Oracle Access Manager 11g as the SSO provider. The load balancer routes some of the authentication request to the OAM Server while the rest continue to be served by the existing OSSO 10g Servers. Once the user is authenticated by either the OSSO 10g or OAM 11g Server, the user can access any of the partner applications without having to re-authenticate.

For more information, see "Pre- and Post-Upgrade Topology and Authentication Examples".

B.3 Pre- and Post-Upgrade Topology and Authentication Examples

This section provides the following topics:

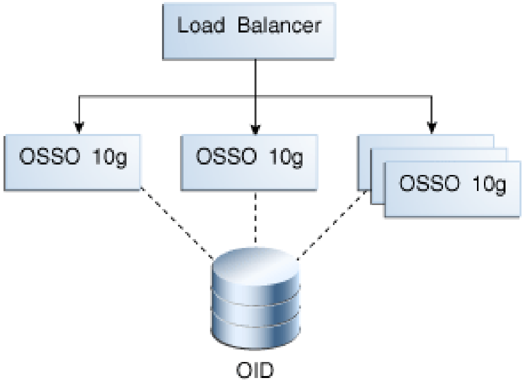
- About Pre-Upgrade OSSO 10g Topology
- About Post-Upgrade Topology and Co-existence
- Simple OSSO 10g with mod_oc4j on a Front-End Proxy Server
- Post-Upgrade: mod_wl Replaces mod_oc4j on the Proxy Server
- Post-Upgrade: No Proxy Server

> **See Also:** "Introduction to Validating Post-Upgrade Co-Existence with OAM 11g" on page B-5

B.3.1 About Pre-Upgrade OSSO 10g Topology

A typical OSSO set up has a number of OSSO Servers with a front-end load balancer. All OSSO Servers in the cluster have the same back-end user identity store. The load balancer routes authentication requests to any of the OSSO servers in this cluster, as shown in Figure B–1.

Figure B–1 Pre-Upgrade OSSO 10g Topology

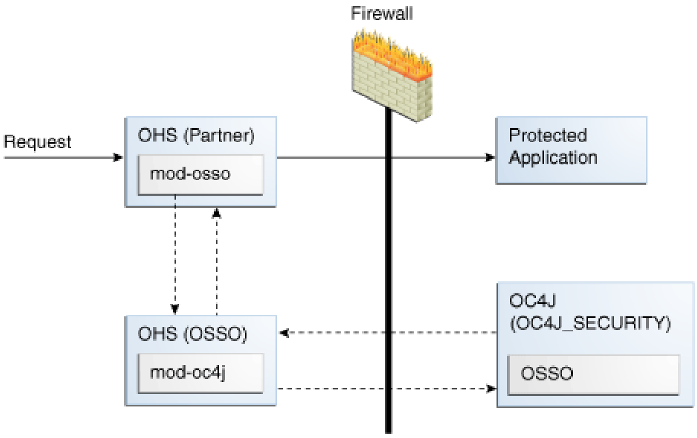


See Also: "Simple OSSO 10g with mod_oc4j on a Front-End Proxy Server"

B.3.1.1 Simple OSSO 10g with mod_oc4j on a Front-End Proxy Server

Figure B–2 illustrates a simple situation where the OHS (Partner) front-ends the protected application. OHS (OSSO) is the front-end proxy Web server protecting the OC4J OSSO application server host. This is needed only if there is an OSSO OC4J server behind it.

Figure B–2 Pre-Upgrade Sample OSSO 10g with Front-End Proxy



After upgrading the environment is configured to use Oracle Access Manager 11g for authentication.

B.3.2 About Post-Upgrade Topology and Co-existence

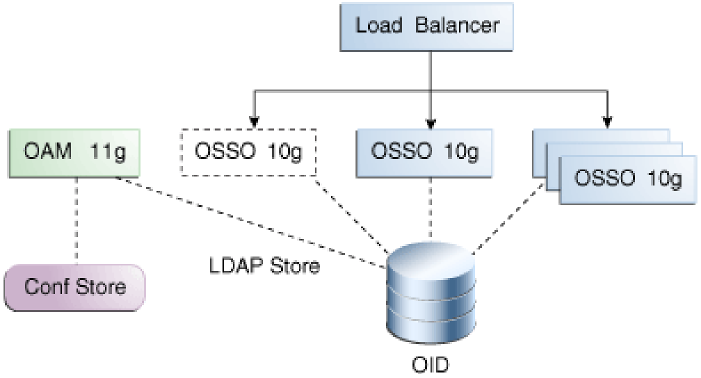
Upgrading to OAM 11g starts by installing a new OAM 11g Server and transferring the partner applications to this OAM Server.

One of the existing OSSO Servers is brought down. The OAM 11g Server replaces the downed OSSO Server and the load balancer starts routing authentication requests to the newly added OAM 11g Server (and continues routing authentication requests to remaining OSSO 10g Servers).

Note: Over time, each of the OSSO 10g servers should be replaced by OAM 11g Servers.

The upgraded OSSO set up is shown in Figure B–3.

Figure B–3 Post-Upgrade OSSO 10g Topology



To provide Single Sign On for the user to access any of the partner applications, OAM 11g accepts the user authenticated by OSSO server as an authenticated user. Also, when OAM 11g validates a user it also sets appropriate cookies that the OSSO server can understand. The OSSO server does not need to validate the user again.

Once the user is authenticated by either OSSO 10g or OAM 11g, the user can access all the partner applications protected by either server. OAM 11g and OSSO 10g set appropriate cookies to achieve single sign on.

OAM 11g creates a session for each request and sets a cookie that contains this session ID. The session represented by this cookie has JAAS subject of the authenticated user among other details.

Note: With OSSO 10g, the server sets a host cookie that contains information about the logged in user.

For additional information, see:

- Post-Upgrade: mod_wl Replaces mod_oc4j on the Proxy Server
- Post-Upgrade: No Proxy Server

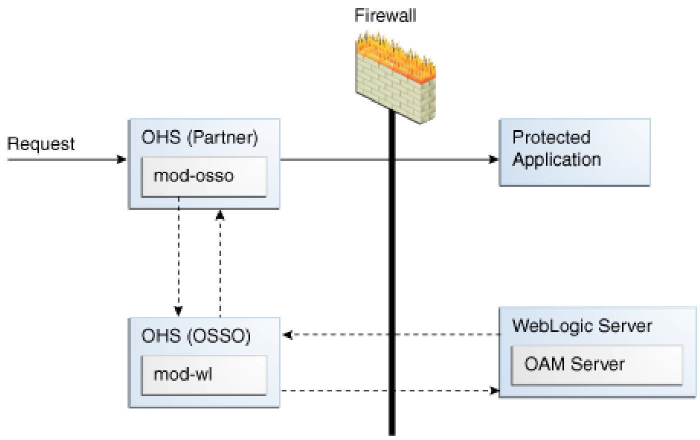
B.3.2.1 Post-Upgrade: mod_wl Replaces mod_oc4j on the Proxy Server

In this post-upgrade view, Oracle Access Manager 11g is used for authentication. The Oracle HTTP Server front-ending the OSSO 10g Server points to the Oracle WebLogic Server where Oracle Access Manager 11g is installed. This means that:

- OHS Partner: mod_osso redirects requests to the 11g OAM Server for authentication through a proxy.
- Proxy: mod_wl replaces mod_oc4j. mod_wl enables single sign-on to work without any changes on the Oracle HTTP Server side.

Figure B–4 illustrates this post-upgrade topology with mod_wl replacing mod_oc4j.

Figure B–4 mod_wl Replaces mod_oc4j on the Proxy Server



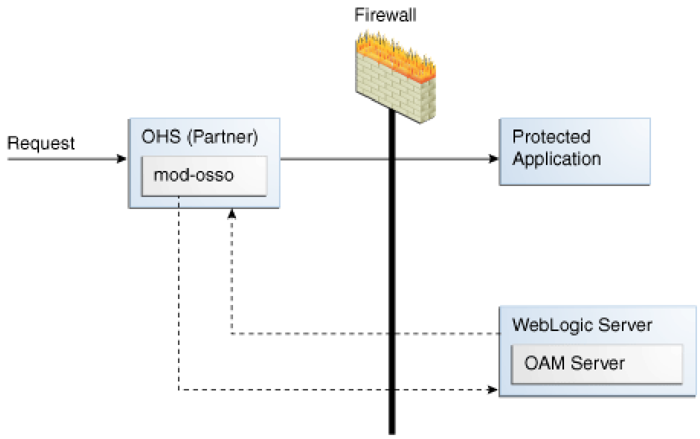
See Also: "Post-Upgrade: No Proxy Server"

B.3.2.2 Post-Upgrade: No Proxy Server

In this example, the proxy server that once included mod_oc4j has been eliminated entirely. The Oracle HTTP Server front-ending the 10g OSSO Server points directly to the Oracle WebLogic Server where Oracle Access Manager 11g is installed.

Figure B–5 illustrates this topology, which most deployments will use.

Figure B–5 Typical Topology Without Proxy Server



B.4 Introduction to Validating Post-Upgrade Co-Existence with OAM 11g

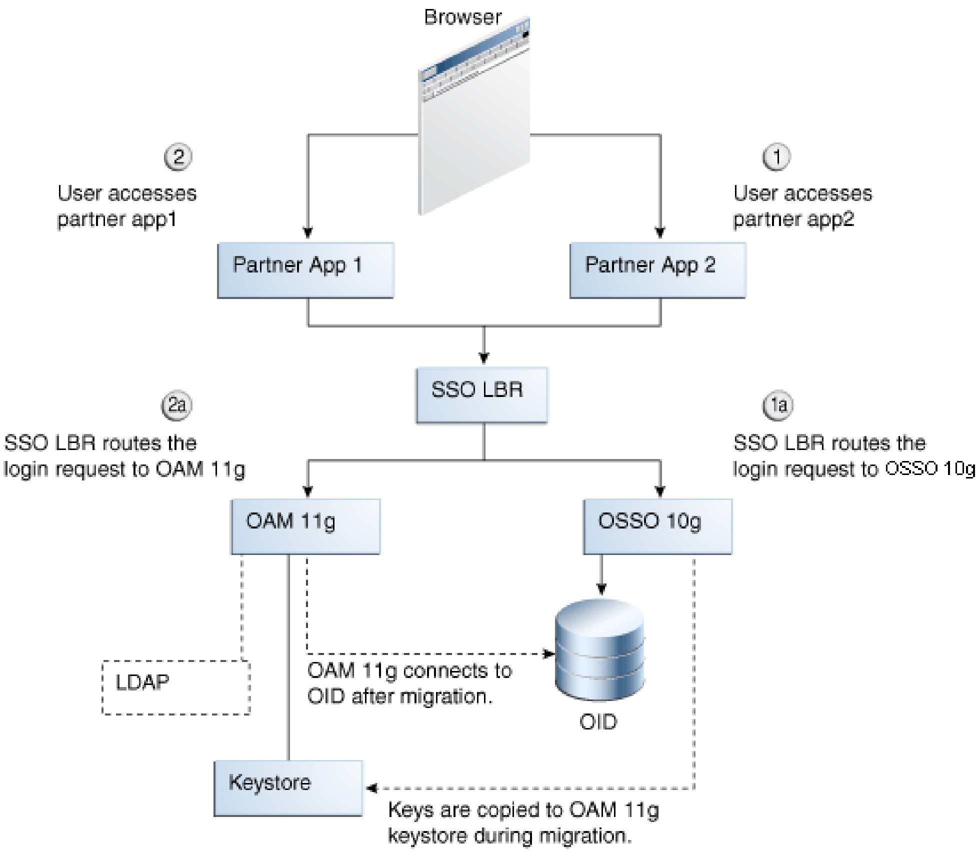
This section provides the following topics:

- About Post-Upgrade SSO
- About Post-Upgrade OSSO 10g Authentication

B.4.1 About Post-Upgrade SSO

Figure B–6 illustrates authentication by OAM 11g SSO in an upgraded environment where OAM 11g co-exists with OracleAS 10g SSO. Details follow the figure.

Figure B–6 Co-existence Processing


Browser
2
User accesses partner app1
1
User accesses partner app2
Partner App 1
Partner App 2
SSO LBR
2a
SSO LBR routes the login request to OAM 11g
1a
SSO LBR routes the login request to OSSO 10g
OAM 11g
OSSO 10g
LDAP
OAM 11g connects to OID after migration.
OID
Keystore
Keys are copied to OAM 11g keystore during migration.

Process overview: Single Sign On between partner applications

1. User accesses a partner application that has been upgraded from OSSO 10g to OAM 11g.

   The load balancer routes the user's authentication request to the OSSO 10g server, which serves the login page. After successful authentication, OAM sets the SSO cookie and updates session information accordingly. The cookie includes a flag indicating that an OSSO cookie must also exist for this cookie to be valid.

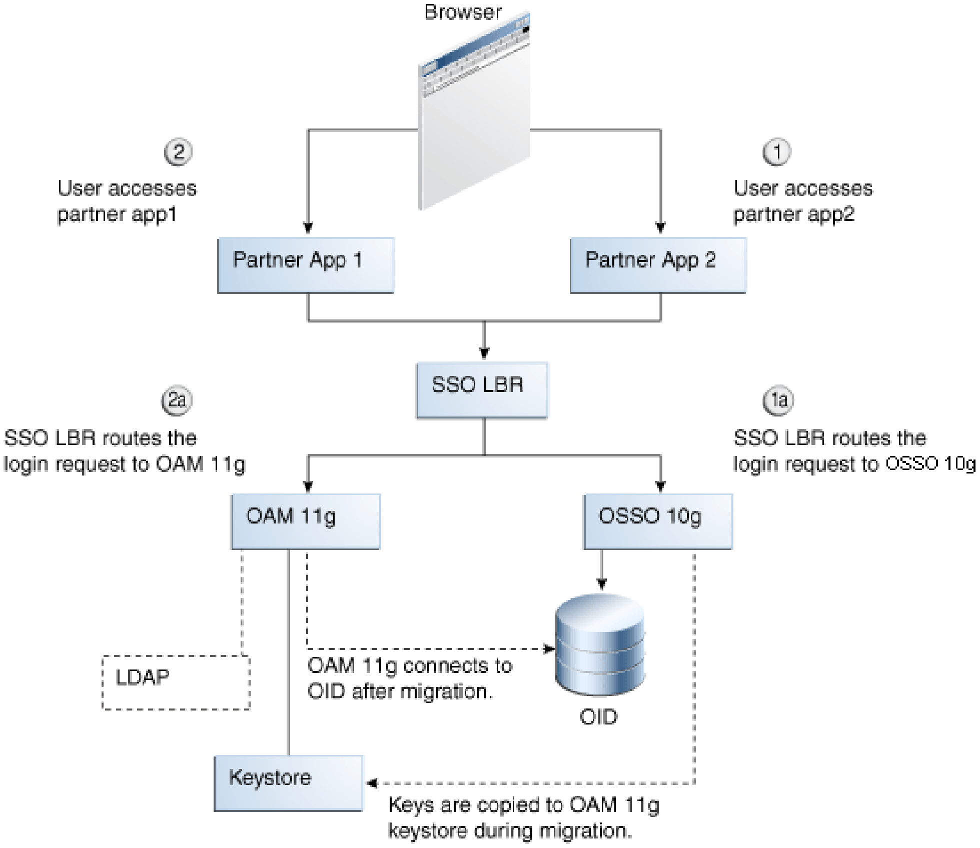
2. When the user accesses an application protected by OSSO 10g an OSSO 10g cookie is already set in the browser, and the user is not challenged for credentials and can access the application.

3. Both the OAM 11g and OSSO 10g cookie must be kept in sync. Any session information that is updated in the OSSO 10g cookie must be synchronized with the OAM 11g cookie and vice-versa.

B.4.2 About Post-Upgrade OSSO 10g Authentication

As shown in Figure B–6, the user accesses the partner application and the user's authentication request is routed to OSSO 10g by the load balancer. Here, the OSSO server displays the login page and after collecting and validating user credentials, only the OSSO cookie is set in the user's browser.

Figure B–7 Co-existence and OSSO 10g Authentication


Browser
2
User accesses partner app1
1
User accesses partner app2
Partner App 1
Partner App 2
SSO LBR
2a
SSO LBR routes the login request to OAM 11g
1a
SSO LBR routes the login request to OSSO 10g
OAM 11g
OSSO 10g
LDAP
OAM 11g connects to OID after migration.
OID
Keystore
Keys are copied to OAM 11g keystore during migration.

Process overview: Post-upgrade OSSO 10g Authentication

1. When the user accesses the partner application protected by the OAM 11g server, the OAM server must first check if a valid OSSO cookie already exists before throwing the login page.

2. If a valid OSSO cookie exists already, the OAM 11g server must create an OAM 11g SSO cookie from the OSSO cookie. The OAM Server configuration has a flag which says whether the coexist mode is enabled or not. If the coexist mode is enabled, then the OAM Server looks for the OSSO cookie to be present along with OAM 11g Server's SSO Cookie. This flag can be either turned on manually in the configuration followed by a server restart or a WLST command can be used to turn the coexist flag on/off.

3. When the user logs out of a partner application and the logout request is routed to the 10g OSSO Server, the 10g OSSO Server deletes the OSSO cookie. With coexistence enabled, with both 10g SSO and 11g OAM Servers behind a loadbalancer, the partner information is shared by all the Servers (10g OSSO or 11g OAM Servers). Therefore, associating one of the servers (10g or 11g) with the partner application is not correct. During the upgrade, all the partner information is also migrated to the 11g OAM Server.

