Oracle® Fusion Middleware

Administrator's Guide for Oracle Access Manager with Oracle Security Token Service

11g Release 1 (11.1.1)

E15478-05

July 2011

Describes how to manage common settings, agents, single sign-on policies, and tokens.

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

E15478-05

Primary Author: Gail Flanegin

Contributing Author: Damien Carru

Contributor: Patricia Fuzesy, Satish Madawand, Neelima Jadhav, Charles Wesley, Harshal X Shaw, Jeremy Banford, Rey Ong, Ramana Turlapati, Deepak Ramakrishnan, Vadim Lander, Vamsi Motokuru, David Goldsmith, Vishal Parashar, Carlos Subi

Contents

Preface xxxvii

What's New xli

Part I Introduction to Oracle Access Manager with Oracle Security Token Service

1 Oracle Product Introduction

Introduction to Oracle Access Manager 1-1
- Introduction to Oracle Access Manager Architecture 1-2
- Introduction to Oracle Access Manager Deployment Types and Installation 1-3
- Comparing Oracle Access Manager 11g, 10g, and OracleAS SSO 10g 1-5

Introduction to Oracle Security Token Service 1-10
- Oracle Security Token Service Key Terms and Concepts 1-11
- About Oracle Security Token Service with Oracle Access Manager 1-14
- About Integrated Oracle Web Services Manager 1-16
- About Oracle Security Token Service Architecture 1-18
- About Oracle Security Token Service Deployments 1-19
- About Installation Options 1-22
- About Oracle Security Token Service Administration 1-23

2 Introduction to This Book

Introduction to This Book 2-1

Part I: Oracle Product Introduction 2-1

Part II: Common Tasks 2-2
- Getting Started with Common Administration and Navigation 2-2
- Managing Services, Certificate Validation, and Common Settings 2-2
- Data Sources 2-3
- OAM Server Instances and the Console 2-4
- Oracle Access Manager Session Management 2-5

Part III, Oracle Access Manager Settings 2-5
- Access Manager Settings 2-5
- Single Sign-on Agents 2-6

Part IV, Single Sign-on, Oracle Access Manager Policies, and Testing 2-7
- Single Sign-On 2-7

Oracle Access Manager Policy Model and Shared Policy Components 2-8
Oracle Access Manager Policy Model, Application Domains, and Policies............................ 2-9
Connectivity and Policy Testing 2-10
Centralized Logout for Oracle Access Manager 11g.. 2-10
Part V: Oracle Security Token Service.. 2-10
Part VI: Common Logging, Auditing, Performance Monitoring.. 2-11
Component Event Message Logging 2-11
Webgate Event Message Logging.. 2-11
Common Audit Framework 2-11
Performance Metrics in the Oracle Access Manager Console 2-12
Performance Metrics in Fusion Middleware Control 2-12
Part VII: Using OAM 10g Webgates with OAM 11g 2-12
Provisioning OAM 10g Webgates for OAM 11g 2-13
Configuring 10g Webgates for Apache v2-based Web Servers (OHS and IHS).................. 2-13
Configuring 10g Webgates for the IIS Web Server 2-13
Configuring 10g Webgates for the ISA Server 2-13
Configuring Lotus Domino for OAM 10g Webgates.. 2-13
Part VIII: Appendixes... 2-13
Co-existence: OAM 11g SSO versus OAM 10g SSO with OracleAS SSO 10g 2-13
Moving OAM 11g From Test (Source) to Production (Target)... 2-14
Integration with Oracle ADF Applications 2-14
Internationalization and Multibyte Data Support for OAM 10g Webgates.......................... 2-14
Secure Communication and Certificate Management 2-14
Custom WebLogic Scripting Tool Commands for OAM 2-15
OAM 11g for IPv6 Clients... 2-15
Creating Deployment-Specific Pages 2-15
Troubleshooting 2-15

Part II Using the Console for Common Tasks

3 Getting Started with Common Administration and Navigation

Prerequisites... 3-1
Introduction to Administrators 3-2
Logging In to and Signing Out of Oracle Access Manager Console.. 3-3
Logging In to the Oracle Access Manager Console 3-3
Signing Out of Oracle Access Manager Console 3-4
Introduction to the Oracle Access Manager Console and Controls 3-4
Console Layout and Controls.. 3-5
Elements on a Page 3-12
Selecting Controls in the Console 3-13
Introduction to Policy Configuration and System Configuration Tabs 3-14
About the System Configuration Tab 3-14
About the Policy Configuration Tab 3-15
Viewing Configuration Details in the Console.. 3-17
Conducting Searches Using the Console... 3-17
Conducting Policy Element Searches Using the Console 3-18
Refining Searches for System Configuration Elements 3-19

Using Online Help 3-22
Command-Line Tools 3-22
Logging, Auditing, Monitoring Performance 3-23

4 Managing Services, Certificate Validation, and Common Settings

Prerequisites 4-1
Introduction to Common Configuration Elements 4-1
Enabling or Disabling Available Services 4-2
Managing the Common Settings 4-3
About Common Settings Pages 4-3
Managing Common Settings 4-4
Viewing Common Coherence Settings 4-5
Managing Global Certificate Validation and Revocation 4-6
About Certificate Validation and Revocation Lists 4-6
Managing Certificate Revocation Lists (CLRs) 4-7
Managing Certificate Validation 4-8
Configuring CDP 4-8

5 Managing Common Data Sources

Prerequisites 5-1
Introduction to Managing Common Data Sources 5-1
About User Identity Stores 5-2
About the Policy and Session Database Store 5-4
About the Oracle Access Manager Configuration Data File 5-5
About Oracle Access Manager Security Keys and the Embedded Java Keystore 5-5
About Oracle Security Token Service Keystores 5-6
Managing User Identity Stores 5-7
About the User Identity Store Registration Page 5-7
Registering a New User Identity Store 5-10
Viewing or Editing a User Identity Store Registration 5-11
Deleting a User Identity Store Registration 5-12
Setting the Default Store and System Store 5-12
About Setting the Default Store and System Store 5-12
Defining a Default Store and System Store 5-13
Managing the Administrators Role 5-14
About Managing the Administrator Role 5-14
Managing Administrator Roles 5-15
Managing the Policy Database by Using the Console 5-16
About Database Deployment for Oracle Access Manager 5-17
Configuring a Separate Database for Session Data 5-17

6 Managing Common OAM Server Registration

Prerequisites 6-1
Introduction to OAM Server Registration and Management 6-1
About Server Side Differences Between OAM 11g and OAM 10g 6-2
About Individual OAM Server Registrations 6-2

About the Embedded Proxy Server and Backward Compatibility 6-3
About OAM 11g SSO and Legacy OAM 10g SSO in Combination with OSSO 6-3
About Communication Between OAM Servers and Webgates 6-4
Managing Individual OAM Server Registrations 6-4
About the OAM Server Registration Page 6-5
Registering a Fresh OAM Server Instance 6-8
Viewing or Editing Individual OAM Server and Proxy Settings 6-9
Deleting an Individual Server Registration 6-10

7 Managing Sessions

Prerequisites 7-1
Introduction to User Sessions and Session Management 7-1
About the User Session Lifecycle 7-3
Oracle Coherence and Session Management 7-4
Configuring User Session Lifecycle Settings 7-6
About Common Session Lifecycle Setting Page 7-6
Viewing or Modifying Common Session Lifecycle Settings 7-7
Managing Active User Sessions 7-8
About the Session Management Page 7-8
Managing Active User Sessions 7-11
Verifying Session Management Operations 7-12
Security 7-14
Secure HTTPS Protocol 7-14
Coherence 7-14
Database Persistence 7-14

Part III Oracle Access Manager Settings Management

8 Configuring Access Manager Settings

Prerequisites 8-1
Introduction to Access Manager Settings 8-1
Managing Access Manager Load Balancing and Secure Error Modes 8-2
About Access Manager Load Balancing Settings and Secure Error Modes 8-2
Managing OAM Server Load Balancing and Secure Error Modes 8-4
Managing SSO Tokens and IP Validation 8-4
About Access Manager SSO Tokens and IP Validation Settings 8-4
Managing SSO Tokens and IP Validation 8-5
Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security 8-5
About Simple and Cert Mode Transport Security 8-6
About the Common OAM Proxy Page for Secure Server Communications 8-7
Viewing or Editing Simple or Cert Settings for OAM Proxy 8-7
Managing Run Time Policy Evaluation Caches 8-8
About Run Time Policy Evaluation Caches 8-8
Managing Run Time Policy Evaluation Caches 8-9
Managing Authentication Modules 8-10
About Default Authentication Modules and Pages 8-10

Creating a New Authentication Module of an Existing Type 8-13
Viewing or Editing Authentication Modules.. 8-14
Deleting an Authentication Module.. 8-14
Creating Custom Authentication Modules 8-15
About Creating Custom Authentication Modules 8-15
About the Custom Authentication Module Plug-ins.. 8-18
Creating a Custom Authentication Module... 8-22

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

Prerequisites.. 9-1
Introduction to Policy Enforcement Agents.. 9-1
About Policy-Enforcement Agents 9-2
About the Pre-Registered IAMSuiteAgent 9-4
About Registering Partners (Agents and Applications).. 9-8
About File System Changes and Artifacts for Registered Agents 9-9
Registering and Managing OAM Agents Using the Console 9-10
About Creating and Editing Webgate Registration 9-11
About User-Defined Webgate Parameters 9-21
About IP Address Validation for Webgates... 9-25
Searching for an OAM Agent Registration 9-25
Registering a Webgate or Programmatic Access Client 9-27
Viewing or Editing an OAM Agent Registration.. 9-28
Deleting Webgate Registration 9-30
Tuning 10g and 11g Webgate Caches 9-31
Introducing Webgate Caches 9-31
Reducing Network Traffic Between Components 9-34
Changing the Webgate Polling Frequency.. 9-34
Registering and Managing OSSO Agents Using the Console 9-35
About OSSO Agents and the OSSO Proxy 9-35
About the Create OSSO Agent Page 9-35
Refining the Search for an OSSO Agent (mod_osso) Registration 9-37
Registering an OSSO Agent (mod_osso) 9-38
Viewing or Editing OSSO Agent (mod_osso) Registration 9-39
Deleting an OSSO Agent (mod_osso) Registration... 9-40