4. After the OSSO Cookie has been deleted, if the user tries to access a protected application, and if the request this time goes to the OAM 11g Server, only the

OAM Server's SSO Cookie is found. The Server learns from the coexist flag in its configuration that the setup is in coexist mode. In coexist mode, the OSSO cookie needs be present for the OAM11g Server Cookie to be valid. Hence the login page is thrown and the user is asked for validation.

5. The OAM 11g server must make sure that the session information in both the OSSO and OAM cookies are in sync.

B.5 Validating Post-Upgrade Co-existence

This section provides the following topics:

- Validating Post-Upgrade Registration and Policies
- Validating Post-Upgrade SSO with Oracle Access Manager Protected Resources
- Validating Post-Upgrade SSO with OSSO-Protected Resources

B.5.1 Validating Post-Upgrade Registration and Policies

The following topics provide information that help you locate transferred OSSO 10g details in the Oracle Access Manager 11g Administration Console:

- Sample Partner Applications Protected Using OSSO 10g
- Policy Enforcement Agent Details
- Shared Components: Host Identifiers for migratedSSOPartners
- Resources in the migratedSSOPartners Application Domain
- Authentication Policy in the migratedSSOPartners Application Domain

> **See Also:** "Introduction to Validating Post-Upgrade Co-Existence with OAM 11g" on page B-5

B.5.1.1 Sample Partner Applications Protected Using OSSO 10g

Details of the sample partner applications that use OSSO 10g are provided here to help you compare the OSSO 10g configuration with the upgraded configuration for Oracle Access Manager 11g.

Table B–1 shows the applications protected by OracleAS 10g OSSO.

Table B–1 Partner Applications Protected by OSSO 10g

Application Name	Host	Oracle Home
oid1_ad2003_lowenthal.vm	ad2003.lowenthal.vm	C:\oracle\oid
portal1_ad2003_lowenthal.vm	ad2003.lowenthal.vm	C:\oracle\portal
Oracle Portal (portal)	ad2003.lowenthal.vm	C:\oracle\portal

OSSO 10g configuration details for each application includes administrator-assigned:

- Name
- Home Page URL
- Success URL
- Logout URL
- Date range for which login to the application is allowed by the server

OSSO 10g configuration also includes:

- Unique Application ID
- Application Token used by the partner when requesting authentication
- Encryption Key used by the OSSO Server to identity the application
- Login URL
- Single Log-Out URL

B.5.1.2 Policy Enforcement Agent Details

For each application in the OracleAS 10g SSO deployment (Table B–1), there is an Oracle HTTP Server instance. Each OHS instance transfers as an OSSO Agent that is named after the application. In the Oracle Access Manager 11g Administration Console, you can locate each transferred OSSO Agent configuration under the System Configuration tab, Agents node in the navigation tree.

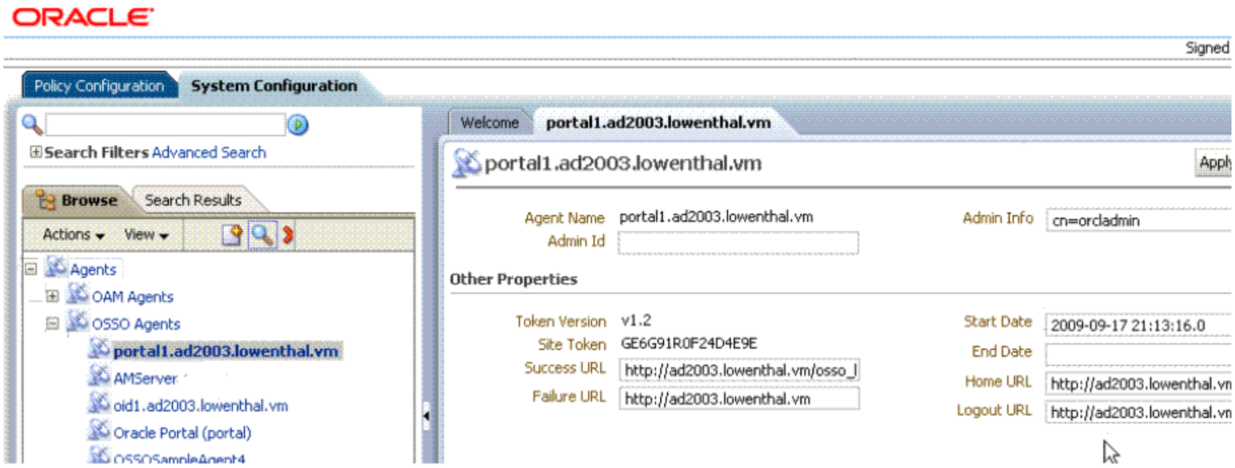
Figures in this topic illustrate the transferred OSSO Agent configurations for the applications identified in Table B–1. Each generated Agent configuration is named as the application it protects.

While the 10g Application ID is not recorded in the Agent configuration for OAM 11g, most configuration details are included and remain the same:

- Site Token: The application token.
- Login URL
- Single Log-Out URL
- Success URL
- Failure URL
- Start Date

Figure B–8 illustrates the generated Agent configuration for the first OSSO 10g-protected application. Details were derived from the OSSO 10g Oracle Internet Directory during automated upgrade processing.

Figure B–8 OSSO Agent Configuration Named for One Application



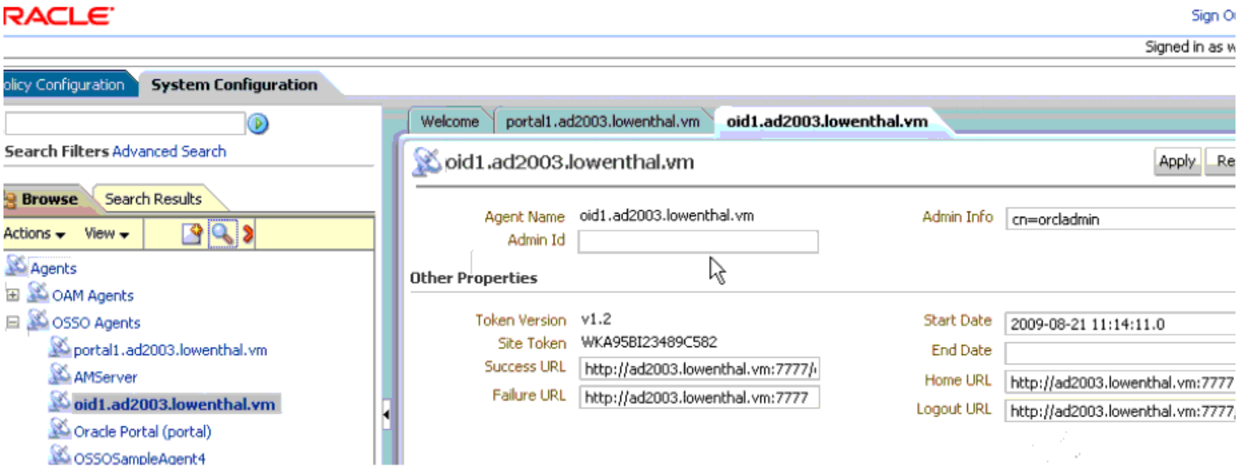
portal1.ad2003.lowenthal.vm

Agent Name portal1.ad2003.lowenthal.vm
Admin Id
Admin Info cn=orcladmin

Token Version v1.2
Site Token GE6G91R0F24D4E9E
Success URL http://ad2003.lowenthal.vm/osso_l
Failure URL http://ad2003.lowenthal.vm
Start Date 2009-09-17 21:13:16.0
End Date
Home URL http://ad2003.lowenthal.vn
Logout URL http://ad2003.lowenthal.vn

Figure B–9 shows the transferred configuration for a second application protected by OracleAS 10g SSO (OSSO).

Figure B–9 OSSO Agent Configuration Named for the Second Application



oid1.ad2003.lowenthal.vm

Agent Name oid1.ad2003.lowenthal.vm

Admin Id

Admin Info cn=orcladmin

Token Version v1.2

Site Token WKA95BI23489C582

Success URL http://ad2003.lowenthal.vm:7777/

Failure URL http://ad2003.lowenthal.vm:7777

Start Date 2009-08-21 11:14:11.0

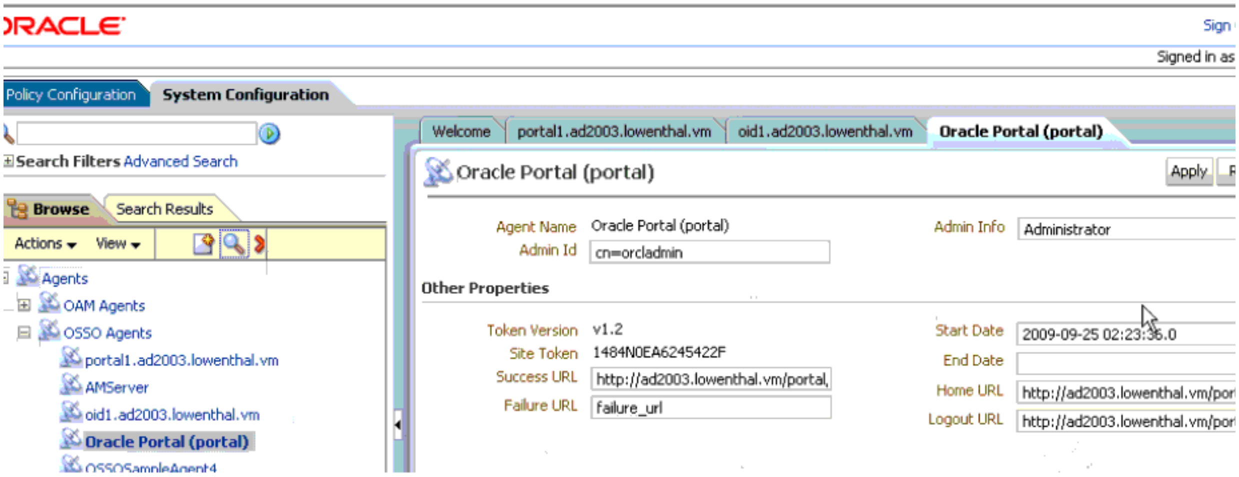
End Date

Home URL http://ad2003.lowenthal.vm:7777

Logout URL http://ad2003.lowenthal.vm:7777,

Figure B–10 shows the transferred OSSO Agent configuration for the third application protected with OracleAS 10g SSO (OSSO).

Figure B–10 OSSO Agent Configuration Named for the Third Application



Oracle Portal (portal)

Agent Name Oracle Portal (portal)

Admin Id cn=orcladmin

Admin Info Administrator

Token Version v1.2

Site Token 1484N0EA6245422F

Success URL http://ad2003.lowenthal.vm/portal,

Failure URL failure_url

Start Date 2009-09-25 02:23:36.0

End Date

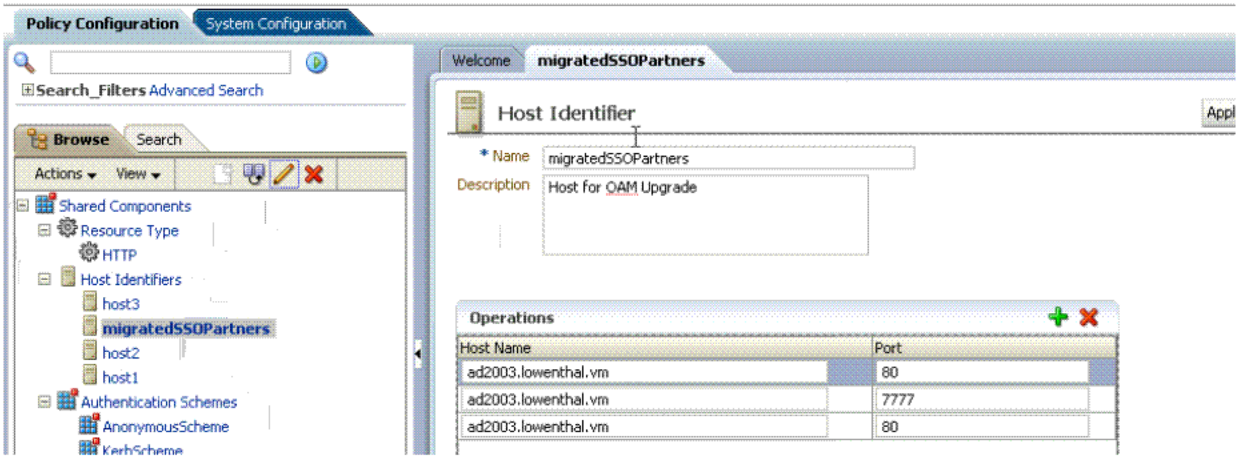
Home URL http://ad2003.lowenthal.vm/por

Logout URL http://ad2003.lowenthal.vm/por

B.5.1.3 Shared Components: Host Identifiers for migratedSSOPartners

Figure B–11 illustrates the transferred Host Identifier. It is named `migratedSSOPartners` because this is the Application Domain name in which it is used. Details were derived from the OSSO 10g Oracle Internet Directory during automated transfer processing.

Figure B–11 Host Identifier for migratedSSOPartners



Host Identifier

* Name: migratedSSOPartners

Description: Host for OAM Upgrade

Operations

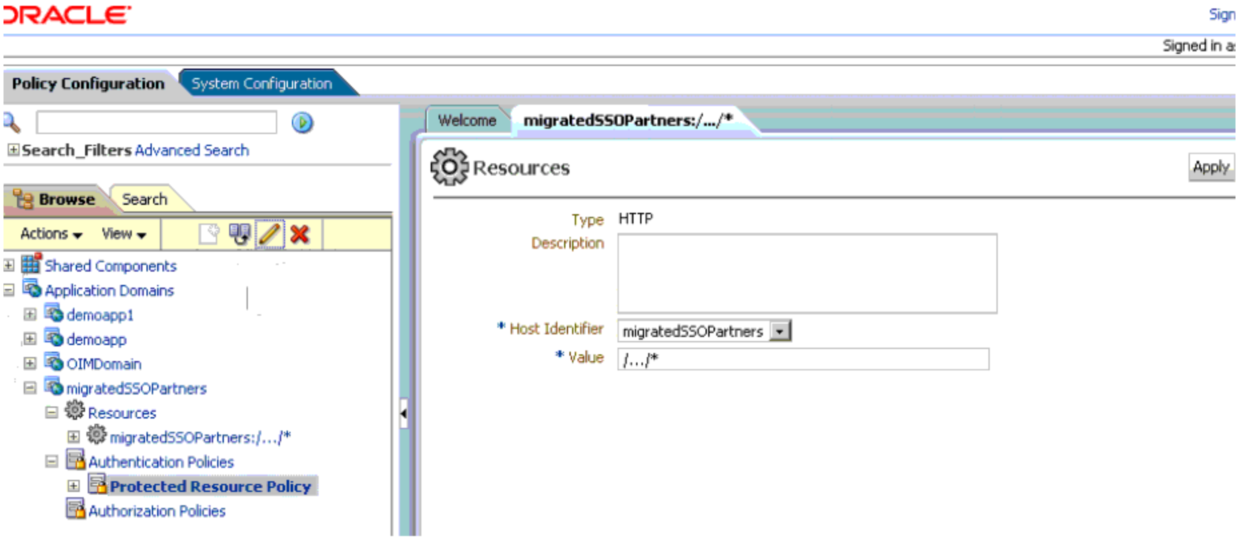
Host Name	Port
ad2003.lowenthal.vm	80
ad2003.lowenthal.vm	7777
ad2003.lowenthal.vm	80

B.5.1.4 Resources in the migratedSSOPartners Application Domain

Figure B–12 illustrates the Application Domain created during the transfer and the resource definition. Details were derived from the OSSO 10g Oracle Internet Directory during automated transfer processing.

Both the Application Domain and the Resources definition are named `migratedSSOPartners`.

Figure B–12 Resources in the migratedSSOPartners Application Domain



Resources

Type: HTTP

Description:

* Host Identifier: migratedSSOPartners

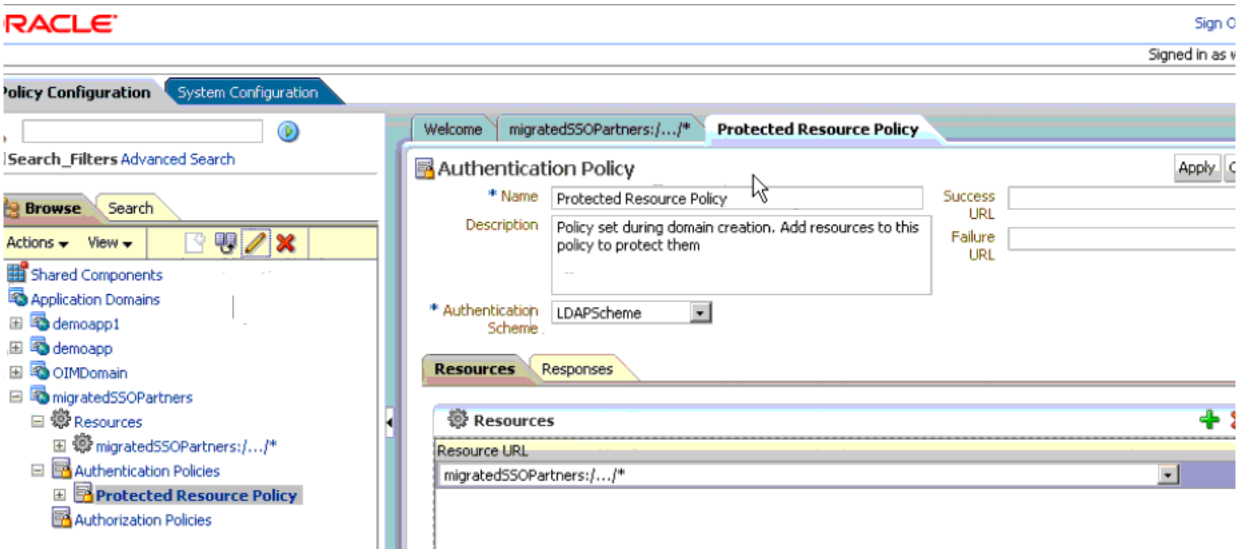
* Value: /.../*

B.5.1.5 Authentication Policy in the migratedSSOPartners Application Domain

Figure B–13 illustrates the Authentication Policy, named `Protected Resource Policy`, for the Application Domain `migratedSSOPartners`.

The default OAM 11g authentication scheme is used: LDAPScheme.

Figure B–13 Authentication Policy for the Application Domain migratedSSOPartners



Authentication Policy

* Name: Protected Resource Policy

Description: Policy set during domain creation. Add resources to this policy to protect them

* Authentication Scheme: LDAPScheme

Resource URL: migratedSSOPartners:/.../*

B.5.2 Validating Post-Upgrade SSO with Oracle Access Manager Protected Resources

You can use the following steps to confirm that single sign-on is occurring in the upgraded environment using Oracle Access Manager 11g.

Perform steps in "Validating Post-Upgrade SSO with Oracle Access Manager Protected Resources" to confirm that OAM 11g protected resources are being accessed through OAM 11g.

To confirm SSO is operational with Oracle Access Manager 11g

1. Stop the Oracle HTTP Server on the computer that is hosting the 10g OSSO server.
2. Restart the 11g OAM Server.
3. In the Oracle Access Manager 11g Administration Console:
   - System Configuration, Agents: Confirm that the upgraded 10g partner applications have appropriate Agent configurations.
   - Policy Configuration, Shared Components: Confirm that a new Host Identifier definition was created: `migratedssopartners`.
   - Policy Configuration, Application Domain:
     - Confirm that a new Application domain was created: `migratedssopartners`
     - Within the new Application domain, confirm that Resources exist for `migratedssopartners`
     - Within the new Application domain, confirm that an Authentication Policy exists with the appropriate Host Identifier: `Protected Resource Policy`
4. Access the partner application and confirm that authentication occurs through Oracle Access Manager 11g (check the login form for 11g).
5. Proceed as follows:

- Success: Continue with "Validating Post-Upgrade SSO with OSSO-Protected Resources".
- No Success: Confirm that you have completed all steps as needed.

B.5.3 Validating Post-Upgrade SSO with OSSO-Protected Resources

You can use the following steps to confirm that single sign-on is occurring after the upgrade in an environment that includes OSSO 10g-protected resources and Oracle Access Manager 11g-protected resources.

To confirm post-upgrade SSO with OSSO-protected resources

1. **OAM-Protected Resources**: Perform steps in "Validating Post-Upgrade SSO with Oracle Access Manager Protected Resources" to confirm that OAM 11g protected resources are being accessed through OAM 11g.
2. **OSSO-Protected Resources**: Perform the following steps to confirm that OSSO-protected resources are being accessed through OSSO 10g:
    a. Stop the OAM Server.
    b. Restart the Oracle HTTP Server on the computer that is hosting the 10g OSSO Server.
    c. Access an OSSO-protected resource to confirm that the 10g OSSO server is authenticating (10g OSSO login page).
3. Restart the OAM Server.

C

Integrating Oracle ADF Applications with Oracle Access Manager 11g SSO

This chapter provides information to help you integrate with Oracle Access Manager 11g any Oracle ADF applications within the same Identity Management domain.

This chapter provides the following topics:

- Introduction to Oracle Platform Security Services and Oracle Application Developer Framework
- Integrating OAM 11g With Web Applications Using Oracle ADF Security and the OPSS SSO Framework
- Confirming Application-Driven Authentication During Runtime

C.1 Introduction to Oracle Platform Security Services and Oracle Application Developer Framework

This section provides the following topics:

- Oracle Platform Security Services Single Sign-on Framework
- Oracle Application Developer Framework

C.1.1 Oracle Platform Security Services Single Sign-on Framework

A single sign-on (SSO) solution must provide a standard way for applications to login and logout users. After successful authentication, the SSO service is responsible to redirect the user to the appropriate URL.

The Oracle Platform Security Services (OPSS) SSO Framework provides a way to integrate applications in a domain with an SSO solution. Specifically, it provides applications with a common set of APIs across SSO products to handle login, auto login, and logout.

The Oracle Application Developer Framework (ADF) and applications that are coded to Oracle ADF standards interface with the OPSS SSO Framework. For more information about Oracle ADF, see "Oracle Application Developer Framework" on page C-2.

The Oracle Access Manager SSO solution is available out-of-the-box and provides the following to applications that are coded to Oracle ADF standards and the OPSS SSO Framework:

- Login (application-driven): Upon accessing a part of a secured artifact that requires authentication, the application triggers authentication and redirects the user to be authenticated by the appropriate solution.
- Auto login: A user who has initially accessed an application anonymously registers an account with the application (Oracle Identity Manager, for instance); upon a successful registration, the user is redirected to the authentication URL; the user can also be automatically logged in without being prompted.
- Global logout: When a user logs out of one application, the logout propagates across to any other application that is enabled by the solution.

> **Note:** The OPSS SSO framework does not support multi-level authentication.

For more information about choosing an SSO solution, and the Oracle Access Manager 10g solution, see *Oracle Fusion Middleware Application Security Guide*, chapter 11, "Configuring Single Sign-On in Oracle Fusion Middleware."

C.1.2 Oracle Application Developer Framework

The Oracle Application Development Framework is an end-to-end application framework that builds on Java EE standards and open-source technologies to simplify and accelerate implementing service-oriented applications.

The development and run-time environment required to deploy and manage ADF applications is similar in many ways to the environment required for other Java EE applications.

The difference between a typical Java EE environment and an environment that supports Oracle ADF applications is the availability of the Oracle ADF run-time libraries:

- In Oracle Fusion Middleware 11g, an Oracle WebLogic Server domain, by default, does not contain the Oracle ADF run-time libraries. However, you can optionally configure or extend your domain to include the Java Run-time Files (JRF). The Oracle ADF run-time libraries are included as part of the JRF component.

  The Oracle WebLogic Server domain can be extended with the Java Run-time Files (JRF) domain template, which includes the required Oracle ADF libraries, and other important Oracle-specific technologies.
- In Oracle Application Server 10g, each instance of OC4J automatically provided the Oracle ADF run-time libraries required to support Oracle ADF applications.

For information about the types of Java EE environments available in 10g and instructions for upgrading those environments to Oracle Fusion Middleware 11g, refer to the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

C.2 Integrating OAM 11g With Web Applications Using Oracle ADF Security and the OPSS SSO Framework

This section describes how to integrate a Web application that uses Oracle ADF security and the OPSS SSO Framework with an Oracle Access Manager 11g SSO security provider for user authentication.

Before the Web application can be run, you must configure the domain-level `jps-config.xml` file on the application's target Oracle WebLogic Server for the Oracle Access Manager security provider.

The domain-level `jps-config.xml` file is in the following path and should not be confused with the deployed application's jps-config.xml file:

```
domain_home/config/fmwconfig/jps-config.xml
```

Note: Do not confuse the domain-level `jps-config.xml` file with the deployed application's jps-config.xml file.

You can use an Oracle JRF WLST script to configure the domain-level jps-config.xml file, either before or after the Web application is deployed. This Oracle JRF WLST script is named as follows:

Linux: wlst.sh

Windows: wlst.cmd

The Oracle JRF WLST script is available in the following path if you are running through JDev:

```
$JDEV_HOME/oracle_common/common/bin/
```

In a standalone JRF WebLogic installation, the path is:

```
$Middleware_home/oracle_common/wlst
```

Note: The Oracle JRF WLST script is required. When running WLST for Oracle Java Required Files (JRF), do **not** use the WLST script under $JDEV_HOME/wlserver_10.3/common/bin.

Command Syntax

```
addOAMSSOProvider(loginuri, logouturi, autologinuri)
```

Table C–1 defines the expected value for each argument in the addOAMSSOProvider command line. addOAMSSOProvider

Table C–1 addOAMSSOProvider Command-line Arguments

Argument	Definition
loginuri	Specifies the URI of the login page **Note**: For ADF security enabled applications, "/<context-root>/adfAuthentication" should be provided for the 'loginuri' parameter. Here is the flow: 1. User accesses a resource that has been protected by authorization policies in OPSS, fox example. 2. If the user is not yet authenticated, ADF redirects the user to the URI configured in 'loginuri'. 3. OAM, should have a policy to protect the value in 'loginuri': for example, "/<context-root>/adfAuthentication. 4. When ADF redirects to this URI, OAM displays a Login Page (depending on the authentication scheme configured in OAM for this URI).

Table C–1 (Cont.) addOAMSSOProvider Command-line Arguments

Argument	Definition
logouturi	Specifies the URI of the logout page **Note**: For ADF security enabled applications, logouturi should be configured based on logout guidelines in Chapter 11. The ■ 11g WebGate the value of the logouturi should be sought from the 11g WebGate administrator. ■ 10g WebGate requires a logouturi value of "/oamsso/logout.html
autologinuri	Specifies the URI of the autologin page.

The procedure to configure domain-level jps-config.xml for a Fusion Web application with Oracle ADF Security enabled is part of a larger task. With the exception of the command syntax, all tasks are the same for Oracle Access Manager 10g and 11g.

See Also:

- *Oracle Fusion Middleware Application Security Guide* chapter "Configuring Single Sign-On in Oracle Fusion Middleware" for all tasks involving Oracle Access Manager 10g SSO providers
- *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* "Infrastructure Security Commands" chapter

All tasks involving Oracle Access Manager 10g SSO are described in the *Oracle Fusion Middleware Application Security Guide* chapter "Configuring Single Sign-On in Oracle Fusion Middleware."

- Sample SSO Configuration for OAM 11g
- SSO Provider Configuration Details

C.2.1 Sample SSO Configuration for OAM 11g

The SSO service configuration entered with the procedure described in Appendix C, "Integrating Oracle ADF Applications with Oracle Access Manager 11g SSO" is written to the file `jps-config.xml`. The data specified includes:

- A particular SSO service
- The auto-login and auto-logout URIs
- The authentication level
- The query parameters contained in the URLs returned by the selected SSO service
- The appropriate settings for token generation

The following fragment of a `jps-config.xml` file illustrates the configuration of an OAM 11g SSO provider. Some values are merely placeholders for actual content. Your configuration should contain values for your implementation.

See Also: "SSO Provider Configuration Details"

Example C–1 Sample SSO Configuration for OAM 11g

```
<propertySets>
  <propertySet name = "props.auth.url">
    <property name = "login.url.BASIC" value = "http://host:port/oam_
```

```
login.cgi?level=BASIC"/>
    <property name = "login.url.FORM" value = "http://host:port/oam_
login.cgi?level=FORM"/>
    <property name = "login.url.DIGEST" value = "http://host:port/oam_
login.cgi?level= DIGEST"/>
    <property name = "autologin.url" value = " http://host:port/obrar.cgi"/>
    <property name = "logout.url" value = "http://host:port/logout.cgi"/>
    <property name = "param.login.successurl"  value = "successurl"/>
    <property name = "param.login.cancelurl"   value = "cancelurl"/>
    <property name = "param.autologin.targeturl" value = "redirectto"/>
    <property name = "param.autologin.token"   value = "cookie"/>
    <property name = "param.logout.targeturl"   value = "targeturl"/>
  </propertySet>

  <propertySet name="props.auth.uri">
    <property name="login.url.BASIC"
value="/${app.context}/adfauthentication?level=BASIC" />
    <property name="login.url.FORM"
value="/${app.context}/adfauthentication?level=FORM" />
    <property name="login.url.DIGEST"
value="/${app.context}/adfauthentication?level=DIGEST" />
    <property name="autologin.url" value="/obrar.cgi" />
    <property name="logout.url" value="/${oamsso/logout.html" />
  </propertySet>

  <propertySet name = "props.auth.level">
    <property name = "level.anonymous" value = "0"/>
    <property name = "level.BASIC"    value = "1"/>
    <property name = "level.FORM"    value = "2"/>
    <property name = "level.DIGEST"   value = "3"/>
  </propertySet>
<propertySets>

<serviceProviders>
  <serviceProvider name = "sso.provider"
    class = "oracle.security.jps.internal.sso.SsoServiceProvider"
    type = "SSO">
    <description>SSO service provider</description>
  </serviceProvider>
</serviceProviders>

<serviceInstances>
  <serviceInstance name = "sso" provider = "sso.provider">
    <propertySetRef ref = "props.auth.url"/>
    <propertySetRef ref = "props.auth.level"/>
    <property name = "default.auth.level" value = "2"/>
    <property name = "token.type" value = "OAMSSOToken"/>
    <property name = "token.provider.class" value =
"oracle.security.wls.oam.providers.sso.OAMSSOServiceProviderImpl"/>
  </serviceInstance>
</serviceInstances>

<jpsContexts default = "default">
  <jpsContext name = "default">
    <serviceInstanceRef ref = "sso"/>
  </jpsContext>
</jpsContexts>
```

C.2.2 SSO Provider Configuration Details

Note the following important points:

- Any SSO provider must define the URI for at least the FORM login with the property `login.url.FORM`. The value need not be a URL.
- If the application supports a self-registration page URI or URL, it must be specified with the property `autologin.url`.
- If the SSO solution supports a global logout URI or URL, it must be specified with the property `logout.url`. The OAM solution supports global logout.
- The following properties, illustrated in Example C–1, are optional:
  - `param.login.successurl`
  - `param.login.cancelurl`
  - `param.autologin.targeturl`
  - `param.login.token`
  - `param.logout.targeturl`
- The use of the variable `app.context` in URI specifications, in values within the property set `props.auth.uri` for instance, is allowed for only ADF applications when integrating with the Oracle Access Manager solution.
- The property set `props.auth.level` is required.
- The reference to `props.auth.url` is required.
- The property `sso.provider.class` within a service instance of the SSO provider is the fully qualified name of the class implementing a specific SSO solution.

  In the case of the OAM solution, the provided class name is `oracle.security.wls.oam.providers.sso.OAMSSOServiceProviderImpl`.
- The property name `default.auth.level` within a service instance of the SSO provider must be set to "2", as illustrated in Example C–1.
- The property `token.type` within a service instance of the SSO provider is required.

  This token type identifies the token set on the HTTP request by the SSO provider upon a successful authentication; the SSO provider uses this token, after the first time, to ensure that the user does not need to be reauthenticated and that his sign-on is still valid. In the case of the OAM solution, the token type must be `OAMSSOToken`, as illustrated in Example C–1.
- The property `token.provider.class` within a service instance of the SSO provider is the fully qualified name of the token class, and it is provider-specific.
- An application that implements a self-registration logic and wants to auto login a user after successful self-registration, it must call the OPSS autoLogin API; in turn, to allow this call, it must grant that application a code source permission named `CredentialMapping` with class `JpsPermission`.

  The following fragment of the file `system-jazn-data.xml` illustrates the specification of this permission to the application `MyApp`:

  ```
  <grant>
    <grantee>
  ```

```
    <codesource>
      <url>file:${domain.home}/servers/MyApp/-</url>
    </codesource>
  </grantee>
  <permissions>
    <permission>
      <class>oracle.security.jps.JpsPermission</class>
      <name>CredentialMapping</name>
    </permission>
  </permissions>
</grant>
```

C.3 Confirming Application-Driven Authentication During Runtime

As mentioned earlier in this chapter, it is the application that triggers authentication and redirects the user to be authenticated by the appropriate solution. For instance, when the application determines that a user is accessing a part of a secured artifact that requires authentication application-driven authentication is triggered, in this case using Oracle Access Manager 11g SSO.

To confirm application-driven authentication during run time

1. Create the application based on the Oracle ADF framework.
2. Configure the Oracle Access Manager SSO Security provider, as described in "Integrating OAM 11g With Web Applications Using Oracle ADF Security and the OPSS SSO Framework" on page C-2.
3. Access the protected field and confirm that the application triggers authentication.

D

Internationalization and Multibyte Data Support for OAM 10g WebGates

The information here might be of interest if you are using OAM 10g WebGates:

- Introduction to Internationalization and Multibyte Data Support

D.1 Introduction to Internationalization and Multibyte Data Support

Oracle Access Manager 11g provides multi-lingual applications and software products that can be accessed and run anywhere simultaneously, without modification, while rendering content in the native user’s language and locale preferences.

A locale is the linguistic and cultural environment in which a system or program is running; data associated with a locale provides support for formatting and parsing of dates, times, numbers, currencies, and the like based on the linguistic and cultural requirements that corresponds to a given language and country.

Oracle product globalization is a two part process that includes internationalization and localization. *Internationalization* (sometimes shortened to "I18N", meaning "I - eighteen letters -N") requires that software products and applications must be usable on a computer running any supported operating system (in any supported language), with non-US keyboards or other country-specific hardware. Oracle applications do not have hard-coded dependencies on language strings, and inter-operate with non-US versions of other products. Oracle applications can handle multibyte characters and differences in a distributed environment, and also being able to detect the user’s desired locale. Oracle Access Manager meets these requirements and conforms to Unicode Standard 4.0.

Localization includes translation of separated file text. In Oracle products, including Oracle Access Manager, information is presented in a manner that is consistent with the user’s local cultural conventions, including data formatting, collation, currency, date, time, and directionality of text (right-to-left or left-to-right), as discussed next.