10 Registering Partners (Agents and Applications) Remotely

Prerequisites.. 10-1
Introduction to Remote Partner Registration 10-1
About In-Band Remote Registration 10-2
About Out-of-Band Remote Registration 10-3
About Key Use, Generation, Provisioning, and Storage 10-4
About the Remote Registration Tool... 10-6
About Remote Registration Request Files 10-9
About Out-of-Band Registration Responses 10-22
Acquiring and Setting Up the Registration Tool 10-22
Creating the Registration Request... 10-23

Performing In-Band Remote Registration 10-24
Performing Out-of-Band Remote Registration 10-25
Validating Remote Registration and Resource Protection 10-26
Validating Remote Registration 10-26
Validating Authentication, Resource Protection, and Access After Remote Registration 10-27
Introducing Remote Management Modes 10-29
About Remote Agent Management Modes 10-29
About Remote Application Domain Management Modes 10-32
Managing Agents Remotely 10-39
Performing Remote Agent Updates 10-39
Performing Remote Agent Validation 10-40
Performing Remote Agent Removal 10-40
Creating or Updating an Application Domain Without an Agent 10-41

Part IV Managing Oracle Access Manager SSO, Policies, and Testing

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

Prerequisites 11-1
Comparing the OAM 11g Policy Model and OAM 10g Model 11-1
Introduction to the OAM 11g Policy Model 11-3
About Resource Types 11-5
About Host Identifiers 11-5
About Authentication, Schemes, and Modules 11-6
About Application Domains and Policies 11-7
About Resources and Resource Definitions 11-8
About Authentication Policies, Responses, and Resources 11-8
About Authorization Policies, Resources, Constraints, and Responses 11-9
Introduction to Configuring OAM Single Sign-On 11-10
Introduction to SSO Components 11-11
About Single Sign-On Components 11-11
About Single Sign-On Cookies During User Login 11-13
About Single Sign-On Cookies 11-14
Introduction to OAM 11g Single Sign-On Implementation Types 11-16
Application SSO 11-17
Single Sign-On with OAM 11g 11-17
Cross-Network Domains and Oracle Access Manager 11g 11-19
Introduction to OAM 11g SSO Processing 11-19
About SSO Log In Processing 11-19
About SSO Log In Processing with OAM Agents 11-20
About SSO Login Log In Processing with OSSO Agents (mod_osso) 11-22
About Single Sign-On Processing with Mixed Release Agents 11-24

12 Managing Policy Components

Prerequisites 12-1
Introduction to Managing Policy Components 12-1
Managing Resource Types 12-2

About Resource Types and Their Use 12-2
About the Resource Type Page 12-3
Searching for a Specific Resource Type 12-4
Managing Host Identifiers 12-5
About Host Identifiers 12-5
About Virtual Web Hosting 12-7
About the Host Identifier Page 12-11
Creating a Host Identifier 12-12
Searching for a Host Identifier Definition 12-13
Viewing or Editing a Host Identifier Definition 12-13
Deleting a Host Identifier Definition 12-14
Managing Authentication Schemes 12-15
About the Authentication Schemes Page 12-15
Creating an Authentication Scheme 12-26
Searching for a Authentication Scheme 12-27
Viewing or Editing a Authentication Scheme 12-28
Deleting an Authentication Scheme 12-28
Configuring Challenge Parameters for Encrypted Cookies 12-29
About ssoCookie Challenge Parameters for Encrypted Cookies 12-29
Configuring Challenge Parameters for Encrypted Cookie Security 12-30
Setting Challenge Parameters for Encrypted Cookie Persistence 12-30

13 Managing Policies to Protect Resources and Enable SSO

Prerequisites 13-1
Introduction to Application Domain Creation 13-2
About Automatic Application Domain Creation 13-2
About Manually Creating Application Domains 13-2
Anatomy of an Application Domain and Policies 13-3
Application Domain General Details 13-4
Default Resources in a Generated Application Domain 13-5
Default Authentication Policies in a Generated Application Domain 13-5
Default Authorization Policies in a Generated Application Domain 13-6
About Token Issuance Policies 13-7
Managing Application Domains using the Console 13-7
About the Application Domains Page 13-7
Creating a Fresh Application Domain Manually 13-8
Searching for an Application Domain 13-9
Viewing or Editing an Application Domain 13-10
Deleting an Application Domain and Its Content 13-10
Adding and Managing Resource Definitions for Use in Policies 13-11
About the Resource Definition Page in an Application Domain 13-11
Adding Resource Definitions to an Application Domain 13-18
Searching for a Resource Definition 13-19
Viewing or Editing a Resource Definition in an Application Domain 13-21
Deleting a Resource Definition from an Application Domain 13-22
Defining Authentication Policies for Specific Resources 13-22
About the Authentication Policy Page 13-23

Adding an Authentication Policy and Resources 13-24
Searching for an Authentication Policy 13-25
Viewing or Editing an Authentication Policy 13-26
Deleting an Authentication Policy 13-27
Defining Authorization Policies for Specific Resources 13-27
About Authorization Policies for Specific Resources 13-28
Adding an Authorization Policy and Specific Resources 13-29
Searching for an Authorization Policy 13-30
Viewing or Editing an Authorization Policy and Resources 13-30
Deleting an Authorization Policy 13-31
Introduction to Policy Responses for SSO 13-32
About Authentication and Authorization Policy Responses for SSO 13-32
About the Policy Response Language 13-33
About the Namespace and Variable Names for Policy Responses 13-34
About Constructing a Policy Response for SSO 13-35
About Policy Response Processing 13-36
Adding and Managing Policy Responses for SSO 13-37
Adding a Policy Response for SSO 13-37
Viewing, Editing, or Deleting a Policy Response for SSO 13-38
Introduction to Authorization Constraints 13-39
About Allow or Deny Type Constraints 13-40
About Classifying Users and Groups for Constraints 13-40
Guidelines for Authorization Responses Based on Constraints 13-41
About Constraints and General Authorization Policy Details 13-41
About the Add Constraint Window 13-42
About Identity Class Constraints 13-43
About IP4Range Class Constraints 13-45
About Temporal Class Constraints 13-45
Defining Authorization Policy Constraints 13-46
Defining Identity Class Constraints 13-46
Defining IP4Range Class Constraints 13-47
Defining Temporal Class Constraints 13-48
Viewing, Editing, or Deleting Authorization Policy Constraints 13-49
Validating Authentication and Authorization in an Application Domain 13-50
Example: Pre-seeded IAM Suite Application Domain and Policies 13-51

14 Validating Connectivity and Policies Using the Access Tester

Prerequisites 14-1
Introduction to the OAM 11g Access Tester 14-1
About OAM Agent and Server Interoperability 14-3
About Access Tester Security and Processing 14-4
About Access Tester Modes and Administrator Interactions 14-5
Installing and Starting the Access Tester 14-8
Installing the Access Tester 14-8
About Access Tester Supported System Properties 14-9
Starting the Tester Without System Properties For Use in Tester Console Mode 14-10
Starting the Access Tester with System Properties For Use in Command Line Mode 14-11

Introduction to the Access Tester Console and Navigation 14-12
Access Tester Menus and Command Buttons 14-13
Testing Connectivity and Policies from the Access Tester Console 14-15
Establishing a Connection Between the Access Tester and the OAM Server 14-16
Validating Resource Protection from the Access Tester Console 14-18
Testing User Authentication from the Access Tester Console 14-20
Testing User Authorization from the Access Tester Console 14-22
Observing Request Latency 14-23
Creating and Managing Test Cases and Scripts 14-24
About Test Cases and Test Scripts 14-24
Capturing Test Cases 14-25
Generating an Input Test Script 14-26
Personalizing an Input Test Script 14-27
Executing a Test Script 14-28
Evaluating Scripts, Log File, and Statistics 14-31
About Evaluating Test Results 14-31
About the Saved Connection Configuration File 14-32
About the Generated Input Test Script 14-33
About the Target Output File Containing Test Run Results 14-34
About the Statistics Document 14-36
About the Execution Log 14-38

15 Configuring Centralized Logout for OAM 11g

Prerequisites 15-1
Introduction to OAM 11g Centralized Logout 15-2
About Centralized Logout with OAM 11g Agents and Servers 15-3
About Centralized Logout with OAM 10g Agents and OAM 11g Servers 15-3
About Centralized Logout with the IAMSuiteAgent 15-3
About Centralized Logout with OSSO Agents (mod_OSSO) and OAM 11g 15-3
About Centralized Logout for Applications Using Oracle ADF Security 15-4
Configuring Centralized Logout for 11g Webgate with OAM 11g Server 15-4
About Configuring Centralized Logout for 11g Webgates 15-4
Configuring Centralized Logout for 11g Webgates 15-6
Configuring Centralized Logout for the IAMSuiteAgent 15-6
Configuring Centralized Logout for 10g Webgate with OAM 11g Servers 15-7
About Centralized Logout Processing for 10g Webgate with OAM 11g Server 15-7
About the Centralized Logout Script for OAM 10g Agents with OAM 11g Servers 15-8
Configuring Centralized Logout for 10g Webgates with OAM 11g 15-10
Configuring Centralized Logout for Oracle ADF-Coded Applications 15-12
About Centralized Logout Processing for Applications Coded to Oracle ADF Standards 15-12
Configuring Centralized Logout for ADF-Coded Applications with OAM 11g 15-13
Removing Custom mod_osso Cookies on Logout 15-15
Validating Global Sign-On and Centralized Logout 15-16
Confirming Global Sign-On 15-16
Validating Global Sign-On with Mixed Agent Types 15-16
Observing Centralized Logout 15-17

Part V Oracle Security Token Service

16 Oracle Security Token Service Implementation Scenarios

Prerequisites ... 16-1
Typical Token Ecosystem ... 16-1
Scenario: Identity Propagation with the OAM Token ... 16-2
Component Processing: Identity Propagation with the OAM Token ... 16-4
RST Attributes and Run Time Processing ... 16-5
Configuration Requirements: Identity Propagation with the OAM Token ... 16-7
Testing Your Implementation ... 16-14
Scenario: Web Service Security With On Behalf Of Username Token ... 16-15
Component interactions for Identity Propagation with Username Token... 16-16
RST Attributes and Processing for Identity Propagation with a Username Token ... 16-16
Configuration Requirements: Identity Propagation with the Username Token ... 16-18