For more information, see:

- Languages For Localized Messages in Oracle Access Manager
- Bi-directional Language Support
- UTF-8 Encoding

D.1.1 Languages For Localized Messages in Oracle Access Manager

Translatable information can be categorized into two types: end-user information (accessible to all users) and administrative information (for users with administrator privileges). When you install Oracle Access Manager 10.1.4 without a Language Pack, English is the default language for Administrators and end users. When you install 10.1.4 with Oracle-provided Language Packs, you can choose the language to be used as the default for Administrative activities. Regardless of the default Administrator language you choose during installation, English is always installed.

> **Note:** Messages added for minor releases (10*g* (10.1.4.2.0) and 10*g* (10.1.4.3) as a result of new functionality might not be translated and can appear in only English.

For end-users, Oracle Access Manager 10.1.4 enables the display of static application data such as error messages, and display names for tabs, panels, and properties in the End Users languages identified in Table D–1. Administrative information can be displayed in only the Administrators languages listed in Table D–1. If administrative pages are requested in any other language (by the browser setting), the language that was selected as the default during product installation is used to display the pages.

Table D–1 Languages for Localized Messages in Oracle Access Manager

Language Tag for Installation Directory	End User Information	Administrators
en-us	English	English
ar-ar	Arabic	
pt-br	Brazilian Portuguese	Brazilian Portuguese
fr-ca	Canadian French	Canadian French
cs-cs	Czech	
da-dk	Danish	
nl-nl	Dutch	
fi-fi	Finnish	
fr-fr	French	French
de-de	German	German
el-gr	Greek	
he-il	Hebrew	
hu-hu	Hungarian	
it-it	Italian	Italian
ja-jp	Japanese	Japanese
ko-kr	Korean	Korean
es-mx	Latin American Spanish	Latin American Spanish
no-no	Norwegian	
pl-pl	Polish	
pt-pt	Portuguese	
ro-ro	Romanian	
ru-ru	Russian	
zh-cn	Simplified Chinese	Simplified Chinese
sk-sk	Slovak	

Table D–1 (Cont.) Languages for Localized Messages in Oracle Access Manager

Language Tag for Installation Directory	End User Information	Administrators
es-es	Spanish/Spain	Spanish
sv-sv	Swedish	
th-th	Thai	
zh-tw	Traditional Chinese	Traditional Chinese
tr-tr	Turkish	

D.1.2 Bi-directional Language Support

Most Western languages are written left to right (LTR), from the top of the page to the bottom. East Asian languages are usually written top to bottom, from the right side of the page to the left (RTL)—although exceptions are frequently made for technical books translated from Western languages.

Some languages, such as Hebrew and Arabic, are written and read predominantly from right to left. Numbers reverse direction in Arabic and Hebrew. While the text is written right to left, numbers within the sentence are written left to right with the most significant digit on the left, as in European and other LTR languages.

When LTR languages are mixed in with RTL languages, the complete document or content is considered *bi-directional*. Oracle Access Manager can support bi-directional languages. If the browser on the host computer is configured to use any bi-directional language, then Oracle Access Manager handles it properly.

Note: No administrative languages require bi-directional support.

To provide support for multiple languages and bi-directional languages, Oracle Access Manager 10.1.4 supports the Unicode standard for encoding.

Note: Writing direction does not affect the encoding of a character. Regardless of the writing direction, Oracle stores data in logical order—the order used by someone typing a language—rather than the order in which it is presented on the screen.

D.1.3 UTF-8 Encoding

UTF-8 encoding and support is provided automatically, whether you have a new 10.1.4 installation or upgrade an older installation to Oracle Access Manager 10.1.4. You do not need to make any changes to your environment. As with previous releases, data in the directory server is stored with UTF-8 encoding.

Note: All of your directory data is UTF-8 format. Oracle Access Manager does not support a mix of data types in the directory.

E

Securing Communication with OAM 11g

This appendix provides the information and steps required to ensure that OAM 11g Servers and OAM Agents are communicating securely across the NetPoint Access Protocol (NAP) channel (also referred to as the Oracle Access Manager Protocol channel). This chapter provides the following details:

- Prerequisites
- Introduction to Securing Communication Between OAM 11g Servers and WebGates
- Configuring Cert Mode Communication for OAM 11g
- Configuring Simple Mode Communication with OAM 11g

E.1 Prerequisites

Confirm that the OAM Server is running.

E.2 Introduction to Securing Communication Between OAM 11g Servers and WebGates

Securing communication between OAM Servers and WebGate Agents means defining the transport security mode for the NAP channel.

Secure communication on the NAP channel requires that each OAM Server and each WebGate use the same Security mode, either:

- Open: Un-encrypted communication

  In Open mode, there is no authentication or encryption between the WebGate and OAM Server. The WebGate does not ask for proof of the OAM Server’s identity and the OAM Server accepts connections from all WebGates. Use *Open* mode if communication security is not an issue in your deployment.

- Simple: Encrypted communication through the Secure Sockets Layer (SSL) protocol with a public key certificate issued by Oracle

  Use Simple mode if you have some security concerns, such as not wanting to transmit passwords as plain text, but you do not manage your own Certificate Authority (CA). In this case, OAM 11g Servers and WebGates use the same certificates, issued and signed by Oracle CA. For more information, see "About Simple Mode, Encryption, and Keys" on page E-10.

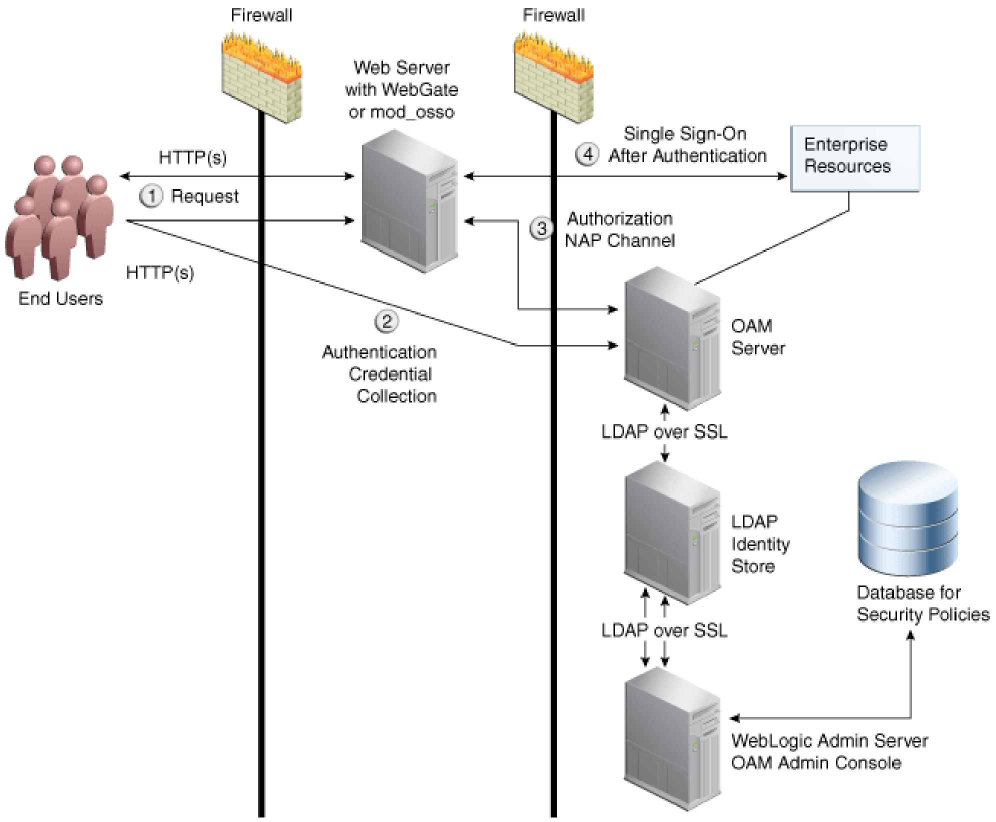
- Cert: Encrypted communication through SSL with a public key certificate issued by a trusted third-party certificate authority.

Use Cert mode if you want different certificates on OAM 11g Servers and WebGates and you have access to a trusted third-party CA. In this mode, you must encrypt the private key using the DES algorithm. Oracle Access Manager components use X.509 digital certificates in PEM format only. PEM refers to Privacy Enhanced Mail, which requires a passphrase. The PEM (Privacy Enhanced Mail) format is preferred for private keys, digital certificates, and trusted certificate authorities (CAs). The preferred keystore format is the JKS (Java KeyStore) format. For more information, see "About Cert Mode Encryption and Files" on page E-4.

> **See Also:** "About Certificates, Authorities, and Encryption Keys" on page E-3

Figure E–1 illustrates the communication channels used by OAM Servers and WebGates during user authentication and authorization.

Figure E–1 Communication Channels for OAM Servers and WebGates


Firewall
Firewall
Web Server with WebGate or mod_osso
HTTP(s)
1 Request
End Users
HTTP(s)
4 Single Sign-On After Authentication
Enterprise Resources
3 Authorization NAP Channel
2 Authentication Credential Collection
OAM Server
LDAP over SSL
LDAP Identity Store
Database for Security Policies
LDAP over SSL
WebLogic Admin Server OAM Admin Console

Process overview: Authentication and authorization

1. Request is intercepted by WebGate.
2. Authentication (credential collection) occurs over HTTP(s) channel.
3. Authorization occurs over the NAP channel with OAM Agents only (not mod_osso).

Using the secure-sockets layer (SSL) protocol helps prevent eavesdropping and successful man-in-the-middle attacks across the HTTP (HTTPS) channel. The SSL protocol is included as part of most Web server products and Web browsers. SSL uses the public-and-private key encryption system, which includes the use of a digital certificate. For details about enabling SSL communication for a Web server or directory server, see your vendor's documentation.

For more information, see:

- About Certificates, Authorities, and Encryption Keys
- About Security Modes and X509Scheme Authentication

E.2.1 About Certificates, Authorities, and Encryption Keys

Oracle Access Manager components use X.509 digital certificates in PEM format only. PEM refers to Privacy Enhanced Mail, which requires a pass phrase.

The PEM (Privacy Enhanced Mail) format is preferred for private keys, digital certificates, and trusted certificate authorities (CAs). The preferred keystore format is the JKS (Java KeyStore) format.

In cryptography, a public key is a value provided by a designated authority to be used as an encryption key. The system for using public keys is called a public key infrastructure (PKI). As part of a public key infrastructure, a certificate authority checks with a registration authority (RA) to verify information provided by the requestor of a digital certificate. When the RA verifies the requestor's information, the CA can issue a certificate.

Private keys can be derived from a public key. Combining public and private keys is known as asymmetric cryptography, which can be used to effectively encrypt messages and digital signatures.

Depending on the public key infrastructure, the digital certificate establishes credentials for Web-based transactions based on:

- Certificate owner's name
- Certificate serial number
- Certificate expiration date
- A copy of the certificate holder's public key, which is used to encrypt messages and digital signatures
- The digital signature of the certificate-issuing authority is provided so that a recipient can verify that the certificate is real

Digital certificates can be stored in a registry from which authenticating users can look up the public keys of other users.

See Also:

- "About Cert Mode Encryption and Files" on page E-4
- "About Simple Mode, Encryption, and Keys" on page E-10

E.2.2 About Security Modes and X509Scheme Authentication

OAM Server configuration defines the end points for the OAM Server and accounts for the deployment of load balancers or reverse proxies. When the HTTPS protocol is specified, the specified Server Port must not be configured to require CLIENT CERTS. This allows the user to interact with the server over SSL for all non-X509 authentication schemes and logout.

X509Module is called after Credential Collection if the corresponding authentication scheme is configured.

The X509 authentication scheme (X509Scheme) requires the X509 challenge method and X509 authentication module. The X509 module is called after credential collection when the X509Scheme is used.

When the X509Scheme is specified as the authentication scheme and the user must be challenged for credentials, the fully-qualified URL to the credential collector must be specified as the Challenge URL parameter of the authentication scheme. For example: https://<oam_server>:<ssl_port>/oam/CredCollectServlet/X509.

> **Note:** When the X509Scheme is specified, the specified SSL Port of the OAM Server must be different from the Server Port and must be configured to require Client Certificates.

The specified SSL Port must be different from the Server Port and must be configured to require CLIENT CERTS. If a relative Challenge URL is specified, the OAM Server uses the specified Server Port/Host/Port to construct the fully-qualified URL of the X509 Credential Collector. However, this configuration will not work.

When the OAM Server is reachable over both HTTP and HTTPS, all requests (come over either transport) are accepted. Administrators must ensure that the OAM Server is only reachable over the transport specified in the OAM Server configuration.

> **See Also:** "Managing the Common SSO Engine" on page 9-45

E.3 Configuring Cert Mode Communication for OAM 11g

This section describes how to configure Cert mode communication for OAM 11g.

The following tasks apply to Cert mode only. In Simple mode, the bundled OAM-CA-signed certificates are used and most of the following tasks here are not needed.

Task overview: Adding certificates for the OAM Server includes

1. Reviewing "About Cert Mode Encryption and Files"
2. Generating a Private Key, Certificate Request, Installing Certificates for OAM Server
3. Retrieving the OAM Keystore Alias and Password Using Custom WLST Commands
4. Importing CA-Signed Certificates Into the Keystore
5. Adding Certificate Details to OAM Common Server Properties
6. Generating a Private Key, Certificate Request, and Getting Certs for WebGates
7. Updating the WebGate to Use Certificates

E.3.1 About Cert Mode Encryption and Files

The certificate request for WebGate generates the request file aaa_req.pem. You must send this WebGate certificate request to a root CA that is trusted by the OAM Sever. The root CA returns the WebGate certificates, which can then be installed either during or after 10g WebGate installation (for 11g WebGate these must be copied to the WebGate instance area manually after WebGate installation and configuration).

- aaa_key.pem
- aaa_cert.pem
- aaa_chain.pem

During component installation in Cert mode, you are asked to present a certificate obtained from an external CA. If you do not yet have a certificate you can request one. Until you receive the certificate, you can configure the WebGate in Simple mode. You cannot complete OAM deployment until the certificates are issued and installed.

If you choose Cert mode when registering an OAM Agent, a field appears where you can enter the Agent Key Password. When editing an 11g WebGate registration, password.xml is updated only when the mode is changed from Open to Cert or Simple to Cert. In cert mode, once generated, password.xml cannot be updated. Editing the agent Key Password does not result in creation of a new password.xml.

You must create a Cert request and send that to the CA. When the certificate is returned you must import it to the OAM Server (or copy it to the WebGate).

E.3.2 Generating a Private Key, Certificate Request, Installing Certificates for OAM Server

Use the following procedure to retrieve the private key, certificate, and CA certificate for the OAM Server.

> **Note:** The certified tool to maintain consistency between 10g and 11g registration, is openSSL. Oracle recommends that you use openSSL rather than other tools to generate certificates and keys in PEM format.

To retrieve the private key and certificates for OAM 11g Server

1. Generate both the certificate request (aaa_req.pem) and Private Key (aaa_key.pem) as follows:

   ```
   openssl req -new -keyout aaa_key.pem -out aaa_req.pem -utf8 -nodes
   ```

2. Submit the certificate request (aaa_req.pem) to a trusted CA.
3. Download the CA Certificate in base64 as aaa_chain.pem.
4. Download the Certificate in both base64 and DER format as aaa_cert.pem and aaa_cert.der.
5. Encrypt the private key (aaa_key.pem) using a password as follows:

   ```
   openssl rsa -in aaa_key.pem -passin pass: -out aaa_key.pem -passout pass:
   ******** -des
   ```

6. Proceed to "Retrieving the OAM Keystore Alias and Password Using Custom WLST Commands".

E.3.3 Retrieving the OAM Keystore Alias and Password Using Custom WLST Commands

Users with valid OAM Administrator credentials can perform the following task to retrieve the keystore alias and password that is required to import a certificate.

To retrieve the OAM Keystore password

1. Confirm the OAM Administration Console is running.
2. On the computer hosting the OAM Administration Console, locate the WebLogic Scripting Tool in the OAM Installation path to use when retrieving the keystore password. For example:

   $ORACLE_IDM/common/bin/

   Here, $ORACLE_IDM is the OAM 11g base installation directory; /common/bin is the path in which the scripting tool is located.
3. Start the WebLogic Scripting Tool:

   ```
   ·/ wlst.sh
   ```
4. In the WLST shell, enter the command to connect and then enter the requested information. For example:

   ```
   wls:/offline> connect()
   Please enter your username [weblogic] :
   Please enter your password [welcome1] :
   Please enter your server URL [t3://localhost:7001] :
   wls:/base_domain/serverConfig>
   ```
5. Enter the following command to change the location to the read-only domainRuntime tree (For help, use help(domainRuntime)). For example:

   ```
   wls:/OAM_AC> domainRuntime()
   ```
6. Enter the following command to list the credentials for the OAM keystore. For example:

   ```
   wls:/OAM_AC/domainruntime> listCred(map="OAM_STORE",key="jks")
   ```

   Here, OAM_STORE represents the name of your OAM Keystore.
7. Pay close attention to the password of the OAM Keystore that is displayed because this is required to import the certificates.
8. Proceed to "Importing CA-Signed Certificates Into the Keystore".

E.3.4 Importing CA-Signed Certificates Into the Keystore

The keystore associated with Oracle Access Manager 11g accepts only PEM format certificates.

If you already have certificates signed by your certificate authority (CA) in PEM format, the following procedure describes how to import the certificate using the `keytool importcert` tool shipped with OAM 11g. The Readme file that is bundled with the tool provides instructions for importing the certificates in the keystore.

Note: If PEM format certificates are not available, create the certificate request and get it signed by your CA.

Following are the steps for using the JDK version 6 keytool. If you have a different version of keytool, refer the documentation for your JDK version.

> **Note:** When you use the keytool utility, the default key pair generation algorithm is Digital Signature Algorithm (DSA). However, OAM and WebLogic Server do not support DSA and you must specify another key pair generation and signature algorithm.

Prerequisites

Retrieving the OAM Keystore Alias and Password Using Custom WLST Commands

To import certificates into the keystore

1. Locate the importcert tool for OAM 11g in the following path:

   $ORACLE_IDM/oam/server/tools/importcert

2. Unzip importcert.zip and locate the Readme file.

3. Import the trusted certificate chain using the following command and details for your environment:

   ```
   keytool -importcert -file aaa_chain.pem - trustcacerts -storepass <password>
   -keystore <MW_HOME>/user_projects/domains/domain_name/config/fmwconfig/
   .oamkeystore -storetype JCEKS
   ```

4. Convert the private key (aaa_key.pem) and signed certificate (aaa_cert.pem) to DER format using openSSL or any other tool. For example:

   ```
   openssl pkcs8 -topk8 -nocrypt -in aaa_key.pem -inform PEM -out aaa_key.der
   -outform DER
   ```

   Perform the following if you do not have aaa_cert.der.

   a. Enter the following command:

      ```
      openssl x509 -in aaa_cert.pem -inform PEM -out aaa_cert.der -outform DER
      -outform DER
      ```

   b. Edit aaa_chain.pem using TextPad to remove all data except that which is contained within the CERTIFICATE blocks, and save the file in a new location to retain the original.

      ```
      -----BEGIN CERTIFICATE-----
      ...
      CERTIFICATE
      ...
      -----END CERTIFICATE-----
      ```

5. Import signed PEM format certificates into the keystore. For example:

   a. Locate the importcert tool for OAM 11g in the following path:

      $ORACLE_IDM/oam/server/tools/importcert

   b. Unzip importcert.zip and locate the Readme file.

   c. Import signed PEM format certificates using the following command line arguments and details for your environment:

      ```
      - java -cp importcert.jar:$CLASSPATH
      oracle.security.am.common.tools.importcerts.CertificateImport -keystore <>
      ```

```
-keystorepassword <> -privatekeyfile <> -signedcertificate <> -alias
[-aliaspassword <>]
```

6. Proceed with "Adding Certificate Details to OAM Common Server Properties"

E.3.5 Adding Certificate Details to OAM Common Server Properties

After importing the certificates into the keystore, you must add the alias and password that you retrieved earlier into the OAM Proxy section of each OAM Server configuration in the Oracle Access Manager 11g Administration Console.

The Secure Sockets Layer (SSL) protocol is commonly used to manage secure communication on the Internet. Using the SSL protocol to protect communication between OAM Servers and WebGates helps prevent eavesdropping and successful man-in-the-middle attacks. The SSL protocol is included as part of most Web server products and Web browsers (Microsoft and Netscape, for instance). SSL uses the public-and-private key encryption system, which includes the use of a digital certificate.

> **Note:** No explicit configuration is needed for Simple mode, which is provided out of the box for OAM 11g.

Prerequisites

Importing CA-Signed Certificates Into the Keystore

To add certificate details to OAM Server configurations

1. From the Oracle Access Manager 11g Administration Console, click the System Configuration tab.
2. From the System Configuration tab, navigation tree, double-click Server Instances to view the OAM Server Common Properties page.
3. Click the OAM Proxy tab.
4. Fill in the alias and alias password details acquired in Step 5c of "Importing CA-Signed Certificates Into the Keystore", like one of the following examples:

   Simple Mode Configuration

   Global Passphrase: *simple_passphrase*

   Cert Mode Configuration

   PEM KeyStore Alias: *my_keystore_alias*

   PEM KeyStore Alias Password: *my_keystore_alias_pw*
5. Click Apply to save the configuration.
6. Close the page.
7. Open the OAM Server registration page, click the Proxy tab, change the Proxy mode to Cert, and click Apply.
8. Restart the OAM Server.
9. Proceed to "Generating a Private Key, Certificate Request, and Getting Certs for WebGates".

E.3.6 Generating a Private Key, Certificate Request, and Getting Certs for WebGates

Use the following procedure to retrieve the private key, certificate, and CA certificate for the WebGate.

> **Note:** The certified tool to maintain consistency between 10g and 11g registration, is openSSL. Oracle recommends that you use openSSL rather than other tools to generate certificates and keys in PEM format.

To retrieve the private key and certificates for WebGates

1. Generate both the certificate request (aaa_req.pem) and Private Key (aaa_key.pem) as follows:

   ```
   openssl req -new -keyout aaa_key.pem -out aaa_req.pem -utf8 -nodes
   ```

2. Submit the certificate request (aaa_req.pem) to a trusted CA.
3. Download the CA Certificate in base64 as aaa_chain.pem.
4. Download the Certificate in base64 format as aaa_cert.pem.
5. Encrypt the private key (aaa_key.pem) using a password as follows:

   ```
   openssl rsa -in aaa_key.pem -passin pass: -out aaa_key.pem -passout pass:
   ******** -des
   ```

6. Proceed to "Updating the WebGate to Use Certificates".

E.3.7 Updating the WebGate to Use Certificates

For all communication modes (Open, Simple, or Cert), the Agent registration should be updated from the OAM Administration Console.

If you choose Cert mode when registering an OAM Agent, a field appears where you can enter the Agent Key Password. When editing an 11g WebGate registration, password.xml is updated only when the mode is changed from Open to Cert or Simple to Cert. In cert mode, once generated, password.xml cannot be updated. Editing the agent Key Password does not result in creation of a new password.xml.

Prerequisites

Adding Certificate Details to OAM Common Server Properties

To update the communication mode in the WebGate Agent registration

1. From the System Configuration tab, navigation tree, expand the Agents node.
2. Expand OAM Agents, expand the 11g Webgates (or 10g Webgates) node, and then double-click the desired agent's name.
3. On the agent's registration page, locate the Security options and click Cert (or Simple).
4. Cert Mode: Enter the Agent key Password as specified in Step 5 of "Generating a Private Key, Certificate Request, and Getting Certs for WebGates".
5. Click Apply to submit the changes.
6. Copy the following updated WebGate files as follows:

   ObAccessClient.xml

cwallet.sso
password.xml

- From: IDM_DOMAIN_HOME/output/AGENT_NAME
- To: OHS_INSTANCE_HOME/config/OHS/ohs2webgate/config

7. Copy the following files (created during "Generating a Private Key, Certificate Request, Installing Certificates for OAM Server") as follows:

   aaa_key.pem
   aaa_cert.pem
   aaa_chain.pem

   - From: IDM_DOMAIN_HOME/output/AGENT_NAME
   - To: OHS_INSTANCE_HOME/config/OHS/ohs2webgate/config

8. Restart the OAM Server and the Oracle HTTP Server instance.

E.4 Configuring Simple Mode Communication with OAM 11g

The transport security communication mode is chosen during OAM installation. When Simple mode is chosen, the installer generates a random global passphrase initially, which can be edited as required later.

When you register an OAM Agent or a new OAM Server, you can specify the mode. However, changing the global passphrase requires that you reconfigure all agents to use Simple mode and the new global passphrase.

During agent registration, at least one OAM Server instance must be running in the same mode as the agent. Otherwise, registration fails. After agent registration, however, you could change the communication mode of the OAM Server. Communication between the agent and server continues to work as long as the WebGate mode is at least at the same level as the OAM Server mode or higher. The agent mode can be higher but cannot be lower.

This section provides the information you need to configure Simple mode communication with OAM 11g.

Task overview: Configuring Simple mode communication with OAM 11g includes

1. Reviewing "About Simple Mode, Encryption, and Keys"
2. Updating the WebGate Registration for Simple Mode
3. Verifying Simple Mode Configuration

E.4.1 About Simple Mode, Encryption, and Keys

For Simple mode encryption, Oracle Access Manager ships a certificate authority with its own private key, which is installed across all WebGates and OAM Servers. For each public key there is a corresponding private key that Oracle Access Manager stores in the aaa_key.pem file.

A program named openSSL in the \tools subdirectory automatically generates the key pair and the following files for Simple mode security:

- cacert.pem the certificate request, signed by the Oracle-provided openSSL Certificate Authority

- password.xml contains the random global passphrase that was designated during agent registration, in obfuscated format. This needs to be copied to the WebGate instance location.
- aaa_key.pem contains your private key (generated by openSSL).
- aaa_cert.pem signed certificates in PEM format

The transport security communication mode is chosen during OAM installation. The installer generates a random global passphrase initially, which can be edited as required later.

When you install an OAM Agent, you can request CA certificates. When you register an OAM Agent or a new OAM Server, you must specify the communication mode. However, changing the global passphrase requires reconfiguring all agents to use Simple mode and the new global passphrase.

E.4.2 Updating the WebGate Registration for Simple Mode

Artifacts generated for Simple Security mode use the Global Pass phrase and a change must be propagated to WebGates. You can delete the WebGate registration and re-register it (specifying Simple mode and disabling the automatic generation of policies) or you can edit the WebGate registration and then copy the artifacts as described here.

To update the WebGate registration for Simple mode

1. From the System Configuration tab, navigation tree, expand the Agents node.
2. Expand OAM Agents, expand the 11g Webgates (or 10g Webgates) node, and then double-click the desired agent's name.
3. On the agent's registration page, locate the Security options and click Simple.
4. Click Apply to submit the changes.
5. Copy the following updated WebGate files as follows:

   ObAccessClient.xml
   cwallet.sso (11g WebGate only)
   password.xml

   - From: $WLS_DOMAIN_HOME/output/*AGENT_NAME* (the WebLogic domain home where the OAM AdminServer is installed)
   - To: OHS_INSTANCE_HOME/config/OHS/ohs2webgate/config
6. Copy the following files, as directed here:

   aaa_key.pem
   aaa_cert.pem

   - From: IDM_DOMAIN_HOME/output/*AGENT_NAME*
   - To: OHS_INSTANCE_HOME/config/OHS/ohs2webgate/config/simple
7. Restart the OAM Server and the Oracle HTTP Server instance.

E.4.3 Verifying Simple Mode Configuration

You must restart the Web server to instantiate the change to Simple mode. Then you can validate the results

To validate Simple mode changes

1. From a command-line window, restart the Web server. For example:

   ```
   d:\middleware\ohs_home\instances\ohs_webgate11g\bin
   opmnctl stopall
   opmnctl startall
   ```

2. In a browser window, enter the URL to a resource protected by the WebGate using Simple mode.
3. Enter your login credentials, when asked.
4. Confirm that the resource is served.

F

Introduction to Custom WLST Commands for OAM Administrators

For certain OAM administrative tasks, the WebLogic Scripting Tool (WLST) provides custom commands that can be used as an alternative to the OAM Administration Console. This appendix provides an introduction to WLST commands for OAM administrators. Details for each command, however, are outside the scope of this book.

Sections in this appendix include:

- Prerequisites
- WLST OAM Command Summary
- Running WLST Commands for OAM Operations

F.1 Prerequisites

Become familiar with information in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

F.2 Introduction to WebLogic Scripting Tool Commands for OAM

Custom WLST commands for OAM can be used for setting and managing OAM System Configuration only by OAM Administrators.

The WebLogic Scripting Tool shares the same foundation layer with the OAM Administration Console. WLST for OAM is available within ORACLE_IDM.

> **Note:** To use the Infrastructure Security custom WLST commands, you must invoke the WLST script from the Oracle Common home. See "Using Custom WLST Commands" in the *Oracle Fusion Middleware Administrator's Guide.*

OAM WLST commands are defined in the oamWlstCmd.py file in the following path:

```
<ORACLE_IDM>/common/wlst
```

The oamWlstCmd.py file refers to jar files available in:

```
<Oracle_IDM>/oam/server/lib/jmx
<Oracle_IDM>/oam/server/lib/wlst
```

Most WLST commands for OAM operate in both online and offline modes. Operational modes are described in Table F–1.

Table F–1 Operational Modes for WLST commands for OAM

Online Mode	Offline Mode
Connects to the Mbean Server running on the WebLogic AdminServer	Method invocation happens locally in the WLST Shell
The Mbean Server can be running remotely	Requires the OAM Domain Home as a mandatory input
Invokes OAM WLST Mbean methods, which are executed in the server	N/A
OAM WLST Mbeans return the result of the execution to the WLST commands.	N/A

F.3 WLST OAM Command Summary

Use the WLST commands listed in Table F–2 to manage Oracle Access Manager (OAM)-related components, such as authorization providers, identity asserters, and SSO providers, as well as to display metrics and deployment topology, manage Oracle Access Manager server and agent configuration and more.

> **See Also:** The section on Oracle Access Manager commands in the chapter "Infrastructure Security Custom WLST Commands" of the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

Table F–2 WLST Oracle Access Manager Commands

Use this command...	To...	Use with WLST...
listOAMAuthnProviderParams	List the parameters set for an Oracle Access Manager authentication or identity assertion provider.	Online
createOAMIdentityAsserter	Create a new identity asserter.	Online
updateOAMIdentityAsserter	Update an existing identity asserter.	Online
createOAMAuthenticator	Create a new authenticator.	Online
deleteOAMAuthnProvider	Delete an existing authentication provider.	Online
updateOAMAuthenticator	Update an existing authenticator.	Online
addOAMSSOProvider	Add a new SSO provider.	Online
displayTopology	List the details of deployed Oracle Access Manager Servers.	Online Offline
displayOamServer	Display Oracle Access Manager Server configuration details.	Online Offline
createOamServer	Create an entry for an Oracle Access Manager Server configuration.	Online Offline
editOamServer	Edit the entry for an Oracle Access Manager Server configuration.	Online Offline
deleteOamServer	Delete the named Oracle Access Manager Server configuration.	Online Offline
displayOssoAgent	Display OSSO Agent configuration details.	Online Offline

Table F–2 (Cont.) WLST Oracle Access Manager Commands

Use this command...	To...	Use with WLST...
editOssoAgent	Edit OSSO Agent configuration details.	Online Offline
deleteOssoAgent	Delete the named OSSO Agent configuration.	Online Offline
displayWebgateAgent	Display 10g WebGate Agent configuration details.	Online Offline
editWebgateAgent	Edit 10g WebGate Agent registration details.	Online Offline
deleteWebgateAgent	Delete the named 10g WebGate Agent configuration.	Online Offline
changeLoggerSetting	Change Logger Settings.	Online Offline
changeConfigDataEncryptionKey	Regenerate the configuration data encryption key and re-encrypt data.	Online Offline
displayUserIdentityStoreConfig	Display a user identity store registration.	Online Offline
editUserIdentityStoreConfig	Edit a user identity store registration.	Online Offline
createUserIdentityStoreConfig	Create a user identity store registration.	Online Offline
deleteUserIdentityStore	Delete a user identity store registration.	Online Offline
configRequestCacheType	Configure the SSO server request cache type.	Online Offline
displayRequestCacheType	Display the SSO server request cache type entry.	Online
exportPolicy	Export Oracle Access Manager policy data from a test (source) to an intermediate Oracle Access Manager file.	Online
importPolicy	Import Oracle Access Manager policy data from the Oracle Access Manager file specified.	Online
importPolicyDelta	Import Oracle Access Manager policy changes from the Oracle Access Manager file specified.	Online
migratePartnersToProd	Migrate partners from the source Oracle Access Manager Server to the specified target Oracle Access Manager Server.	Online
exportPartners	Export the Oracle Access Manager partners from the source to the intermediate Oracle Access Manager file specified.	Online
importPartners	Import the Oracle Access Manager partners from the intermediate Oracle Access Manager file specified.	Online

Table F–2 (Cont.) WLST Oracle Access Manager Commands

Use this command...	To...	Use with WLST...
configureOAAM	Configure the Oracle Access Manager-Oracle Adaptive Access Manager basic integration.	Online
registerOIFDAPPartner	Register Oracle Identity Federation as Delegated Authentication Protocol (DAP) Partner.	Online Offline
enableCoexistMode	Enable the Coexist Mode.	Online
disableCoexistMode	Disable the Coexist Mode.	Online
editGITOValues	Edit GITO configuration parameters.	Online Offline
editWebgate11gAgent	Edit an 11g WebGate registration.	Online
deleteWebgate11gAgent	Remove an 11g WebGate Agent registration.	Online Offline
displayWebgate11gAgent	Display an 11g WebGate Agent registration.	Online Offline
displayOAMMetrics	Display metrics of OAM Servers.	Online Offline
updateOIMHostPort	Update the Oracle Identity Manager configuration when integrated with Oracle Access Manager.	Online Offline
configureOIM	Creates an Agent registration specific to Oracle Identity Manager when integrated with Oracle Access Manager.	Online
updateOSSOResponseCookieConfig	Updates OSSO Proxy response cookie settings.	Online Offline
deleteOSSOResponseCookieConfig	Deletes OSSO Proxy response cookie settings.	Online Offline

F.4 Running WLST Commands for OAM Operations

OAM Administrators can use the following procedure as a guide for using WLST commands for OAM-specific operations. Included here are several operations:

> **See Also:** The section on Oracle Access Manager commands in the chapter "Infrastructure Security Custom WLST Commands" of the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

- Starting the WLST Shell and Logging In
- Changing the Request Cache Type in a High Availability Environment

F.4.1 Starting the WLST Shell and Logging In

Use the following procedure for general information when you are starting the WLST shell.