17 Managing Oracle Security Token Service Settings and Set Up

Prerequisites ... 17-1
Introduction to Oracle Security Token Service Configuration ... 17-1
Post-Installation Configuration... 17-2
About Servers and Oracle Security Token Service... 17-3
About Oracle Security Token Service Clients ... 17-4
About Agents and Oracle Security Token Service ... 17-4
About Oracle Security Token Service End Points and Policies ... 17-5
Enabling and Disabling Oracle Security Token Service ... 17-8
About Oracle Security Token Service and the Oracle Access Manager Console... 17-8
About Enabling Services for Oracle Security Token Service ... 17-9
Enabling and Disabling Services for Oracle Security Token Service... 17-10
Defining Security Token Service Settings Using Oracle Access Manager Console ... 17-10
About Security Token Service Settings ... 17-10
Managing Security Token Service Settings ... 17-12
Using and Managing WSS Policies for Oracle WSM Agents... 17-13
Using and Modifying Web Service Security Policies... 17-13
Managing WSS Policies for Oracle Security Token Service: Classpath... 17-14
Managing WSS Policies for Oracle Security Token Service: Oracle WSM Policy Manager... 17-15
Configuring OWSM for WSS Protocol Communication... 17-15
About Oracle WSM Agent WS-Security Policies for Oracle Security Token Service... 17-16
Retrieving the Oracle WSM Keystore Password ... 17-16
Extracting the Oracle STS/Oracle WSM Signing and Encryption Certificate ... 17-17
Adding Trusted Certificates to the Oracle WSM Keystore... 17-17
Validating Trusted Certificates in the Oracle WSM Keystore... 17-18
Managing and Migrating Oracle Security Token Service Policies... 17-19
About Managing and Migrating Oracle Security Token Service Policies... 17-19
Managing Oracle Security Token Service Policies ... 17-19
Migrating Oracle Security Token Service Policies... 17-19
Introduction to Logging Oracle Security Token Service Messages... 17-20
Introduction to Auditing for Oracle Security Token Service ... 17-21

About Oracle Security Token Service Audit Record Storage 17-21
About Audit Reports and Oracle Business Intelligence Publisher 17-22
About the Audit Log.. 17-22
Auditing Oracle Security Token Service Administrative and Run-time Events 17-22
About Audit Record Content Common to All Events.. 17-22
Oracle Security Token Service Administrative Events You Can Audit 17-23
Oracle Security Token Service Run-time Events You Can Audit 17-25

18 Managing Oracle Security Token Service Certificates and Keys

Prerequisites.. 18-1
Introduction to Certificates and Keys for Oracle Security Token Service 18-1
About Keystores and Oracle Security Token Service 18-2
About the Oracle Web Services Manager Keystore (default-keystore.jks)............................ 18-3
About Using the OPSS Keystore for Requester Certificates 18-4
Managing Oracle Security Token Service Encryption/Signing Keys 18-4
Retrieving the System Keystore (.oamkeystore) Password 18-5
Adding a New Key Entry to the System Keystore (.oamkeystore) 18-5
Extracting an Oracle Security Token Service Certificate 18-6
Managing Partner Keys for WS-Trust Communications 18-7
About Partner Certificates 18-7
About Downloading the Relying Party's Certificate at Run Time.. 18-8
Setting the Partner's Signing or Encryption Certificate... 18-9
Managing Certificate Validation 18-9
Retrieving the Trust Anchors Store (amtruststore) Password... 18-10
Managing the Trust Anchors Store (amtruststore) 18-10
Managing Certificate Revocation Lists 18-11
Using a Custom Trust Anchor Store for Oracle Security Token Service 18-11

19 Managing Templates, Endpoints, and Policies

Prerequisites.. 19-1
Introduction.. 19-1
Searching for an Existing Template 19-2
About Template Search Controls 19-3
Searching for a Template 19-6
Managing Token Issuance Templates 19-7
About Managing Token Issuance Templates... 19-7
Managing a Token Issuance Template.. 19-14
Managing Token Validation Templates 19-15
About Managing Token Validation Templates 19-15
Managing Token Validation Templates 19-27
Managing Oracle Security Token Service Endpoints.. 19-29
About Managing Endpoints 19-29
Managing EndPoints 19-30
Managing Token Issuance Policies and Constraints with Oracle Access Manager.............. 19-31
About Token Issuance Policies... 19-31
About Managing Token Issuance Policies and Constraints 19-31

Managing Token Issuance Policies and Constraints 19-33
Managing TokenServiceRP Type Resources 19-34
About Managing TokenServiceRP Type Resources in Oracle Access Manager 19-35
Managing TokenServiceRP Type Resources in Application Domains 19-36

20 Managing Token Service Partners and Partner Profiles

Prerequisites 20-1
Introduction Token Service Partners and Partner Profiles 20-1
About Token Service Partners 20-1
About Partner Profiles 20-2
About Partner and Profile Data 20-2
Managing Token Service Partners 20-2
About Managing Token Service Partners 20-3
Managing a Token Service Partner 20-5
Refining Partner Searches 20-6
Managing Token Service Partner Profiles 20-7
About Managing Partner Profiles 20-7
Managing a Token Service Partner Profile 20-18
Refining a Profile Search 20-19

21 Troubleshooting Oracle Security Token Services

Authorization Issues 21-1
Endpoint Issues 21-2
Mapping Operation Issues 21-2

Part VI Common Logging, Auditing, Performance Monitoring

22 Logging Component Event Messages

Prerequisites 22-1
Introduction to Logging Component Event Messages 22-1
About Component Loggers 22-2
Sample Logger and Log Handler Definition 22-3
About Logging Levels 22-4
Configuring Logging for Oracle Access Manager 22-5
Modifying the Logger Level for Oracle Access Manager 22-5
Adding an Oracle Access Manager-Specific Logger and Log Handler 22-7
Configuring Logging for Oracle Security Token Service 22-8
Configuring Logging for Oracle Security Token Service 22-8
Defining the Log Level and Log Details for Oracle Security Token Service 22-9
Validating Run-time Event Logging Configuration 22-10

23 Logging Webgate Event Messages

About Logging, Log Levels, and Log Output 23-1
About Log Levels 23-2
About Log Output 23-3
About Log Configuration File Paths and Contents 23-4

Log Configuration File Paths and Names 23-4
Log Configuration File Contents 23-5
About Directing Log Output to a File or the System File........ 23-9
Structure and Parameters of the Log Configuration File........ 23-10
The Log Configuration File Header 23-11
The Initial Compound List........ 23-11
The Simple List and Logging Threshold 23-11
The Second Compound List and Log Handlers 23-13
The List for Per-Module Logging 23-14
The Filter List........ 23-14
About XML Element Order 23-15
About Activating and Suppressing Logging Levels........ 23-16
About Log Handler Precedence........ 23-16
Mandatory Log-Handler Configuration Parameters 23-17
Settings in the Default Log Configuration File 23-18
Configuring Different Threshold Levels for Different Types of Data........ 23-21
About the MODULE_CONFIG Section 23-22
Configuring a Log Level Threshold for a Function or Module........ 23-24
Filtering Sensitive Attributes........ 23-26

24 Auditing Administrative and Run-time Events

Prerequisites........ 24-1
Introduction to Auditing 24-1
About Oracle Access Manager Auditing Configuration 24-2
About Oracle Access Manager Audit Record Storage........ 24-3
About Audit Reports and Oracle Business Intelligence Publisher 24-4
About the Audit Log........ 24-5
Oracle Access Manager Events You Can Audit 24-5
Oracle Access Manager Administrative Events You Can Audit........ 24-5
OAM Run-time Events You Can Audit 24-8
About Authentication Event Auditing 24-10
Setting Up Auditing for Oracle Access Manager with Oracle Security Token Service........ 24-11
Setting Up the Audit Database Store........ 24-11
Preparing Oracle Business Intelligence Publisher EE........ 24-12
About the Auditing Configuration Section in Oracle Access Manager Console........ 24-13
Adding, Viewing, or Editing Common Audit Settings within Oracle Access Manager.... 24-14
Validating Oracle Access Manager Auditing and Reports 24-14

25 Monitoring Performance by Using Oracle Access Manager Console

Introduction to Performance Monitoring........ 25-1
Monitoring Server Performance Metrics Using the Console........ 25-2
Monitoring Server Instance Performance........ 25-2
Reviewing Server Metrics 25-2
Monitoring SSO Agent Performance Metrics........ 25-5
Monitoring SSO Agent Performance Metrics 25-5
Reviewing OAM Agent Metrics 25-5

Reviewing OSSO Agent Metrics 25-6
OXM Proxy Performance Tuning Parameters 25-8
About OAM Proxy Metrics 25-8
OAM Proxy Server Tuning Parameters 25-8

26 Monitoring Performance and Logs with Fusion Middleware Control

Prerequisites 26-1
Introduction to Fusion Middleware Control 26-1
Logging In to and Out of Fusion Middleware Control 26-2
About the Login Page for Fusion Middleware Control 26-3
Logging In To Fusion Middleware Control 26-3
Logging Out of Fusion Middleware Control 26-4
Displaying Menus and Pages in Fusion Middleware Control 26-4
About the Farm Page in Fusion Middleware Control 26-4
About Context Menus and Pages in Fusion Middleware Control 26-5
Displaying Context Menus and Target Details in Fusion Middleware Control 26-8
Viewing Performance in Fusion Middleware Control 26-8
About Performance Overview Pages in Fusion Middleware Control 26-9
About the Metrics Palette and the Performance Summary Page 26-16
Displaying Performance Metrics in Fusion Middleware Control 26-19
Displaying Component-Specific Performance Details 26-20
Managing Log Level Changes in Fusion Middleware Control 26-21
About Dynamic Log Level Changes 26-21
Setting Log Levels Dynamically Using Fusion Middleware Control 26-25
Managing Log File Configuration from Fusion Middleware Control 26-25
About Log File Configuration 26-25
Managing Log File Configuration by Using Fusion Middleware Control 26-28
Viewing Log Messages in Fusion Middleware Control 26-29
About Finding, Viewing, and Exporting Log Messages 26-29
Viewing Logged Messages With Fusion Middleware Control 26-33
Displaying MBeans in Fusion Middleware Control 26-34
About the System MBean Browser 26-35
Managing Mbeans 26-37
Displaying Farm Routing Topology in Fusion Middleware Control 26-38
About the Routing Topology 26-38
Viewing the Routing Topology using Fusion Middleware Control 26-40