To run WLST commands for OAM operations

1. Ensure that the OAM AdminServer is running.
2. Set up the environment for WLST by running the following command:

   ```
   DOMAIN_HOME/bin/setDomainEnv.sh
   ```

3. Go to the OAM_HOME path: <Oracle_IDM>/common/bin.
4. Execute the appropriate command to enter the WLST shell.

   ```
   Linux: wlst.sh
   Windows: wlst.cmd
   ```

5. Execute help commands, as needed: help('oam') to list available OAM WLST commands.

   ```
   OAM WLST: help('oam')
   Specific Command: wlst.cmd
   ```

6. Connect to your domain. For example:

   ```
   wls:/base_domain/serverConfig> connect()
   ```

7. Enter the WebLogic Administration username and password, and enter the URL for the Administration Server in the following format:

   ```
   Please enter your username
   Please enter your password
   Please enter your server URL : t3://OAMHOST1.mycompany.com:7001
   wls:/base_domain/serverConfig>
   ```

8. Offline Mode: Provide ‘domainHome’ as an input to the command.
9. Online Mode: Connect to the Mbean server using the command ‘connect ()’
10. Check the section on Oracle Access Manager commands in the chapter "Infrastructure Security Custom WLST Commands" of the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* for full details.

F.4.2 Changing the Request Cache Type in a High Availability Environment

In high availability configurations, the Request Cache type must be changed from BASIC to COOKIE using Infrastructure Security custom WLST commands.

See Also:

- "Managing Run Time Policy Evaluation Caches" on page 4-16
- OAM_REQ cookie in Table 7–4, " SSO Cookies"

To change the Request Cache Type in a high-availability environment

1. Log in to the WLST shell and connect to your domain as described in "Starting the WLST Shell and Logging In" on page F-4.
2. Run the following command to configure the request cache type for a high-availability deployment as COOKIE:

   ```
   wls:/base_domain/serverConfig> configRequestCacheType(type="COOKIE")
   ```

3. Validate that the command worked using the following command:

   ```
   wls:/base_domain/serverConfig> displayRequestCacheType
   ```

4. Restart the OAM Servers.

G

Configuring OAM 11g for IPv6 Clients

Internal communication among Oracle Access Manager 11g and its dependencies uses Internet Protocol Version 4 (IPv4). However, external communication is supported in IPv6 with Oracle HTTP Server with the mod_wl_ohs plug-in.

This appendix provides the following topics:

G.1 Prerequisites

Regardless of the manner in which you plan to use Oracle Access Manager with IPv6 clients, the following tasks should be completed before you start activities herein:

- An Oracle HTTP Server instance must be installed to act as a reverse proxy to the Web server (required for 11g WebGates).
- Oracle Access Manager must be installed as described in Oracle Fusion Middleware Installation Guide for Oracle Identity Management

> **See Also:**
>
> - "Using IPV6" in the chapter on changing network configurations in the *Oracle Fusion Middleware Administrator's Guide* for details about configuring OAM 10g WebGates for IPv6 clients.
> - *Oracle HTTP Server Administrator's Guide*

G.2 Introduction to Oracle Access Manager 11g and IPv6

Among other features, IPv6 supports a larger address space (128 bits) than IPv4 (32 bits), providing an exponential increase in the number of computers that can be addressable on the Web. IPv6 is enabled with Oracle HTTP Server with the mod_wl_ohs plug-in.

The OAM Server and WebGate (10g and 11g) are IPv4 only. However, an IPv6 client can access WebGate on IPv4 through reverse proxy on an IPv4/IPv6 dual-stack host.

> **Note:** You can configure Oracle Access Manager 11g to work with clients that support IPv6 by setting up a reverse proxy server.

The supported topologies for OAM 11g with IPV4/IPV6 are outlined in following lists.

Topology

- WebGate10g or WebGate 11g +protected applications on IPv4 protocol host
- OHS reverse proxy on dual-stack host
- Client on IPv6 protocol host
- OAM Server Proxy

IPv6 client can access WebGate10g or WebGate11g through OHS reverse proxy.

> **Note:** When the OAM Server is not running, login to the WebLogic Administration Console is successful,. However, when OAM Server is running, login to the WebLogic Administration Console is redirected to the OAM Server and authentication fails because the Identity Store fails to initialize. IPV6 for the Identity Store is not yet supported.

For more information, see:

- Configuring IPv6 with OAM 11g and Challenge Redirect
- Considerations

For a look at all supported topologies, including configuration for OAM 10g WebGates, see "Using IPV6" in the chapter on changing network configurations in the *Oracle Fusion Middleware Administrator's Guide*

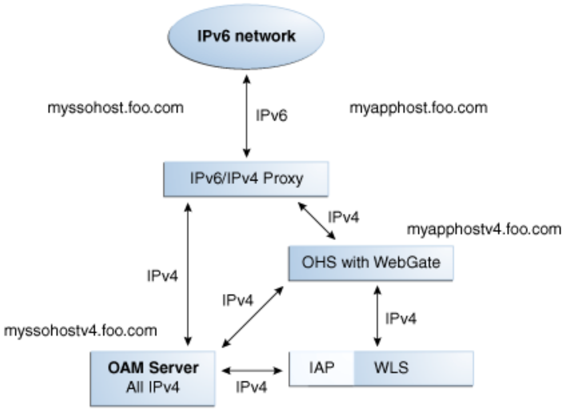
G.2.1 Configuring IPv6 with OAM 11g and Challenge Redirect

Figure G–1 illustrates configuration with a single IPv6 to IPv4 Proxy (even though *myssohost* and *myapphost* can use separate proxies).

With OAM 11g, the virtual host name must be specified as a host name, for example, *myapphost.foo.com*, not as an IP address. The redirect host name, for example, *myssohost.foo.com* must also be specified as a host name and not an IP address. The IPv6 address cannot be specified in a WebGate registration.

> **Note:** With OAM 11g, there is no concept of an authenticating WebGate or a resource WebGate. Instead, redirection always goes to OAM Server whether you have 11g WebGates or 10g WebGates.

Figure G–1 IPv6 with OAM 11g and Challenge Redirect



As illustrated in Figure G–1, the IPv6 network communicates with the IPv6/IPv4 proxy, which in turn communicates with the Oracle HTTP Server using IPv4. WebGate, Oracle Access Manager Server, and Oracle WebLogic Server with the Identity Asserter all communicate with each other using IPV4.

You should be able to access the application from a browser on the IPv6 network to the IPv6 server host (*myapphost.foo*.com) and have login with redirect to IPv6 *myssohost.foo*.com.

G.2.2 Considerations

The following considerations apply to each intended use scenario:

- IP validation does not work by default. To enable IP validation, you must add the IP address of the Proxy server as the WebGate's IPValidationException parameter value in the OAM Administration Console.

  See Also: "Single Sign-On with OAM 11g" on page 7-16

- IP address-based authorization does not work because all requests come through one IP (proxy IP) that would not serve its purpose.
- ipValidationException is required if IPValidation is On (parameter "ipValidation"=1). However, you cannot add this parameter using either the Administration Console or the remote registration tool. Instead, you must add the proxy's IP as single-valued user-defined parameter for the proxy in the oam-config.xml file, as described in later procedures in this chapter.

G.3 Configuring IPv6: Separate Proxy for OAM 11g and WebGates

OAM 10g provided a resource WebGate configuration (that redirects) and an Authenticating WebGate configuration. The OAM 11g credential collector replaces and performs the function of an OAM 10g authenticating WebGate.

Note: With OAM 11g, the 10g WebGate always redirects to the OAM 11g credential collector which acts like the earlier "authenticating" WebGate.

In this configuration you have multiple proxies: for example a separate proxy for the OAM Server and another proxy for the WebGate.

You can access the application from a browser on the IPv4 network directly to an IPv4 server host name with a login redirect to an IPv6 host. For example:

WebGate is on http://*myapphostv4.foo*.com/
OAM Server is on http://*myssohostv4.foo*.com

Proxy used for *myapphostv4.foo*.com should be *myapphost.foo*.com
Proxy used for *myssohostv4.foo*.com should be *myssohost*.com

> **Note:** You cannot use the IPv6 proxy name as the Preferred HTTP host in a WebGate registration.

With OAM 11g, the ProxyRequests parameter must be "On" because WebGates (11g or 10g) always redirect to obrareq.cgi. This directive makes the proxy act as a forward proxy.

The Preferred http host should be set to the host:port of the Web server hosting the WebGate (or SERVER_NAME if the Web server hosting the WebGate is configured for virtual hosting).

If IPValidation is ON, IPValidationException must be added for the proxy.

If reverse proxy is configured to perform SSL termination, then the user-defined WebGate `proxySSLHeaderVar` parameter must be defined during remote registration. As described in Table 6–4, " Elements Common to Remote Registration Requests", this parameter is used when the WebGate is located behind a reverse proxy. The value of the `proxySSLHeaderVar` parameter defines the name of the header variable the proxy must set. The value of the header variable must be "ssl" or "nonssl". If the header variable is not set, the SSL state is decided by the SSL state of the current Web server. Syntax is as follows:

```
<name>proxySSLHeaderVar</name>
<value>IS_SSL</value>
```

Modify the Load Balancing Router (reverse proxy Web server) settings to insert an HTTP header string that sets the IS_SSL value to ssl. For example, in the F5 load balancer, in Advanced Proxy Settings, you add the HTTP header string IS_SSL:ssl

In the following procedure, *OHS_host* and *OHS_port* are the host name and port of the actual Oracle HTTP Server that is configured for WebGate. Be sure to use values for your own environment. Your values will be different.

Prerequisites

Install and configure OHS Web server for reverse proxy. Ensure that you have a separate Web server instance for each proxy.

To configure IPv6 with a separate proxy for OAM 11g and WebGates

1. Enable mod_proxy to OAM 11g Server and WebGate: Configure Oracle HTTP Server 11g Release 1 (11.1.1) or any other server for multiple proxies, as follows:

   a. Stop Oracle HTTP Server for the corresponding proxy instance with the following command:

   ```
   opmnctl stopproc ias-component=<OHS instance name>
   ```

b. Edit the following file for the OHS instance for the OAM Server corresponding proxy:

```
UNIX: ORACLE_INSTANCE/config/OHS/ohs_name1/httpd.conf
Windows: ORACLE_INSTANCE\config\OHS\ohs_name1\httpd.conf
```

c. **Proxy to OAM 11g Server**: Append the following information for your environment to the httpd.conf file to enable mod_proxy. For example:

```
<IfModule mod_proxy.c>
ProxyRequests On
ProxyPreserveHost On

ProxyPass / http://<oam_server_host:port>/
ProxyPassReverse / http://<oam_server_host:port>/
</IfModule>
```

d. **Reverse Proxy to 11g WebGate**: Append information for your environment to the httpd.conf file to enable mod_proxy, as follows:

```
<IfModule mod_proxy.c>
ProxyRequests On
ProxyPreserveHost On

ProxyPass / http://<webgate_OHS_host:port>/
ProxyPassReverse / http://<webgate_OHS_host:port>/
</IfModule>
```

e. Restart Oracle HTTP Server with the following command:

```
opmnctl startproc ias-component=<OHS instance name>
```

2. In the Authentication Scheme, change the Challenge Redirect URL to http://<*oam_server_proxy_host*:port>/oam/server.

3. Set the Preferred HTTP host for each WebGate to the host:port of the Web server hosting the WebGate (or SERVER_NAME if the Web server hosting WebGate is configured for virtual hosting):

> **Note:** You can specify Preferred HTTP host using the appropriate field of the *Request.xml input during remote registration or using the Administration Console as shown here. See also, "About Remote Registration Requests" on page 6-8.

a. Log in to OAM Administration Console. For example:

```
    http://hostname:port/oamconsole
```

b. Click **System Configuration**, and then expand **Agents, OAM Agents, 11g (or 10g) Agents**.

c. Double-click an agent name to display the registration page.

d. **Preferred HTTP Host**: The name of the Oracle HTTP Server Web server that is configured for this WebGate. For instance, a WebGate deployed on *myapphostv4.foo*.com must use *myapphostv4.foo*.com as the Preferred HTTP host.

See Also:

- "About Remote Registration Requests"
- "About Virtual Web Hosting" on page 8-7

e. Click Apply.

f. Repeat for each WebGate and specify name of the Oracle HTTP Server Web server that is configured for this WebGate.

4. **IPValidationException**: If IPValidation is On (parameter "ipValidation"=1), add the proxy's IP as single-valued user-defined parameter for the proxy in the oam-config.xml file.

   a. Stop all OAM Servers and the AdminServer.

   b. Locate the oam-config.xml in the following path:

      <WLS_DOMAIN_HOME>/config/fmwconfig/oamconfig.xml

   c. Enter the following information:

```
<Setting Name="ipValidationExceptions"Type="xsd:string"> 10.1.1.1</Setting
>
```

   d. Save the file.

   e. Restart the OAM Servers and AdminServer.

   f. If reverse proxy is configured to perform SSL termination, the WebGate user-defined "`proxySSLHeaderVar`" parameter must be set (default is "IS_SSL"). Please modify the Load Balancing Router (reverse proxy Web server) settings to insert an HTTP header string that sets the IS_SSL value to ssl. For example, in the F5 load balancer, in Advanced Proxy Settings, you add the HTTP header string IS_SSL:ssl.

H

Troubleshooting

This chapter provides troubleshooting tips.

H.1 Introduction to OAM 11g Troubleshooting

OAM is a business critical system; downtime comes with a potentially high cost to your business. The goal of system analysis is to quickly isolate and correct the cause of any problem. This requires a big picture view of your system and the tools to observe the live system and correlate components to the bigger picture.

To assist administrators in performing a quick diagnosis, this section provides the following topics:

- About System Analysis and Problem Scenarios
- About LDAP Server or Identity Store Issues
- About OAM Server or Host Issues
- About Agent-Side Configuration and Load Issues
- About Runtime Database (Audit or Session Data) Issues
- About Change Propagation or Activation Issues
- About Policy Store Database Issues

H.1.1 About System Analysis and Problem Scenarios

System analysis includes understanding how the product works, what can go wrong, how likely the scenarios are, and the consequences or observable issues.

System problems can be divided into two basic categories:

- Cascading catastrophic failure
- Gradual breakdown in performance

Cascading catastrophic failure might be caused by:

- LDAP server is loaded and unresponsive
- Morning peak load starts
- WebGates send requests to the primary OAM Server
- WebGate requests time-out and WebGates retry to secondary OAM Server

Gradual breakdown in performance might occur over time when, for example:

- OAM is sized and rolled out for 10,000 users and 500 groups
- Over the course of a year, the number of users and groups increases significantly (to 50,000 users and 250 groups for example)

For information on the most commonly encountered issues, see the following topics:

- About System Analysis and Problem Scenarios
- About LDAP Server or Identity Store Issues
- About OAM Server or Host Issues
- About Agent-Side Configuration and Load Issues
- About Runtime Database (Audit or Session Data) Issues
- About Change Propagation or Activation Issues
- About Policy Store Database Issues

H.1.2 About LDAP Server or Identity Store Issues

This topic provides symptoms, probable cause, and steps to diagnose the following issues:

- Symptoms: Operational Slowness
- Symptoms: Total loss of service

Symptoms: Operational Slowness

- Poor user experience
- Agent timeouts lead to retries

Cause

- Non-OAM load might be impacting OAM operations
- Capacity problems due to gradual increase in peak load

Symptoms: Total loss of service

Cause

- Outage of all LDAP servers
- The load balancer is timing out old connections

Diagnosis

1. Shut down the LDAP server.
2. Restart your browser.
3. Try to access a protected site.
4. Review errors in the OAM Server log file, as described in Chapter 13 (alternatively, in Chapter 16).
5. Try to access OAM Administration Console.
6. Observe errors in WebLogic AdminServer log file.
7. Bring up the LDAP server again.
8. Retry access to a protected application.
9. Retry access to the OAM Administration Console.
10. Correct the issue based on the requirements in your environment.

H.1.3 About OAM Server or Host Issues

This topic provides symptoms, probable cause, and steps to diagnose the following issues:

- Symptoms: Capacity Problems
- Symptoms: Interference with Other Services on the Host

Symptoms: Capacity Problems

- Poor user experience due to slow operations
- Agent timeouts and retry can result in extra load

Cause

- CPU cycles
- Memory issues

Symptoms: Interference with Other Services on the Host

- Poor user experience due to slow operations
- Agent timeouts and retry may result in extra load

Cause

- CPU cycle contention
- Memory contention
- File system full

Diagnosis: OAM Server

1. Shut down the OAM Server
2. Try to access a WebGate or mod_osso protected resource
3. Bring up the OAM Server
4. Use the Access Tester to test authentication and authorization as described in Chapter 10.
5. Use 'top' to figure out the CPU and Memory consumption of the OAM Server as you use the access tester
6. Get a thread dump of the OAM server.

Diagnosis: OAM Admin Server

1. Shut down the OAM AdminServer
2. Restart your browser and access a protected resource, which should work.
3. Use remote registration to register a new partner, as described in Chapter 6 (this should fail).
4. Startup OAM AdminServer.

H.1.4 About Agent-Side Configuration and Load Issues

This topic provides symptoms, probable cause, and steps to diagnose time issues between agents and servers.

Symptoms: Difference in Clock time Between Agent and Server

- High CPU usage at both agent and server
- User experiences a system hang

Cause

- Agent thinks the token issued by the server is invalid
- Agent keeps going back to the server to re-issue the token

Diagnosis

1. Access protected resource.

2. Confirm: Client access hangs.
3. Confirm: High CPU usage on agent and server.

H.1.5 About Runtime Database (Audit or Session Data) Issues

The audit and session functions are both write intensive operations. The policy database can be tuned for read intensive service.

Symptoms

- Audit and session operations are slow
- File system on the OAM Server is full with audit data that is not yet written to the database
- Loss of in-memory session data when one of the servers in the cluster fails

Cause

- Database is not tuned for write intensive operations
- Database is unavailable due to maintenance
- Space issues in the database

Diagnosis

1. Shut down the database used to store Audit and Session data.
2. Try to access a protected resource.
3. Review error and warning messages in the OAM Server log files, as described in Chapter 13 (alternatively, in Chapter 16).

H.1.6 About Change Propagation or Activation Issues

This topic provides symptoms, probable cause, and steps to diagnose the following issues:

Symptoms

- Changes to policy do not take immediate effect
- Changes to system configuration do not take immediate effect

Cause

- Servers being too busy handling runtime requests (CPU contention)
- Coherence network slowness

Diagnosis: See "About Policy Store Database Issues"

H.1.7 About Policy Store Database Issues

This topic provides symptoms, probable cause, and steps to diagnose policy database issues.

Symptoms: No policy changes are allowed; no impact on runtime

Cause

- Database is unavailable (down for maintenance)

- Space issues in the database

Diagnosis

1. Shut down the database containing OAM policies.
2. Try to access a protected resource and observe the runtime access is not impacted.
3. Try to access the OAM Administration Console to edit policies, and then observe errors in the AdminServer log file.

H.2 Authentication Issues

This section provides the following information:

- Anonymous Authentication Issues

H.2.1 Anonymous Authentication Issues

Problem

Challenge Redirect URL can be NULL; however, Challenge Method cannot be NULL.

If you open the Anonymous authentication scheme to edit, and click Apply without adding a value for Challenge method, the following errors might appear:

```
Messages for this page are listed below.

* Challenge Method You must make at least one selection.

* Challenge Redirect You must enter a value.
```

Solution

You must include both a challenge method and a challenge redirect whenever you edit an anonymous authentication scheme.

H.3 Authorization Issues

Problem: Constraint Error

An error is logged in the oam-server diagnostic log file whenever you create or edit an IPv4 range or temporal constraint:

```
.... refreshPolicy specified but no response collector supplied
```

Cause

This is a message that is erroneously being logged at the ERROR level. The correct level of the message is INFO.

H.4 Cannot Access Authentication LDAP or Database

If the LDAP directory that is used for authentication is down or inaccessible (or the database that is configured as the policy store), it might be due to a heavy load or a timeout. You see a message when attempting to a protected resource that uses this LDAP or policy store.

Solution

1. Manually shut down the registered LDAP or database.
2. Restart the registered LDAP or database.

H.5 Cannot Find Configuration

H.5.1 Configuration Does Not Exist ...

If you attempt to create and apply configuration details for an OAM Server before configuring the OAM Server in the WebLogic Server domain, a message informs you of the following:

```
Configuration does not exist for path
/DeployedComponent/Server/oamServer/Instance/test

For more information, please see the server's error log for
an entry beginning with: Server Exception during PPR, #6.
```

To resolve this issue, you must configure the OAM Server in the WebLogic Server domain before you register the configuration with OAM 11g.

H.6 Could Not Find Partial Trigger

In the Administration Server output, you might see a "Could Not Find Partial Trigger" error (multiple times for each selected node when you click Policy Cconfiguration tab or a host identifier node) and then you click any of other nodes in the navigation tree.

This does not block functionality.

H.7 Deployments with Freshly Installed OAM 10g WebGates

Use the OAM server’s diagnostic features to debug on the OAM Server side. This section includes the following topics:

- Authentication Issues with OAM 10g WebGates
- Logout Issues with OAM 10g WebGates

> **See Also:** Chapter 11, "Configuring Centralized Logout for OAM 11g"

H.7.1 Authentication Issues with OAM 10g WebGates

Use the following methods to troubleshoot authentication issues when you have freshly installed OAM 10g WebGates in your OAM 11g deployment.

- Confirm that your request was protected using an http header trace like Internet Explorer HTTP Headers or Firefox Live HTTP Headers
- Confirm that the request is sent to the OAM server for authentication
  - GET /oam/server/obrareq.cgi?…..
  - Host: oam-server:port

H.7.2 Logout Issues with OAM 10g WebGates

Use the following methods to troubleshoot logout issues when you have freshly installed OAM 10g WebGates in your OAM 11g deployment.

- Make liberal use of HTTP Header Trace
- Confirm that the specific logout.html was copied to /access/oamsso folder in the 10g WebGate installation directory. If not present, you must create the logout.html as described in "Configuring Centralized Logout for 10g WebGate with OAM 11g Servers" on page 11-7.
- Change the OAM 10g WebGate's httpd.conf to remove the following lines:

```
<LocationMatch "/oamsso/*">
Satisfy any
</LocationMatch>
```

- From the OAM Administration Console, confirm that the LogoutUrls parameter (/oamsso/logout.html) is configured for this WebGate

H.8 Diagnosing OAM 11g Initialization and Performance Issues

This section includes the following topics:

- Diagnosing an Initialization Issue
- Diagnosing a Performance Issue
- Diagnosing Out-of-Memory Issues With a Heap Dump

H.8.1 Diagnosing an Initialization Issue

Problem

OAM Server does not start up.

Solution

1. Locate and review the OAM Server log file on the computer hosting the OAM Server.

   ```
   DOMAIN_HOME/servers/SERVER-NAME/logs/SERVER-NAME-diagnostics.log
   ```

2. Enable logging for this computer, as described in Chapter 13, "Logging Component Event Messages":

   ```
   DOMAIN_HOME/config/fmwconfig/servers/SERVER-NAME/logging.xml
   ```

3. Restart the OAM Server, observe the behavior, check the log file again if needed.

H.8.2 Diagnosing a Performance Issue

Problem

Monitoring the OAM Server reveals a significant spike in latency during authentication.

Solution

1. Locate and review the OAM Server log file on the computer hosting the OAM Server.

```
DOMAIN_HOME/servers/SERVER-NAME/logs/SERVER-NAME-diagnostics.log
```

2. Enable logging for this computer, as described in Chapter 13, "Logging Component Event Messages":

```
DOMAIN_HOME/config/fmwconfig/servers/SERVER-NAME/logging.xml
```

3. Restart the OAM Server, observe the behavior, check the log file again if needed.

H.8.3 Diagnosing Out-of-Memory Issues With a Heap Dump

Problem

Debugging for all expression parsing and evaluation produced a significant performance drag within ~20 hours due to memory growth; running out of memory in ~50 hours.

Configuration: 2GB heap; 3 minute session timeout; jdbc connections tuned min=32 max=200; jdbc connection idle timeout disabled; jbo pool size min = 10 & max=150

Solution

To generate heap-dumps for comparison, you use the following command-line tools jmap for Sun jvm or jrcmd for jrockit jvm located under JAVA_HOME/bin.

For jrockit jvm

```
jrcmd pid <command>
/jrockit_160_14_R27.6.5-32/bin/jrcmd 16775 heap_diagnostics
/jrockit_160_14_R27.6.5-32/bin/jrcmd 16775 print_threads
/jrockit_160_14_R27.6.5-32/bin/jrcmd 16775 jrarecording ....
```

For Sun jvm

```
jmap -histo <pid>
jmap -dump:live,format=b,file=heap.bin <pid>
```

H.9 Disabling Windows Challenge/Response Authentication on IIS Web Servers

The IIS Web server on Windows supports Challenge/Response Authentication, which defaults to On when IIS is installed. This enables users to use their domain log-ins when requesting resources from IIS and can conflict with Oracle Access Manager's authentication.

For example, on the first request from an Internet Explorer (IE) browser to a resource on IIS protected by Oracle Access Manager with a basic authentication scheme, IE displays a login dialog box requesting a domain along with the user name and password login provided by Oracle Access Manager.

To disable Windows challenge/response authentication

1. Launch the Microsoft Management Console for IIS.
2. Select the Web Server Host under Internet Information Server in the left hand panel.
3. Right click and select Properties.

4. Scroll down and select Edit the Master Properties for WWW Service.
5. Select the Directory Security tab.
6. Select Edit Anonymous Access and Authentication Control.
7. Complete the appropriate step for your platform:

   Windows 2000: Clear the Integrate Windows Authentication box.
8. Click OK.
9. In the Windows IIS properties screen, click OK.
10. Close the Microsoft Management Console.

H.10 IIS Web Server Issues

The following topics are provided to assist you:

- Form Authentication or Pass-Through Not Working
- IIS and General Web Component Guidelines
- Issues with IIS v6 Web Servers
- Page Cannot Be Displayed Error
- Removing and Reinstalling IIS DLLs

H.10.1 Form Authentication or Pass-Through Not Working

If form authentication or pass-through functionality is not working, the problem might be that either "UseWebGateExtForPassthrough" parameter is not set to true in the WebGate profile or that webgate.dll is not configured as Wild Card Application Mapping in IIS. In such cases, WebGate does not perform authentication or authorization for HTTP "POST" requests for the resources protected by form-based authentication.

Solution: Confirm that the `UseWebGateExtForPassthrough` parameter is configured in the WebGate profile with a value of `true` and that webgate.dll is configured as Wild Card Application Mapping.

H.10.2 IIS and General Web Component Guidelines

Following are some general guidelines to follow when installing Oracle Access Manager WebGates with IIS Web servers.

Account Privileges: The account that performs Oracle Access Manager installation must have administration privileges. The user account that is used to run OAM services must have the "Log on as a service" right, which can be set by selecting **Administrative Tools, Local Policy, Local Policies, User Rights Assignments, Log on as a service.**

IIS 6 Web Servers: You must run the WWW service in IIS 5.0 isolation mode. This is required by the ISAPI postgate filter. During Oracle Access Manager installation, this is usually set automatically. If it is not, you must set it manually for the Default Web site.

WebGate for IIS 7 Web Server: To use Form-based authentication without enabling pass through functionality (for example, "access/oblix/apps/webgate/bin/webgate.dll" is an action in the Form-based authentication scheme), ensure that the entry "<add segment="bin"/>" is not present in the applicationHost.config file. If the entry is present, you must remove it. Use the following steps to check this entry:

- Go to Windows\System32\inetsrv\config and open the file applicationHost.config.
- Search for the `<hiddenSegments>` module and remove the entry **`<add segment="bin"/>`** if it is present.

WebGate: When installing IIS WebGates, setting various permissions for the /access directory is required for IIS WebGates only when you are installing on a file system that supports NTFS. For example, suppose you install the ISAPI WebGate in Simple or Cert mode on a Windows 2000 computer running the FAT32 file system. The last installation panel provides instructions for manually setting various permissions that cannot be set on the FAT32 file system. In this case, these instructions may be ignored.

H.10.3 Issues with IIS v6 Web Servers

On IIS 6 Web servers only, you must run the WWW service in IIS 5.0 isolation mode, which is a requirement of the ISAPI postgate filter. This scenario will work if you have 32-bit Oracle Access Manager binaries running on a 32-bit Windows operating system. However, there is an issue if you attempt to run a 32-bit postgate.dll on a 64-bit Windows machine with IIS running in 32-bit mode.

Problem

When running IIS in IIS5.0 isolation mode, you see the following message:

"ISAPI Filter 'C:\webgate\access\oblix\apps\webgate\bin\webgate.dll' could not be loaded due to a configuration problem.

Cause

The current configuration only supports loading images built for an AMD 64-bit processor architecture. The data field contains the error number.

Solution

To learn more about this issue, including how to troubleshoot this kind of processor architecture mismatch error, see the following Web site:

`http://go.microsoft.com/fwlink/?LinkId=29349`

For more information, see Help and Support Center at:

`http://go.microsoft.com/fwlink/events.asp`

Problem

IIS5 never existed as 64-bit. However, IIS v6's IIS5 Compatibility Mode on 64-bit Windows computers only runs as 64-bit.

Cause

It is architecturally impossible run IIS5 Isolation Mode 32- bit on 64-bit Windows, as described in documentation available through the following URLs:

```
http://www.microsoft.com/communities/newsgroups/en-us/default.aspx?dg=microsoft.pu
blic.inetserver.iis&tid=5dd07102-8896-40cc-86cb-809060fa9426&cat=en_US_
02ceb021-bb43-476d-8f8f-6c00a363ccf5&lang=en&cr=US&p=1
```

```
http://blogs.msdn.com/david.wang/archive/2005/12/14/HOWTO-Diagnose-one-cause-of-W3
```

```
SVC-failing-to-start-with-Win32-Error-193-on-64bit-Windows.aspx
```

H.10.4 Page Cannot Be Displayed Error

A "The page cannot be displayed" error that appears after configuring WebGate for pass-through functionality, indicates a configuration issue.

Solution: Confirm that the `UseWebGateExtForPassthrough` parameter is configured in the WebGate profile with a value of `true` and that webgate.dll is configured as Wild Card Application Mapping.

H.10.5 Removing and Reinstalling IIS DLLs

When Oracle Access Manager is running with Microsoft's IIS Web server, you must manually uninstall and reinstall the following ISAPI filters when reinstalling Oracle Access Manager.

- `tranfilter.dll`
- `oblixlock.dll` (if you installed WebGate)
- `webgate.dll` (if you installed WebGate)

To remove and reinstall IIS DLLs

1. Uninstall Oracle Access Manager.
2. Manually uninstall the preceding DLLs.
3. Reinstall Oracle Access Manager.Active Directory.
4. Manually reinstall the DLLs.

> **Note:** These filters can change depending on the version of IIS you are using. If these filters do not exist or there are others present, contact Oracle to determine if the filters that are present need to be removed.

H.11 jps Logger Class Instantiation Warning is Logged on Authentication

A jps logger class instantiation warning is might appear on the back end upon authentication. However, this is a harmless warning and no action is required.

H.12 Login Failure for a Protected Page

Problem

After installing OAM and protecting a page using a physical host and port, register the partner application using the OHS physical host and port. Login fails to prompt the user for credentials when accessing the protected page. The log file shows that the URL is re-directed to a Virtual Host despite the fact that all configuration and registration is setup correctly.

Solution

Remove any Virtual Host Directives from httpd.conf when protecting a page using the Oracle HTTP Server (OHS) physical host and port.

H.13 OAM Metric Persistence Timer IllegalStateException: SafeCluster

Problem

After using the WebLogic Configuration Wizard to create an OAM Server cluster on two computers, and starting AdminServer, all servers start up properly. After shut down, a third server is added using the WebLogic Server Administration Console to create a new managed server and add it to the cluster. The third server goes into Running mode when started, with some exceptions in the start up log.

```
... Exception in thread "OAM Metric Persistence Timer"
```

Solution

in addition to the actions in the WebLogic Administration Console, you must register the server using the OAM Administration Console to ensure that the server can identify itself.

> **Note:** When adding and registering a second server instance for the same computer, all port numbers must differ: OAM Proxy port; the "port" that must match the one in the WebLogic Server Console; and the Coherence port.

For server registration details, see "Managing Individual OAM Server Registrations" on page 4-4.

H.14 Partial Cluster Failure and Intermittent Login and Logout Failures

Problem

In the event of a partial outage of Oracle Access Manager (on some, but not all instances of the cluster), end users might see intermittent login and logout failures.

Workarounds

1. Remove OHS from the deployment
2. Configure the OHS cluster such that each OHS instance is pinned to a WebLogic Server instance.
3. The WebLogic Server container with the malfunctioning Oracle Access Manager application must be removed from service (shutdown) and brought back up upon recovery.

H.15 Registration Issues

Problem: Remote Registration Tool Failure

Solution

Ensure that the agent name is unique (does not already exist) and that the AdminServer is running.

Problem: No ObAccessClient.xml File Generated

Solution

Protected and public resources must be described as relative URLs of the format '/index.html'. If the resource does not begin with a '/', no ObAccessClient.xml file will be generated.Verify the protected and public resource URLs and ensure all begin with a "/". For more information, see "About the Resource URL" on page 9-14.

Problem: Partner Registration Failure

Partner registration can fail if you do not supply a unique agent name, which is also used to create an application domain. The agent name and application domain name must be the same and must be unique. Using the oamreg validate command can fail when the agent name does not match the application domain name.

Solution

Ensure that the agent name and application domain name are the same.

H.16 Rowkey does not have any primary key attributes Error

While browsing across the Resources table in the Resource Type tab the following error message is logged:

```
@ <Error>
<oracle.adfinternal.view.faces.model.binding.CurrencyRowKeySet>
@ <BEA-000000> <ADFv: Rowkey does not have any primary key attributes. Rowkey:
oracle.jbo.Key[], table: model.ResTypeVOImpl@620289.>
```

This is harmless and does not hinder any functionality.

H.17 SELinux Issues

Delivered with Oracle Enterprise Linux, SELinux modifications provide a variety of policies through the use of Linux Security Modules (LSM) within the Linux kernel.

SELinux requires performing additional steps after installing Oracle Access Manager WebGates and before starting the associated Web server.

Problem

The following errors could be reported in WebServer logs/console when starting a Web server on Linux distributions that have more strict SELinux policies in place (after installing an Oracle Access Manager WebGate):

OAM 11g WebGate

```
$WebGate_OH/webgate/ohs/lib/webgate.so: cannot restore segment prot after reloc:
Permission denied.
```

OAM 10g WebGate

```
$WebGate_install_dir/access/oblix/apps/webgate/bin/webgate.so: cannot restore
segment prot after reloc:
Permission denied.
```

Cause

These errors are reported due to Secure Linux security context policies on files.

Solution

To avoid these errors and start the Web server, run following `chcon` commands to change the security context on files after installing each Oracle Access Manager Web component and before restarting the associated Web server. For more information on the `chcon` command, see your Linux documentation.

1. Run chcon -t texrel_shlib_t PATH_TO_LIBWEBPLUGINS.SO. For example:

   ```
   chcon -t texrel_shlib_t  /WebGate_install_dir/access/oblix/lib/webgate.so
   ... and libxmlengine.so
   ```

2. Run chcon -t texrel_shlib_t PATH_TO_LIBWEBGATE.SO. For example:

   ```
   chcon -t texrel_shlib_t  /WebGate_install_dir/access/oblix/apps/webgate/
   bin/webgate.so
   ```

H.18 SSL versus Open Communication

If both the SSL and Open ports of the Managed Server are enabled, then the Managed Server is set to the SSL port by default.