Part VII Using 10g Webgates with Oracle Access Manager 11g

27 Managing OAM 10g Webgates with OAM 11g

Prerequisites 27-1
Introduction to OAM 10g Agents for OAM 11g 27-2
About Replacing the IAMSuiteAgent with an OAM 10g Webgate 27-2
About Legacy OAM 10g Deployments and Webgates 27-2
About Installing Fresh OAM 10g Webgates to Use With OAM 11g 27-2
Provisioning a 10g Webgate with OAM 11g 27-4

Locating and Installing the Latest OAM 10g Webgate for OAM 11g........ 27-6
Preparing for a Fresh 10g Webgate Installation with OAM 11g........ 27-6
Locating and Downloading 10g Webgates for Use with OAM 11g 27-8
Starting Webgate 10g Installation........ 27-9
Specifying a Transport Security Mode........ 27-10
Requesting or Installing Certificates for Secure Communications........ 27-10
Specifying Webgate Configuration Details........ 27-11
Updating the Webgate Web Server Configuration........ 27-11
Finishing Webgate Installation 27-13
Installing Artifacts and Certificates 27-14
Confirming Webgate Installation 27-14
Configuring Centralized Logout for 10g Webgate with OAM 11g........ 27-14
Replacing the IAMSuiteAgent with an OAM 10g Webgate........ 27-15
Provisioning a 10g Webgate to Replace the IAMSuiteAgent 27-15
Installing a 10g Webgate to Replace the IAMSuiteAgent 27-18
Updating the WebLogic Server Plug-in 27-18
Confirming the AutoLogin Host Identifier for an OAM / OIM Integration 27-19
Configuring OAM Security Providers for WebLogic........ 27-20
Disabling the IAMSuiteAgent........ 27-24
Verification........ 27-24
Deploying Applications in a WebLogic Container........ 27-24
Removing a 10g Webgate from the OAM 11g Deployment........ 27-25

28 Configuring Apache, OHS, IHS for 10g Webgates

Prerequisites........ 28-1
About Oracle HTTP Server and Oracle Access Manager........ 28-1
About Oracle Access Manager with Apache and IHS v2 Webgates........ 28-2
About the Apache HTTP Server 28-3
About the IBM HTTP Server 28-3
About the Apache and IBM HTTP Reverse Proxy Server........ 28-3
About Apache v2 Architecture and Oracle Access Manager........ 28-4
Requirements for Oracle HTTP Server, IHS, Apache v2 Web Servers........ 28-5
Requirements for IHS2 Web Servers........ 28-6
Requirements for Apache and IHS v2 Reverse Proxy Servers........ 28-6
Requirements for Apache v2 Web Servers........ 28-6
Preparing Your Web Server 28-7
Preparing the IHS v2 Web Server 28-8
Preparing Apache and Oracle HTTP Server Web Servers on Linux........ 28-11
Preparing Oracle HTTP Server Web Servers on Linux and Windows Platforms 28-12
Setting Oracle HTTP Server Client Certificates........ 28-12
Preparing the Apache v2 Web Server on UNIX 28-12
Preparing the Apache v2 SSL Web Server on AIX........ 28-16
Preparing the Apache v2 Web Server on Windows 28-17
Activating Reverse Proxy for Apache v2 and IHS v2........ 28-19
Activating Reverse Proxy For Apache v2 Web Servers........ 28-19
Activating Reverse Proxy For IHS v2 Web Servers 28-20
Verifying httpd.conf Updates for Oracle Access Manager Webgates 28-22

Verifying Webgate Details 28-22
Verifying Language Encoding 28-25
Tuning Oracle HTTP Server for Oracle Access Manager Webgates 28-25
Tuning OHS /Apache Prefork and MPM Modules for OAM 28-26
Tuning Oracle HTTP Server /Apache Prefork Module 28-26
Tuning Oracle HTTP Server /Apache MPM Module 28-27
Kernal Parameters Tuning 28-27
Starting and Stopping Oracle HTTP Server Web Servers 28-27
Tuning Apache/IHS v2 for Oracle Access Manager Webgates 28-28
Removing Web Server Configuration Changes After Uninstall 28-30
Helpful Information 28-30

29 Configuring the IIS Web Server for 10g Webgates

Prerequisites 29-1
Webgate Guidelines for IIS Web Servers 29-1
Guidelines for ISAPI Webgates 29-2
Prerequisite for Installing Webgate for IIS 7 29-5
Prerequisite for Installing Any 10g Webgate for IIS 7 29-5
Prerequisite for Installing a 32-bit Webgate for IIS 7 29-6
Updating IIS 7 Web Server Configuration on Windows 2008 29-6
Completing Webgate Installation with IIS 29-7
Enabling Client Certificate Authentication on the IIS Web Server 29-7
Ordering the ISAPI Filters 29-8
Enabling Pass-Through Functionality for POST Data 29-9
Protecting a Web Site When the Default Site is Not Setup 29-13
Installing and Configuring Multiple 10g Webgates for a Single IIS 7 Instance 29-14
Installing Each IIS 7 Webgate in a Multiple Webgate Scenario 29-14
Setting the Impersonation DLL for Multiple IIS 7 Webgates 29-16
Enabling Client Certification for Multiple IIS 7 Webgates 29-17
Configuring IIS 7 Webgates for Pass Through Functionality 29-18
Confirming IIS 7 Webgate Installation 29-19
Installing and Configuring Multiple Webgates for a Single IIS 6 Instance 29-19
Installing Each Webgate in a Multiple Webgate Scenario 29-20
Setting the Impersonation DLL for Multiple Webgates 29-22
Enabling SSL and Client Certification for Multiple Webgates 29-23
Confirming Multiple Webgate Installation 29-24
Finishing 64-bit Webgate Installation 29-24
Setting Access Permissions, ISAPI filters, and Directory Security Authentication 29-25
Setting Client Certificate Authentication 29-26
Confirming Webgate Installation on IIS 29-26
Starting, Stopping, and Restarting the IIS Web Server 29-27
Removing Web Server Configuration Changes Before Uninstall 29-27

30 Configuring the ISA Server for 10g Webgates

Prerequisites 30-1
About Oracle Access Manager and the ISA Server 30-1
Compatibility and Platform Support 30-2

Installing and Configuring Webgate for the ISA Server 30-2
Installing Webgate with ISA Server 30-3
Changing /access Directory Permissions 30-3
Configuring the ISA Server for the ISAPI Webgate 30-3
Registering Oracle Access Manager Plug-ins as ISA Server Web Filters 30-4
Configuring ISA Firewall Policies for ISA Web Filters 30-4
Ordering the ISAPI Filters 30-6
Starting, Stopping, and Restarting the ISA Server 30-7
Removing Oracle Access Manager Filters Before Webgate Uninstall on ISA Server 30-7

31 Configuring Lotus Domino Web Servers for 10g Webgates

Prerequisites 31-1
Installing the Domino Web Server 31-1
Setting Up the First Domino Web Server 31-2
Starting the Domino Web Server 31-3
Enabling SSL (Optional) 31-3
Installing a Domino Security (DSAPI) Filter 31-4
Completing the Webgate Installation 31-5

Part VIII Appendixes

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

Prerequisites A-1
Introduction to Upgrading and Co-existence with OracleAS 10g SSO A-1
Pre- and Post-Upgrade Topology and Authentication Examples A-2
About Pre-Upgrade OSSO 10g Topology A-2
About Post-Upgrade Topology and Co-existence A-3
Introduction to Validating Post-Upgrade Co-Existence with OAM 11g A-5
About Post-Upgrade SSO A-6
About Post-Upgrade OSSO 10g Authentication A-6
Validating Post-Upgrade Co-existence A-8
Validating Post-Upgrade Registration and Policies A-8
Validating Post-Upgrade SSO with Oracle Access Manager Protected Resources A-11
Validating Post-Upgrade SSO with OSSO-Protected Resources A-12

B Transitioning OAM 11g from a Test to a Production Environment

Prerequisites B-1
Introduction to Deployment Scenarios and Data Types B-1
Introduction to Methods and Tools B-3
About New versus Existing Production Environments B-3
About Methods to Move from Test to Production B-4
About Migrating OSSO Information from One OAM Instance to Another B-6
About the WebLogic Scripting Tool Commands B-6
About Policy Conflict Resolution B-8
About Building a Dependency Tree for Each Application Domain B-8
Planning an OAM 11g Move from Test to Production B-8

Choose the Method B-9
Noting Differences Between Source and Target Environments B-9
Developing Deployment Inventories B-9
Developing Tests B-9
Understanding Change Propagation B-9
Scheduling and Notifications B-10
Backup and Recovery Strategies B-10
Moving OAM 11g From Test to Production B-10
Exporting OAM 11g Data from Test (Source) B-10
Importing OAM 11g to Production (Target) B-11

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

Introduction to Oracle Platform Security Services and Oracle Application Developer Framework C-1
Oracle Platform Security Services Single Sign-on Framework C-1
Oracle Application Developer Framework C-2
Integrating OAM 11g With Web Applications Using Oracle ADF Security and the OPSS SSO Framework C-2
Sample SSO Configuration for OAM 11g C-4
SSO Provider Configuration Details C-6
Confirming Application-Driven Authentication During Runtime C-7

D Internationalization and Multibyte Data Support for OAM 10g Webgates

Introduction to Internationalization and Multibyte Data Support D-1
Languages For Localized Messages in Oracle Access Manager D-1
Bi-directional Language Support D-3
UTF-8 Encoding D-3

E Securing Communication for Oracle Access Manager 11g

Prerequisites E-1
Introduction to Securing Communication Between OAM 11g Servers and Webgates E-1
About Certificates, Authorities, and Encryption Keys E-3
About Security Modes and X509Scheme Authentication E-3
About the Importcert Tool E-4
Generating Client Keystores for OAM Tester in Cert Mode E-5
Configuring Cert Mode Communication for OAM 11g E-6
About Cert Mode Encryption and Files E-6
Generating a Certificate Request and Private Key for OAM Server E-7
Retrieving the OAM Keystore Alias and Password E-7
Importing the Trusted, Signed Certificate Chain Into the Keystore E-8
Adding Certificate Details to Access Manager Settings E-10
Generating a Private Key and Certificate Request for Webgates E-10
Updating Webgate to Use Certificates E-11
Configuring Simple Mode Communication with OAM 11g E-12
About Simple Mode, Encryption, and Keys E-13
Retrieving the Global Passphrase for Simple Mode E-13

Updating Webgate Registration for Simple Mode E-14
Verifying Simple Mode Configuration E-15

F Introduction to Custom WLST Commands for Administrators

Prerequisites F-1
Introduction to WebLogic Scripting Tool Commands F-1
WLST Command Summary: Oracle Access Manager F-2
WLST Command Summary: Oracle Security Token Service F-4
Running WLST Commands F-7
Starting the WLST Shell and Logging In F-7
Changing the Request Cache Type in a High Availability Environment F-8

G Configuring OAM 11g for IPv6 Clients

Prerequisites G-1
Introduction to Oracle Access Manager 11g and IPv6 .... G-1
Configuring IPv6 with OAM 11g and Challenge Redirect G-2
Considerations G-3
Configuring IPv6: Separate Proxy for OAM 11g and Webgates G-3

H Creating Deployment-Specific Pages

How the Single Sign-On Server Uses Deployment-Specific Pages H-1
Change Password Page Behavior H-2
How to Write Deployment-Specific Pages H-3
Login Page Parameters H-3
Forgot My Password H-4
Change Password Page Parameters H-5
Single Sign-Off Page Parameters H-6
External Application Login Page Parameters H-6
Page Error Codes H-8
Login Page Error Codes H-8
Post-Login Messages H-9
Change Password Page Error Codes H-10
Change External Application Login Page Error Codes H-10
Adding Globalization Support H-10
Deciding What Language to Display the Page In H-11
Rendering the Page H-12
Guidelines for Deployment-Specific Pages H-12
Installing Deployment-Specific Pages H-12
Using policy.properties to Install Login, Single Sign-Off, and Change Password Pages H-12
Using policy.properties to Install Wireless Login and Change Password Pages H-13
Using policy.properties to Install External Application Login Pages H-13
Examples of Deployment-Specific Pages H-14
Using Custom Classes H-14
Adding an External Application H-14

I Troubleshooting

Introduction to OAM 11g Troubleshooting I-2

About System Analysis and Problem Scenarios I-2

About LDAP Server or Identity Store Issues I-3

About OAM Server or Host Issues I-3

About Agent-Side Configuration and Load Issues I-4

About Runtime Database (Audit or Session Data) Issues I-5

About Change Propagation or Activation Issues I-5

About Policy Store Database Issues I-6

Oracle Access Manager Console Inconsistent State I-6

AdminServer Won't Start if the Wrong Java Path Given with WebLogic Server Installation ... I-6

Agent Naming Not Unique I-7

Application URL Requirements I-7

Authentication Issues I-7

Anonymous Authentication Issues I-7

X.509 Protected Resource and Single Sign Off I-7

Authorization Issues I-8

Cannot Access Authentication LDAP or Database I-8

Cannot Find Configuration I-8

Configuration Does Not Exist I-8

Could Not Find Partial Trigger I-9

Denial of Service Attacks I-9

Protecting the OAM Server from Crashing Under Load I-10

Compensating for Network Latency I-10

Protecting OAM Servers from a Flood of HTTP Requests I-10

Deployments with Freshly Installed OAM 10g Webgates I-11

Authentication Issues with OAM 10g Webgates I-11

Logout Issues with OAM 10g Webgates I-11

Diagnosing OAM 11g Initialization and Performance Issues I-12

Diagnosing an Initialization Issue I-12

Diagnosing a Performance Issue I-12

Diagnosing Out-of-Memory Issues With a Heap Dump I-13

Disabling Windows Challenge/Response Authentication on IIS Web Servers I-13

Changing UserIdentityStore1 Type Can Lock Out Administrators I-14

IIS Web Server Issues I-14

Form Authentication or Pass-Through Not Working I-14

IIS and General Web Component Guidelines I-14

Issues with IIS v6 Web Servers I-15

Page Cannot Be Displayed Error I-16

Removing and Reinstalling IIS DLLs I-16

jps Logger Class Instantiation Warning is Logged on Authentication I-16

Languages and Translation I-16

Automatically Generated Descriptions Are Not Translated I-16

Locales, Languages, and Oracle Access Manager Console Login Page I-17

Console Looks Messy I-17

Login Failure for a Protected Page I-17

OAM Metric Persistence Timer IllegalStateException: SafeCluster I-17

Partial Cluster Failure and Intermittent Login and Logout Failures I-18
Registration Issues I-18
Rowkey does not have any primary key attributes Error I-19
SELinux Issues I-19
Session Issues I-20
Session Impersonation Not Enabled by Default I-20
Sessions with Oracle Access Manager with Oracle Identity Federation I-20
SSL versus Open Communication I-20
Start Up Issues I-21
Synchronizing OAM Server Clocks I-22
Using Coherence I-22
Validation Errors I-23
Web Server Issues I-24
Server Fails on an Apache Web Server I-24
Apache v2 on HP-UX I-25
Apache v2 Bundled with Red Hat Enterprise Linux 4 .... I-25
Apache v2 Bundled with Security-Enhanced Linux I-25
Apache v2 on UNIX with the mpm_worker_module for Webgate I-25
Domino Web Server Issues I-26
Errors, Loss of Access, and Unpredictable Behavior I-27
Known Issues for ISA Web Server I-27
Oracle HTTP Server Fails to Start with LinuxThreads I-27
Oracle HTTP Server Webgate Fails to Initialize On Linux Red Hat 4 .... I-28
Oracle HTTP Server Web Server Configuration File Issue I-28
Issues with IIS v6 Web Servers I-29
PCLOSE Error When Starting Sun Web Server I-29
Removing and Reinstalling IIS DLLs I-30
Windows Native Authentication I-30

List of Figures

1–1 Oracle Access Manager 11g Components and Services 1-2
1–2 Oracle Access Manager 11g Component Distribution 1-3
1–3 Oracle STS Architecture 1-19
1–4 Oracle STS Token Support 1-19
1–5 Token Translation at a Centralized Authority 1-20
1–6 Translating Tokens Behind a Firewall 1-21
1–7 Web Services SSO 1-22
3–1 Oracle Access Manager 11g Log In Page 3-3
3–2 Sign Out Link, Upper-right Corner 3-4
3–3 Oracle Access Manager Console Welcome Page 3-5
3–4 Sample Navigation Trees with Menu and Tool Bars 3-8
3–5 Menu and Tool Bar Above Common Configuration Navigation Tree 3-8
3–6 View Menu 3-10
3–7 Actions Menu 3-10
3–8 Tabs of Open Pages, and Page Controls 3-11
3–9 System Configuration Tab (Collapsed and Fully Expanded) 3-15
3–10 Policy Configuration, Shared Components, Collapsed Application Domains 3-16
4–1 Common Configuration Nodes in Navigation Tree 4-1
4–2 Available Services Page 4-2
4–3 Common Settings Page (Collapsed View) 4-3
4–4 Common Coherence Settings 4-5
4–5 OCSP/CDP Settings for Global Certificate Validation 4-7
4–6 Certificate Revocation List Dialog Box 4-7
5–1 Completed Registration for the Default Store 5-8
5–2 Completed Registration for System Store 5-10
5–3 Fresh Default Store and System Store Options 5-13
5–4 Default Store Designation 5-13
5–5 Common Settings Page: Default and System Identity Stores 5-13
5–6 System Store Registration with Access System Administrators Section 5-15
5–7 Add System Administrator Roles 5-15
6–1 OAM Server Registration Page with Proxy Tab Displayed 6-5
6–2 Coherence Page and Values for an Individual OAM Server 6-8
7–1 Session Data and the Role of Oracle Coherence 7-5
7–2 Session Details: Common Settings Page 7-6
7–3 Common Configuration: Session Management Page 7-8
8–1 Access Manager Settings 8-1
8–2 Access Manager Settings: Load Balancer 8-2
8–3 Access Manager Settings: SSO 8-5
8–4 Common Policy Evaluation Caches 8-9
8–5 Pre-configured Kerberos Authentication Module 8-10
8–6 Pre-Configured LDAP Authentication Module 8-11
8–7 Pre-Configured X509 Authentication Module 8-12
8–8 Custom Authentication Modules Node and General Subtab 8-16
8–9 Adding a Step and Associating a Plug-in 8-16
8–10 Custom Authentication Module Steps Subtab and Details Section 8-17
8–11 Custom Authentication Module Steps Orchestration Subtab 8-17
8–12 KerberosPlugin 8-19
8–13 Default KerberosPlugin Steps and Details 8-19
8–14 Default KerberosPlugin Steps and Orchestration 8-20
8–15 LDAPPlugin 8-20
8–16 Default LDAPPlugin Steps and Details 8-20
8–17 Default Orchestration of Steps for LDAPplugin 8-21
8–18 X509Plugin 8-21
8–19 X509Plugin Default Steps and Details 8-22