If you must use the non-ssl port, the credential collector URL the authentication scheme must be set to the absolute URL which points to 'http' as the protocol and non-ssl port.

H.19 Start Up Issues

Problem: Connection to OAM Server could not be established: Exception in connecting to server. Connection refused.

Cause:

This is normal and expected behavior for the Managed Server where the OAM Server runs because the IDMDomainAgent agent is started before the OAM Server.

The IDMDomainAgent is deployed on every WebLogic container. When the WebLogic container starts, the agent tries to connect to the OAM Server. If it fails to connect, this message is logged and the agent tries to establish the connection in subsequent requests. When the agent is successful, this message is no longer displayed.

Solution

If the connection to the OAM Server is not successful, the IDMDomainAgent falls back and the WebLogic container handles protection (including login), if it is configured.

H.20 Synchronizing OAM Server Clocks

The state of a session is the source of truth for relying parties. Synchronization of system clocks of the various Servers is required.

The system clock of the relying party might be out of synchronization with the SME clock. If the relying party’s clock is:

- Ahead of the session clock A relying party's request for authentication is made and the active sessionID is returned.
- Behind the session clock: Event notifications to the relying party help invalidate the session.

For example, if a Web server clock is ahead of the server clock, a request sent from the WebGate to the OAM Server will contain a time that, to the OAM Server, has not yet occurred. This can cause login events to fail. When running in Simple or Cert mode, time stamps might become out of sync, or the client certificate might appear to be invalid.

> **Note:** To avoid event notification issues, ensure that all OAM Server clocks are synchronized to Time Services such as NIST internet time service.

For successful operation:

- Ensure all computer clocks are synchronized. There is no tolerance level. If, for example, the WebGate clock is even slightly ahead of the OAM Server clock, a cookie generated by the WebGate will appear to be in the future and can cause problems in the OAM Server.
- Confirm that the clock on each computer running a WebGate is *not* running ahead of the OAM Servers with which it is associated. The OAM Server must be ahead of the WebGate clock by a maximum of 60 seconds.

H.21 Unable to Cancel Some Operations

You might have a problem cancelling an operation rather than applying changes. Errors might appear stating that you need to supply specific values. For instance, this can occur when you create an empty Host ID row and click you Cancel without providing input.

To recover, just close the page.

H.22 Using Oracle Coherence for Troubleshooting

> **WARNING: Oracle recommends that you do not modify Oracle Coherence settings unless requested to do so by an Oracle Support Representative.**

H.22.1 Troubleshooting OAM Servers Using Oracle Coherence Properties

Whether you are viewing Oracle Coherence settings for an individual server instance or Oracle Coherence details that are common to all OAM Servers, Oracle recommends that you do not modify Oracle Coherence settings unless requested to do so by an Oracle Support Representative.

Oracle Coherence is a JCache-compliant in-memory caching and data management solution for clustered Java EE applications and application servers. Oracle Coherence shares and manages data in an Oracle Coherence cluster by coordinating updates to the data using cluster-wide concurrency control, replicating and distributing data modifications across the cluster, and delivering notifications of data modifications to any servers that request them. Functionality such as HTTP Session Management is available out-of-the-box for applications deployed to Oracle WebLogic Server.

Oracle Coherence logging appears in the WebLogic Server log only. There is no bridge from Oracle Coherence logging to Oracle Access Manager logging.

See Also: Oracle Coherence documentation.

H.23 Validation Errors

Problem: Resource not added to Authentication or Authorization Policy

While creating an Authentication or Authorization Policy, if you add a resource that is already used in another Authentication or Authorization Policy, a validation error appears when you click Apply. This is expected.

If you click OK in the error window and then attempt to add a valid resource that is not used within another Authentication or Authorization Policy, the resource is not added and the Authentication or Authorization Policy is not created.

Solution

1. Click Apply and close the Authentication or Authorization Policy page.
2. From the navigation tree, click the named policy again, click the Edit to open the page, and add the new resource.

Problem: Validation Failure - "description" attribute is not valid

A validation error appears if you enter an optional description longer than 200 characters.

Solution

Keep optional descriptions to 200 characters in length and less than 10 lines.

H.24 Web Server Issues

The following issues with Web servers may arise:

- Access Server Fails on an Apache Web Server
- Apache v2 on HP-UX
- Apache v2 Bundled with Red Hat Enterprise Linux 4
- Apache v2 Bundled with Security-Enhanced Linux
- Apache v2 on UNIX with the mpm_worker_module for WebGate
- Domino Web Server Issues
- Errors, Loss of Access, and Unpredictable Behavior
- Known Issues for ISA Web Server
- Oracle HTTP Server Fails to Start with LinuxThreads
- Oracle HTTP Server WebGate Fails to Initialize On Linux Red Hat 4
- Oracle HTTP Server Web Server Configuration File Issue
- Issues with IIS v6 Web Servers
- PCLOSE Error When Starting Sun Web Server

- Removing and Reinstalling IIS DLLs

H.24.1 Access Server Fails on an Apache Web Server

Symptom: You are running an Apache Web server, and an Access Server fails, displaying the following message:

```
libthread panic: cannot create new lwp
(PID: 9035 LWP 2). stackrace:
ff3424cc
0
```

This symptom may be caused by the Apache Web server launching more instances of itself. This can happen when the server determines that more instances are needed to service the number of connections between one or more WebGates and the Access Server.

The additional instances create even more connections, which exceed the number of connections by the Access Server.

Solution: Reduce the number of MinSpareServers, MaxSpareServers, StartServers, and MaxClients parameters.

Go to the Access Server's configuration directory and open the http.d configuration file.

Recommended parameter settings:

- MinSpareServers 1
- MaxSpareServers 5
- StartServers 3
- MaxClients 5

H.24.2 Apache v2 on HP-UX

When running Apache v2 on HP-UX, do not use nobody for User or Group, because shared memory may not work. Instead, use your login name as User Name with a your group as Group Name On HP-UX (on Solaris, "www" is equivalent to "nobody").

When running Apache v2 on HPUX 11.11, ensure that the AcceptMutex directive in the Apache httpd.conf file is set to "fcntl". If the directive is not present, add it to the httpd.conf file (AcceptMutex fcntl). For more information, see:

http://issues.apache.org/bugzilla/show_bug.cgi?id=22484

H.24.3 Apache v2 Bundled with Red Hat Enterprise Linux 4

After installing a WebGate on vendor-bundled Apache, the Web server may give the following error upon startup:

```
Error: Cannot load libgcc_s.so.1 library - Permission denied.
```

Solution: Change the Security-Enhanced Linux (SELinux) policy rules for Oracle Access Manager WebGates as described in "Tuning Apache/IHS v2 for Oracle Access Manager WebGates" on page 18-28.

H.24.4 Apache v2 Bundled with Security-Enhanced Linux

Errors might be reported in WebServer logs/console when starting a Web server on Linux distributions, which have stricter SELinux policies in place, after installing an Oracle Access Manager Web component. You can avoid these errors by running appropriate `chcon` commands for the installed Web component before restarting the Web server.

See Also: "SELinux Issues" on page H-14

H.24.5 Apache v2 on UNIX with the mpm_worker_module for WebGate

The following item is required only if you compile Apache v2 for WebGate on UNIX with the mpm_worker_module. In this case, you need to modify the thread.c file from the Apache source for the UNIX environment. Making this change ensures that the default pthread stacksize for WebGate produces optimal performance during multithreaded server implementation. If this change is not made, the default pthread stack size would not be sufficient for WebGate and could result in a crash.

Apache 2.0 does not support the ThreadStackSize option. Therefore:

- With UNIX-based Apache v2.1 and later you must use the ThreadStackSize directive to set the size of the stack (for autodata) of threads that handle client connections and call modules to help process those connections.
- With UNIX-based Apache 2, it is best to use the compilable source while adding the mpm_worker_module and changing the thread.c file to avoid a stack overflow.

The following procedure shows how to modify the Apache v2.0 thread.c file to provide the default pthread stacksize needed by WebGate for optimal performance during multi-threaded server implementation. For details about the Apache v2.1+ ThreadStackSize directive, see `http://httpd.apache.org/docs/2.2/mod/mpm_common.html#threadstacksize.`

Note: The following procedure should be performed only for the Apache 2.0 WebGate. Otherwise, the default pthread stack size is not sufficient for the WebGate and could result in a crash.

To modify the Apache v2.0 thread.c file for WebGate in a UNIX environment

1. Locate the thread.c file. For example:

```
APACHE 2.0.52 source/srclib/apr/threadproc/unix/thread.c
```

2. Locate the function named apr_threadattr_create(apr_threadattr_t **new,apr_pool_t *pool) in the following code segment:

```
**new,apr_pool_t *pool) in the following code segment:
1-----> apr_status_t stat;
2
3-----> (*new) = (apr_threadattr_t *)apr_pcalloc(pool, sizeof(apr_threadattr_
t));
4-----> (*new)->attr = (pthread_attr_t *)apr_pcalloc(pool, sizeof(pthread_attr_
t));
5
6-----> if ((*new) == NULL || (*new)->attr == NULL) {
7----->             return APR_ENOMEM;
8-----> }
9
10----->(*new)->pool = pool;
```

```
11----->stat = pthread_attr_init((*new)->attr);
12
13-----> if (stat == 0) {
14----->            return APR_SUCCESS;
15-----> }
16----->#ifdef PTHREAD_SETS_ERRNO
17----->stat = errno;
18----->#endif
19
20----->return stat;
21
```

3. Add the following code before line 13 shown earlier.

```
int stacksize = 1 << 20;
pthread_attr_setstacksize(&(*new)->attr, stacksize);
```

4. Run configure, make, and make install to set up the Apache Web server with the mpm_worker_module.

H.24.6 Domino Web Server Issues

Failure Authentication Event: For Domino Web servers, the redirection of a URL through Oracle Access Manager may not work if the authentication type is set as Basic Over LDAP and the URL to be redirected is mentioned as one of the following:

Either a relative path present on the same Web server

Or the Full path URL on the same Web server containing a computer name defined in the host identifier string combinations.

To overcome a failure authentication event, you must set the redirected URL with a computer name that is not defined under the host identifier group. For example, the IP address of the computer.

This problem does not occur with a form-based authentication type.

Header Variables: It may not be possible to pass header variables other than REMOTE_USER to WebGates installed on Lotus Notes Domino Web servers when using Client Certificate authentication scheme.

For example, header variables cannot be set on the one request where Client Certificate authentication occurs. However, all other requests do allow header variables to be set.

For more information, see Chapter 21, "Configuring Lotus Domino Web Servers for 10g WebGates".

H.24.7 Errors, Loss of Access, and Unpredictable Behavior

Symptom: If you installed Oracle Access Manager on UNIX under a different user ID than you used to create your Web server instance, Oracle Access Manager can become unstable. Users may experience behavior such as:

- Random bug report pages
- Failure to write to log file errors
- Loss of access to Web pages

Solution: Change file permissions using the chown command. Change the Oracle Access Manager directory to the same user ID that you used to create your Web server instance.

H.24.8 Known Issues for ISA Web Server

WebGate uses ISAPI extension for displaying user deny error message and for displaying the diagnostic page. However, ISA 2006 does not support extensions. Therefore:

- If the user is denied access by WebGate, the user gets Page Cannot be displayed error message instead of Oracle Access Manager denied access error message.
- The following diagnostic URL does not work for ISA: http(s)://hostname:port/access/oblix/apps/webgate/bin/webgate.dll?progid=1 for webgate .

H.24.9 Oracle HTTP Server Fails to Start with LinuxThreads

After installing a WebGate instance on an Oracle HTTP Server, the server does not start up. This occurs because Oracle Access Manager uses an older Linux threading model.

> **Note:** When running Oracle Access Manager, LinuxThreads is used by default. This requires setting the environment variable LD_ASSUME_KERNEL to 2.4.19. If you are using NPTL with Oracle Access Manager, you do not set LD_ASSUME_KERNEL to 2.4.19

Solution: When using LinuxThreads mode, comment out the Perl module in the httpd.conf file, update the LD_ASSUME_KERNEL environment variable, and restart, as described in the following procedure.

To resolve the failure to start Oracle HTTP Server in LinuxThreads mode

1. Comment out the Perl module in the httpd.conf file in the following location:

   Oracle HTTP Server 11g: `ORACLE_INSTANCE/config/OHS/<ohs_name>/httpd.conf`

   Oracle HTTP Server v2: `OH$/ohs/conf/httpd.conf`

   Oracle HTTP Server v1.3: `OH$/Apache/Apache/conf/httpd.conf`

2. To update the LD_ASSUME_KERNEL value, open the following file in a text editor:

   ```
   OH$/opmn/conf/opm.xml
   ```

3. Find the following line:

   ```
   <process-type id="HTTP_Server" module-id="OHS">
   ```

   Add the following information under the line you found in the previous step:

   ```
   <environment>
   <variable id="LD_ASSUME_KERNEL" value="2.4.19" />
   </environment>
   ```

4. Save this file.
5. Run the following commands to implement your changes:

   ```
   opmnctl stopall
   opmnctl startall
   ```

H.24.10 Oracle HTTP Server WebGate Fails to Initialize On Linux Red Hat 4

This situation might arise whether you are using Oracle Access Manager with LinuxThreads or NPTL.

Symptom: WebGate fails to initialize when installed on an Oracle HTTP Server running Red Hat Enterprise Server version 4.0 with a kernel version lower than 2.6.9-34.EL. Version 2.6.9-34.EL is supplied with the Red Hat version 4, update 3.

Solution: To prevent this problem, you must upgrade to Red Hat version 4, update 3 or higher.

H.24.11 Oracle HTTP Server Web Server Configuration File Issue

Problem

With Oracle Application Server 10.1.x, OC4J, when the httpd.conf file is modified automatically during WebGate installation, it can be corrupted.

Solution

Before installing WebGate, run the following command to prevent the httpd.conf file from being overwritten.

```
$ORACLE_HOME/dcm/bin/dcmctl updateConfig -ct ohs
```

H.24.12 Issues with IIS v6 Web Servers

On IIS 6 Web servers only, you must run the WWW service in IIS 5.0 isolation mode, which is a requirement of the ISAPI postgate filter. This scenario will work if you have 32-bit Oracle Access Manager binaries running on a 32-bit Windows operating system. However, there is an issue if you attempt to run a 32-bit postgate.dll on a 64-bit Windows machine with IIS running in 32-bit mode.

Problem

When running IIS in IIS5.0 isolation mode, you see the following message:

"ISAPI Filter 'C:\webgate\access\oblix\apps\webgate\bin\webgate.dll' could not be loaded due to a configuration problem.

Cause

The current configuration only supports loading images built for an AMD 64-bit processor architecture. The data field contains the error number.

Solution

To learn more about this issue, including how to troubleshoot this kind of processor architecture mismatch error, see the following Web site:

http://go.microsoft.com/fwlink/?LinkId=29349

For more information, see Help and Support Center at:

http://go.microsoft.com/fwlink/events.asp

Problem

IIS5 never existed as 64-bit. However, IIS v6's IIS5 Compatibility Mode on 64-bit Windows computers only runs as 64-bit.

Cause

It is architecturally impossible run IIS5 Isolation Mode 32- bit on 64-bit Windows, as described in documentation available through the following URLs:

```
http://www.microsoft.com/communities/newsgroups/en-us/default.aspx?dg=microsoft.pu
blic.inetserver.iis&tid=5dd07102-8896-40cc-86cb-809060fa9426&cat=en_US_
02ceb021-bb43-476d-8f8f-6c00a363ccf5&lang=en&cr=US&p=1

http://blogs.msdn.com/david.wang/archive/2005/12/14/HOWTO-Diagnose-one-cause-of-W3
SVC-failing-to-start-with-Win32-Error-193-on-64bit-Windows.aspx
```

H.24.13 PCLOSE Error When Starting Sun Web Server

Symptom: When attempting to start the Sun Web server, you get an error like the following:

```
Unable to start, PCLOSE
```

Solution: A number of problems can cause this error:

- A syntax error in your `obj.conf` file
- Leading spaces in your `obj.conf` file
- Installing Oracle Access Manager as a different user ID than what you used to create your Web server instance
- A carriage return at the end of the `obj.conf` file

H.24.14 Removing and Reinstalling IIS DLLs

When Oracle Access Manager is running with Microsoft's IIS Web server, you must manually uninstall and reinstall the following ISAPI filters when reinstalling Oracle Access Manager.

- `tranfilter.dll`
- `oblixlock.dll` (if you installed WebGate)
- `webgate.dll` (if you installed WebGate)

To remove and reinstall IIS DLLs

1. Uninstall Oracle Access Manager.
2. Manually uninstall the preceding DLLs.
3. Reinstall Oracle Access Manager.Active Directory.
4. Manually reinstall the DLLs.

> **Note:** These filters can change depending on the version of IIS you are using. If these filters do not exist or there are others present, contact Oracle to determine if the filters that are present need to be removed.

H.25 Windows Native Authentication

Problem

After setting up Windows Native Authentication, and accessing the WNA-protected page, the browser might give an error indicating that the user name and/or password are incorrect.

Cause

The Identity Store used by Oracle Access Manager might not point to Windows Active Directory. By default, the identity store is Embedded LDAP.

Solution

1. In the OAM Administration Console, review the identity store configuration: System Configuration, Data Sources, User Identity Store.
2. Confirm the LDAP store settings point to Active Directory.

Index