8–20 Default Orchestration for X509Plugin Steps 8-22
9–1 IAMSuiteAgent Configuration in the WebLogic Administration Console 9-6
9–2 IAMSuiteAgent Characteristics 9-7
9–3 Resources Protected by the IAMSuiteAgent 9-8
9–4 Create OAM 11g Webgate Page 9-11
9–5 Create OAM 10g Webgate Page 9-12
9–6 Confirmation Window and Expanded 11g Webgate Page with Defaults 9-15
9–7 Expanded OAM 10g Webgate Registration Page 9-16
9–8 Webgate Search Controls and Create Agent Buttons 9-26
9–9 Create OSSO Agent Page 9-36
9–10 OSSO Agent Page and Confirmation Window 9-37
10–1 Key Generation 10-5
10–2 OSSOUpdateAgentRequest.xml 10-30
10–3 OAM11GUpdateAgentRequest.xml 10-31
10–4 OAMUpdateAgentRequest.xml 10-32
10–5 CreatePolicyRequest.xml 10-35
10–6 UpdatePolicyRequest.xml 10-36
10–7 Remote Management Elements for Application Domains 10-37
11–1 Oracle Access Manager 11g Policy Model and Shared Components 11-4
11–2 SSO Log-in Processing with OAM Agents 11-21
11–3 SSO Login Processing with OSSO Agents 11-23
12–1 Policy Components: Relationship to an Application Domain 12-2
12–2 Default wl_authen Resource Type Definition 12-3
12–3 TokenServiceRP 12-4
12–4 Host Identifier Page 12-12
12–5 Default LDAPScheme Page 12-16
13–1 Application-Specific Components of the OAM Policy Model 13-4
13–2 New Application Domain Generated During Agent Registration 13-4
13–3 Default Resource Definition in a Generated Application Domain 13-5
13–4 Default Authentication Policy for Protected Resources 13-6
13–5 Default Authorization Policy for Protecting Resources 13-7
13–6 Fresh Application Domains General Page 13-8
13–7 Application Domains Navigation Tree 13-8
13–8 Resources Page in an Application Domain 13-12
13–9 Searching for Resource Definitions within an Application Domain 13-19
13–10 Search Results Table for Resource Definitions in an Application Domain 13-20
13–11 Authentication Policy: IAM Suite Application Domain 13-23
13–12 Authorization Policy Page: IAM Suite Application Domain 13-28
13–13 Authorization Policy Response in the Console 13-33
13–14 Simple Response Samples 13-35
13–15 Sample Complex Responses 13-36
13–16 Authorization Policy Page, General Details 13-41
13–17 Add Constraint Window 13-42
13–18 Constraint Containers on the Authorization Policy Page 13-43
13–19 Identity Class Constraint Details: Selected User and Groups Table 13-44
13–20 Identity Class Add User Population Entries Window 13-44
13–21 Selected User and Groups Window 13-45
13–22 IP4Range Class Constraints 13-45
13–23 Temporal Constraint Class Details Page 13-46
13–24 OAM Admin Console Policy, Scheme, and Resources 13-51
13–25 Protected HigherLevel Policy, LDAP Scheme, and Resources 13-52
13–26 Protected LowerLevel Policy, Autentication Scheme, and Resources 13-52
13–27 Public Policy, Anonymous Scheme, and Resources 13-53
13–28 IAM Suite Authorization Policy 13-54
13–29 IAM Suite Token Issuance Policy and Resource URLs 13-54

14–1 OAM Agent (PEP) and OAM Server (PDP) Inter-operability 14-3
14–2 User Interactions with the Access Tester 14-7
14–3 Access Tester Console 14-12
14–4 Server Connection Panel in the Access Tester 14-16
14–5 Protected Resource URI Panel in the Access Tester 14-18
14–6 Access Tester User Identity Panel 14-21
14–7 Test Case Workflow 14-25
16–1 Typical Token Ecosystem 16-2
16–2 Identity Propagation with the OAM Token 16-3
16–3 Process Flow During Identity Propagation 16-3
16–4 Identity Propagation Deployment 16-4
16–5 Identity Propagation Processing 16-4
16–6 Required v1.0 WebLogic Server Identity Assertion Providers 16-9
16–7 IAP-OSTS Details 16-10
16–8 LDAP Provider: IAP-DSEE 16-11
16–9 Default Identity Store Defined in Oracle Access Manager 16-12
16–10 Token Issuance Policy for Identity Propagation 16-12
16–11 Authentication Policy for Identity Assertion by Webgate 16-13
16–12 /wssuser Endpoint for Identity Assertion 16-13
16–13 Default Identity Store Defined in Oracle Access Manager 16-19
16–14 Token Issuance Policy for Identity Propagation 16-20
16–15 /wss11user Endpoint for Identity Assertion 16-20
17–1 Default Endpoints, Policies, and Validation Templates 17-5
17–2 WS-Security 1.0 and 1.1 Policies 17-7
17–3 Oracle Access Manager with Oracle Security Token Service Enabled 17-9
17–4 Security Token Service Settings Page 17-10
19–1 Validation Templates Search Controls 19-3
19–2 Issuance Template Search Controls 19-3
19–3 Issuance Template: General Details and Defaults 19-8
19–4 Issuance Properties: Username Token Type 19-8
19–5 Issuance Properties: SAML Token Types 19-9
19–6 Security Details: SAML Tokens 19-11
19–7 New Validation Template page: General Page Defaults 19-17
19–8 New Validation Template: General Authentication Details 19-18
19–9 Token Mapping: SAML2 WS-Security Validation Template 19-21
19–10 Token Mapping, username-wstrust-validation-template 19-21
19–11 Token Mapping: x509-wss-validation-template 19-22
19–12 Endpoints Page 19-29
19–13 IAM Suite Token Issuance Policy and Resource URLs 19-31
19–14 Token Issuance Policies and Constraints 19-32
19–15 Pre-defined Resource Type: TokenServiceRP 19-35
19–16 Search: Resource Type TokenServiceRP in Application Domain 19-36
20–1 New Requester Partner Page 20-3
20–2 Defined Requester Partner 20-5
20–3 Partner Search Controls 20-7
20–4 Requester Profile: General 20-7
20–5 Requester Profile: Token and Attributes 20-9
20–6 Relying Party Profile Token and Attributes 20-10
20–7 Token and Attributes: Issuing Authority 20-14
20–8 Issuing Authority Profile: Token Mapping Tab 20-16
20–9 Search Profiles Page: Requester 20-19
23–1 Log-Level Activation in the Default Log Configuration File 23-21
24–1 Audit to Database Architecture 24-4
24–2 Common Settings: Auditing Configuration 24-13
25–1 Server Processes Overview Page 25-3

25–2 Session Operations Monitoring Page 25-3
25–3 Server Operations Monitoring Page 25-4
25–4 OAM Agents Monitoring Page 25-4
25–5 OAM Agent Monitoring Characteristics 25-5
25–6 Detached OAM 10g Agent Connection Table 25-6
25–7 Detached OAM 10g Agent Operations Overview Table 25-6
25–8 Detached OAM 10g Agent Operations Detail Table 25-6
25–9 Detached OAM 10g Agent Information Table 25-6
25–10 OSSO 10g Agent Monitoring Page with Operation Details 25-7
25–11 OSSO 10g Agent Monitoring Process Overview Table Detached 25-7
25–12 OSSO 10g Agent Information Table Detached 25-7
26–1 Fusion Middleware Control (AS-Control) Deployment Architecture 26-2
26–2 Fusion Middleware Control Login Page with Help Topics 26-3
26–3 OAM Farm Page in Fusion Middleware Control 26-4
26–4 Farm Navigation Tree in Fusion Middleware Control 26-5
26–5 Node Information Page in Fusion Middleware Control 26-6
26–6 Application Deployment Summary for the Selected Internal Application 26-6
26–7 Application Deployment Menu 26-7
26–8 WebLogic Server Domain Summary with Context Menu Exposed 26-7
26–9 Oracle Access Manager Cluster Page 26-10
26–10 Key Metrics for Oracle Access Manager Server Pages 26-10
26–11 Aggregated Access Manager Component Metrics for the Cluster 26-13
26–12 Access Manager Component Metrics for a Single OAM Server Instance 26-14
26–13 Aggregated STS Component Metrics for the Cluster 26-15
26–14 STS Component Metrics for an Individual OAM Server Instance 26-15
26–15 Performance Summary Command 26-16
26–16 Performance Summary Page with Metric Palette 26-17
26–17 Oracle Access Manager Log Levels on the Log Configuration Tab 26-22
26–18 Log Levels for Oracle Security Token Service 26-23
26–19 Log Files Configuration Page 26-26
26–20 Typical Log Messages Page in Fusion Middleware Control 26-30
26–21 System MBean Browser and Attributes Tab 26-36
26–22 Routing Topology with Context Menu 26-39
A–1 Pre-Upgrade OSSO 10g Topology A-3
A–2 Pre-Upgrade Sample OSSO 10g with Front-End Proxy A-3
A–3 Post-Upgrade OSSO 10g Topology A-4
A–4 mod_wl Replaces mod_oc4j on the Proxy Server A-5
A–5 Typical Topology Without Proxy Server A-5
A–6 Co-existence Processing A-6
A–7 Co-existence and OSSO 10g Authentication A-7
A–8 OSSO Agent Configuration Named for One Application A-9
A–9 OSSO Agent Configuration Named for the Second Application A-10
A–10 OSSO Agent Configuration Named for the Third Application A-10
A–11 Host Identifier for migratedSSOPartners A-10
A–12 Resources in the migratedSSOPartners Application Domain A-11
A–13 Authentication Policy for the Application Domain migratedSSOPartners A-11
B–1 Source and Target processing B-8
B–2 Dependency Tree for Each Application Domain B-8
E–1 Communication Channels for OAM Servers and Webgates E-2
G–1 IPv6 with OAM 11g and Challenge Redirect G-3

List of Tables

1–1 Deployment Types ... 1-3
1–2 Enhancements in Oracle Access Manager 11g ... 1-6
1–3 OAM 10g Functionality Not Available with Oracle Access Manager 11g ... 1-6
1–4 Comparison: OAM 11g versus OAM 10g versus OSSO 10g ... 1-8
1–5 Oracle Security Token Service Terms ... 1-11
1–6 Oracle Security Token Service 11g Infrastructure ... 1-15
1–7 Integrated Oracle Web Services Manager ... 1-17
2–1 OAM 11g Co-existence Summary ... 2-14
3–1 Role Mapping from an LDAP Group to Administrator ... 3-2
3–2 Welcome Page and Shortcuts ... 3-6
3–3 Function Tabs and Descriptions ... 3-7
3–4 Command Buttons in the Tool Bar ... 3-9
3–5 View Menu Command Descriptions ... 3-10
3–6 System Configuration, Actions Menu, Command Descriptions ... 3-11
3–7 Controls for Open Pages ... 3-12
3–8 Page Elements and Descriptions ... 3-12
3–9 Selection Tasks and Controls ... 3-13
3–10 Policy Configuration Subtabs ... 3-16
3–11 Policy Configuration Search Controls ... 3-18
3–12 Common System Configuration Search Controls ... 3-20
4–1 Common Configuration Nodes in Navigation Tree ... 4-2
4–2 Common Settings ... 4-4
4–3 Common Coherence Settings ... 4-5
5–1 Oracle Access Manager 11g, 10g, and OSSO Key Comparison ... 5-6
5–2 User Identity Store Elements ... 5-8
6–1 Summary: Server-side Differences with OAM 11g versus OAM 1g versus OSSO 10g... 6-2
6–2 OAM Server Instance Settings ... 6-5
6–3 OAM Proxy Settings for an Individual OAM Server ... 6-6
6–4 Default Coherence Settings for Individual OAM Servers ... 6-8
7–1 Common Session Settings ... 7-7
7–2 Session Management Controls and the Results Table ... 7-8
8–1 Access Manager Settings ... 8-2
8–2 Access Manager Settings: Load Balancer ... 8-2
8–3 External Error Codes, Trigger Conditions, and Recommended Messages ... 8-3
8–4 Access Manager Settings: SSO ... 8-5
8–5 Summary: Simple and Cert Mode ... 8-6
8–6 Server Common OAM Proxy Secure Communication Settings ... 8-7
8–7 Policy Evaluation Caches ... 8-9
8–8 Kerberos Authentication Module Definition ... 8-11
8–9 LDAP Authentication Module Definition ... 8-11
8–10 X509 Authentication Module Definition ... 8-13
8–11 Add New Step Entries, Steps Results Table, and Details Section ... 8-16
8–12 Steps Orchestration Subtab ... 8-18
9–1 Agents for OAM 11g ... 9-2
9–2 Comparing Agent Types and Differences ... 9-3
9–3 Comparing IAMSuiteAgent and 11g and 10g Webgates ... 9-7
9–4 Create Pages for OAM 10g and 11g Webgates ... 9-12
9–5 Expanded OAM 11g and 10g Webgate Elements and Defaults ... 9-16
9–6 User-Defined Webgate Parameters ... 9-21
9–7 OAM Agent Search Controls ... 9-26
9–8 Webgate Caches ... 9-31
9–9 Create OSSO Agent Page Elements ... 9-36
9–10 Expanded OSSO Agent Elements ... 9-37

10–1 Remote Registration Request Files 10-7
10–2 Remote Registration Sample Commands 10-7
10–3 Results of Remote Registration 10-9
10–4 Elements Common to Remote Registration Requests 10-11
10–5 OSSO-Specific Elements in a Remote Registration Request 10-12
10–6 Elements Common to Full Remote Registration Requests 10-13
10–7 Variables Required for Remote Registration 10-23
10–8 Remote Agent and Policy Updates 10-29
10–9 Remote Application Domain Management Modes 10-33
10–10 <rregApplicationDomain> Remote Management Template Elements 10-38
11–1 Comparing OAM 11g Policy Model with OAM 10g 11-2
11–2 Host Identifiers Examples 11-5
11–3 OAM 11g SSO versus OSSO 10g Component Summary 11-12
11–4 SSO Cookies 11-13
12–1 Resource Type Definition 12-4
12–2 Host Identifier Definition 12-12
12–3 Authentication Scheme Definition 12-16
12–4 Pre-configured Authentication Schemes 12-18
12–5 Challenge Parameters in Pre-configured Schemes 12-24
12–6 Challenge Parameters for Encrypted Cookies 12-30
13–1 Resource Definition Elements 13-12
13–2 HTTP Resources Sample URL Values 13-14
13–3 Resource URLs for.jsp 13-17
13–4 Resource Evaluation Outcomes 13-17
13–5 Search Elements for a Resource in an Application Domain 13-20
13–6 Authentication Policy Elements and Descriptions 13-23
13–7 Authorization Policy Elements and Descriptions 13-28
13–8 Response Elements 13-33
13–9 Namespace Request Variables for Single Sign-On 13-34
13–10 Namespace Session Variables for Single Sign-On 13-34
13–11 Namespace User Variables 13-34
13–12 Simple Responses and Descriptions 13-35
13–13 Complex Responses 13-36
13–14 Authorization Policy General Details 13-41
13–15 Add Constraint Window Elements 13-42
13–16 Identity Class Constraint Details 13-44
13–17 Temporal Constraint Class Details 13-46
14–1 User Interactions Using Tester Console Mode versus Command Line Mode Operations 14-7
14–2 Access Tester Supported System Properties 14-9
14–3 Access Tester Console Panels 14-13
14–4 Command Buttons in Access Tester Panels 14-13
14–5 Additional Access Tester Buttons 14-13
14–6 Access Tester Menus 14-14
14–7 Connection Panel Information 14-16
14–8 Protected Resource URI Panel Fields and Controls 14-19
14–9 Access Tester User Identity Panel Fields and Controls 14-21
14–10 Access Tester Capture Request Options 14-25
14–11 Generate Script Command 14-27
14–12 Test Script Control Parameters 14-28
14–13 Run Test Script Commands 14-29
14–14 Mismatched Results Reasons in the Statistics Document 14-32
15–1 Centralized Logout Circumstances 15-2
15–2 Logout Elements in OAM 11g Webgate Registration 15-4
15–3 Sample end_url Parameter Specifications 15-10

17–1 Policies Transport Security when Message-level Security Not Required 17-7
17–2 Security Token Service Settings 17-11
17–3 Oracle Security Token Service Configuration Management Operations..................... 17-23
17–4 Oracle Security Token Service-specific Run-time Events 17-25
18–1 OSTS Public Keys Used at Run Time..................... 18-2
18–2 Keystores for Oracle Access Manager with Oracle Security Token Service..................... 18-2
18–3 Keystore Mbeans..................... 18-3
18–4 Partner Keys for WS-Trust Communications 18-8
18–5 Conditions for Oracle Security Token Service Certificate Validation 18-9
18–6 Successful Certificate Validation Requirements..................... 18-9
19–1 Template Search Controls..................... 19-4
19–2 Issuance Template Requirements..................... 19-7
19–3 Issuance Template: General Details 19-8
19–4 Issuance Properties: Username Token Type..................... 19-8
19–5 Issuance Properties: SAML Token Types..................... 19-10
19–6 Security Details: SAML Tokens 19-11
19–7 Issuance Template: Attribute Mapping, SAML Token 19-12
19–8 Validation Template Protocols..................... 19-16
19–9 New Validation Template: General Details 19-17
19–10 New Validation Template: Authentication Details..................... 19-19
19–11 New Validation Template: Token Mapping 19-22
19–12 Endpoints Page..................... 19-29
19–13 Constraints Tab: Token Issuance Policy 19-32
20–1 Elements for Oracle Security Token Service Partners 20-3
20–2 Profile: General..................... 20-8
20–3 Requester Profile: Token and Attributes 20-9
20–4 Relying Party Profile Requirements..................... 20-10
20–5 Token and Attributes Elements: Issuing Authority..................... 20-15
20–6 Issuing Authority Token Mapping Elements 20-16
22–1 Oracle Access Manager Server-Side Components 22-3
22–2 Oracle Access Manager Shared-Service Engine Components..................... 22-3
22–3 Oracle Access Manager Foundation APIs Components 22-3
22–4 Mapping of ODL to Java Levels 22-4
22–5 Oracle Security Token Service Logger..................... 22-8
23–1 Logging Levels 23-2
23–2 Log Configuration File Names for Components..................... 23-5
23–3 Log Writers 23-10
23–4 Global Parameters in the First Compound List..................... 23-11
23–5 Factors that Determine Whether Logging Is Active 23-16
23–6 Mandatory Log Configuration File Parameters 23-17
23–7 Log Data File Configuration Parameters..................... 23-18
23–8 ParamName Values You Can Configure for Per-Module Logging Threshold..................... 23-23
24–1 Oracle Access Manager Administrative Audit Events..................... 24-6
24–2 OAM Run-time Audit Events 24-8
24–3 Audit Configuration Elements..................... 24-13
25–1 OAM Proxy Metrics..................... 25-8
25–2 OAM Proxy Tuning Parameters 25-9
26–1 Farm Page Sections 26-5
26–2 Resulting Pages for Selected Nodes and Targets 26-9
26–3 Summary of Performance Overviews in Fusion Middleware Control..................... 26-11
26–4 Access Manager Component Metrics 26-14
26–5 STS Component-Specific Metrics..................... 26-16
26–6 Status and Controls on Performance Summary Pages..................... 26-17
26–7 OAM Log Availability and Functions in Fusion Middleware Control..................... 26-21
26–8 Log Levels Tab on Log Configuration Page..................... 26-23

26–9 Log Files Elements 26-27
26–10 OAM Log Message Search Controls in Fusion Middleware Control........ 26-30
26–11 System MBean Browser 26-35
26–12 26-35
26–13 System MBean Browser 26-37
26–14 Farm Topology 26-39
27–1 Installation Comparison with OAM 10g Webgates 27-3
27–2 Preparing for 10g Webgate Installation with OAM 11g........ 27-7
29–1 IIS 7 Webgate Windows Server 2008........ 29-6
A–1 Partner Applications Protected by OSSO 10g........ A-8
B–1 Deployment Types........ B-2
B–2 Differences when Transferring Data to New versus Existing Target Environments B-3
B–3 Full Replication B-4
B–4 Golden Template........ B-5
B–5 Delta-Replication........ B-5
B–6 Application Re-association B-6
B–7 Export Commands B-7
B–8 Import Commands........ B-7
C–1 addOAMSSOProvider Command-line Arguments........ C-3
D–1 Languages for Localized Messages in Oracle Access Manager D-2
E–1 importcert Command Syntax........ E-4
F–1 Operational Modes for WLST commands for OAM........ F-2
F–2 WLST Oracle Access Manager Commands........ F-2
F–3 WLST Commands Oracle Security Token Service F-5
H–1 Login Page Parameters Submitted to the Page by the Single Sign-On Server........ H-3
H–2 Login Page Parameters Submitted by the Page to the Single Sign-On Server........ H-4
H–3 Change Password Parameters Submitted to the Page........ H-5
H–4 Change Password Page Parameters Submitted by the Page H-5
H–5 Parameters Submitted to the Single Sign-Off Page........ H-6
H–6 Parameters Submitted to the External Application Login Page H-6
H–7 Parameters the External Application Login Page Submits to the Application........ H-7
H–8 Login Page Error Codes H-8
H–9 Post-Login Messages H-9
H–10 Change Password Page Error Codes H-10
H–11 External Application Login Page Error Codes H-10
H–12 External Application Login H-14
H–13 Authentication Method........ H-15
H–14 Additional Fields........ H-15

List of Examples

10–1 OSSORequest.xml 10-10
10–2 Sample Simplified Request: OAMRequest_short.xml 10-11
14–1 Connection Configuration File........ 14-32
14–2 Generated Input Test Script........ 14-33
14–3 Output File Generated During a Test Run 14-35
14–4 Sample Statistics Document 14-36
14–5 Execution Log 14-38
15–1 logout.html Script 15-8
16–1 Sample exchange: Request Security Token Sent By the Client 16-21
16–2 Request Security Token Response sent by the OSTS Server........ 16-22
22–1 Configuring Oracle Access Manager Loggers and Log Handlers........ 22-4
23–1 The Default Log Configuration File with Comments 23-6
23–2 Simple Lists with Global Settings (First Compound List in oblog_config_wg.xml).... 23-12
23–3 FILTER_LIST Masks Sensitive Attributes in Log Files........ 23-15
23–4 Valid Name/Value List........ 23-15
23–5 Another Valid Name/Value List........ 23-16
23–6 Opening tag for a Name/Value List 23-16
23–7 Opening tag for a Name/Value List 23-16
23–8 A Default Log Configuration File Without Embedded Comments 23-19
27–1 Updates for the 10g Webgate in mod_wl_ohs.conf 27-18
C–1 Sample SSO Configuration for OAM 11g........ C-4

Preface

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

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
- Security token concepts
- Integration of the Policy store, Identity store, and familiarity with Oracle Identity Management and OIS might be required

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, 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, JAWS 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 Patching Guide—Describes the various tools and processes that are required to patch your Oracle Fusion Middleware software to the latest version.
- 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 Administrator's Guide for Oracle Access Manager with Oracle Security Token Service—Explains how to manage configuration and policies for Oracle Access Manager with Oracle Security Token Service.
- Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service—Explains how to write custom applications and plug-ins to functions programmatically, to create custom Access Clients that protect non-Web-based resources.
- 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".
- Oracle Fusion Middleware Security and Administrator's Guide for Web Services—Describes how to administer and secure Web services.

Conventions

The following text conventions are used in this document:

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

What's New

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

New Features in Patch Set 1

Patch set 1 provides new functions and enhancements, as introduced in the following topics:

- Administration Console Enhancements
- Authentication Plug-in Enhancements
- Query String-based HTTP Resource Definitions
- Excluded Resource List
- Session Search and Session Management Enhancements
- Multiple Identity Store Support
- OAM Tester Enhancements
- Oracle Secure Token Service
- Oracle Access Manager SDK, Custom Authentication Plug-ins, Custom Tokens
- Remote Registration Enhancements
- Webgate Enhancements

Administration Console Enhancements

The System Configuration tab has been divided into three new sections:

- Common Configuration
- Access Manager Settings
- Security Token Service Settings

See Also:

- Chapter 3, "Getting Started with Common Administration and Navigation"
- Chapter 4, "Managing Services, Certificate Validation, and Common Settings"
- Chapter 8, "Configuring Access Manager Settings"
- Chapter 17, "Managing Oracle Security Token Service Settings and Set Up"

Authentication Plug-in Enhancements

Authentication is governed by specific authenticating schemes that rely on one or more plug-ins that test the credentials provided by a user when she tries to access a resource. The plug-ins can be taken from a standard set provided with OAM Server installation, or custom plug-ins created by your own Java developers.

See Also:

- "Managing Authentication Modules" on page 8-10
- Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service

Query String-based HTTP Resource Definitions

The Policy Model supports Query String-based HTTP Resource Definitions within Access Policies.

See Also:

- Table 13–1, " Resource Definition Elements" on page 13-12

Excluded Resource List

Oracle Access Manager provides support to help you keep certain resources public (not protected by the OAM Agent).

See Also:

- Table 10–6, " Elements Common to Full Remote Registration Requests" on page 10-13
- Table 13–1, " Resource Definition Elements" on page 13-12

Session Search and Session Management Enhancements

User-session lifecycle settings are part of the Common Settings shared by all OAM Servers. These have moved to the Common Settings page

See Also: "About Common Session Lifecycle Setting Page" on page 7-6

Authenticated clients can manage Session operations.

See Also: Table 7–1, " Common Session Settings" on page 7-7 for details on the Allow Management Operations parameter.

Database Persistence for Active Sessions: You can persist active sessions to the configured database session store, in addition to the local and distributed caches. Sessions are retained even if all managed servers die off.

> **See Also:** Table 7–1, " Common Session Settings" on page 7-7 for details about the Database Persistence for Active Sessions parameter.

Oracle Access Manager provides enhanced Session Search controls that enable you to create a query based on filter conditions.

> **See Also:** Table 7–2, " Session Management Controls and the Results Table" on page 7-8

Multiple Identity Store Support

Multiple user identity stores are supported:

- Only the User Identity Store designated as the System Store is used to authenticate Administrators signing in to use the Oracle Access Manager Console, remote registration, and custom administrative commands in WLST.
- Users attempting to access an OAM-protected resource can be authenticated against any store, not necessarily the only one marked as Default User Identity Store.
- Oracle Security Token Service uses only the Default User Identity Store. When adding User constraints to a Token Issuance Policy, for instance, the identity store from which the users are to be chosen must be Default User Identity Store.

> **See Also:** "About User Identity Stores" on page 5-2

OAM Tester Enhancements

CERT mode connections are supported in this release which requires having stores with a client certificate and a root certificate. Both stores can be generated using the IMPORTCERT tool.

The OAM Tester can also run concurrent tests in multi-threaded mode, which can be used to stress test the policy server. The tests are run in command-line mode only and the input configuration file specifies the number of threads and the number of iterations each thread should execute. Each thread then open a dedicated connection to the policy server and run the specified input script the specified number of iterations.

> **See Also:** "Validating Connectivity and Policies Using the Access Tester"

Oracle Secure Token Service

Oracle Security Token Service is deployed with Oracle Access Manager and can be activated as a service.

Oracle Security Token Service provides a foundation to the current security infrastructure to facilitate a consistent and streamlined model for token acquisition, renewal, and cancellation that is protocol and security infrastructure agnostic.

Oracle Security Token Service is a Web Service (WS) Trust-based token service that allows for policy-driven trust brokering and secure identity propagation and token exchange between Web Services. Oracle Security Token Service can be deployed as a Security and Identity Service needed to simplify the integration of distributed or federated Web services within an enterprise and its service providers.

See Also:

- Chapter 16, "Oracle Security Token Service Implementation Scenarios"
- Chapter 17, "Managing Oracle Security Token Service Settings and Set Up"
- Chapter 18, "Managing Oracle Security Token Service Certificates and Keys"
- Chapter 19, "Managing Templates, Endpoints, and Policies"
- Chapter 20, "Managing Token Service Partners and Partner Profiles"
- Chapter 21, "Troubleshooting Oracle Security Token Services"

Oracle Access Manager SDK, Custom Authentication Plug-ins, Custom Tokens

The Oracle Access Manager 11g Access SDK is a platform independent package that Oracle has certified on a variety of enterprise platforms (using both 32-bit and 64-bit modes) and hardware combinations. It is provided on JDK versions that are supported across Oracle Fusion Middleware applications.

Oracle Access Manager 11g provides authentication plug-in interfaces and SDK tooling to build customized authentication modules (plug-ins) to bridge the out-of-the-box features with individual requirements.

When Oracle Security Token Service does not support the token that you want to validate or issue out-of-the-box, you can write your own validation and issuance module classes.

See Also:

- *Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service*
- *Oracle Security Token Service Java API Reference*
- *Oracle Access Manager Access SDK Java API Reference*
- *Oracle Access Manager Extensibility Java API Reference*

Remote Registration Enhancements

Remote registration tooling permits Administrators and application deployers to remotely register an application for protection by Oracle Access Manager. Enhancements to the remote registration tool, oamreg, have been made to mirror enhancements to Webgate registration. Certain changes have been made to the templates used to perform remote registration. New modes are available to manage Agents remotely. A new option is available to pipe in passwords.

See Also:

- Table 10–1, " Remote Registration Request Files" on page 10-7
- Table 10–2, " Remote Registration Sample Commands" on page 10-7 provide details of the -noprompt option
- "About Remote Registration Request Files" on page 10-9
- Table 10–6, " Elements Common to Full Remote Registration Requests" on page 10-13 provides details about the `virtualhost` and `hostportVariations` parameters; `excludedresourcesList` parameter; `allowManagementOperations` parameter; `cachePragmaHeader` and `cacheControlHeader` parameters; `ipValidationExceptions` parameter; and more.
- "Introducing Remote Management Modes" on page 10-29
- "Managing Agents Remotely" on page 10-39
- "Creating or Updating an Application Domain Without an Agent" on page 10-41

Webgate Enhancements

Webgate caches resources from an exception list that should not be checked for authorization and should just be allowed to pass through.

See Also: Table 9–5, " Expanded OAM 11g and 10g Webgate Elements and Defaults" on page 9-16

You can implement certain user-defined parameters in the Webgate registration page.

See Also: "About User-Defined Webgate Parameters" on page 9-21

Only privileged agents can invoke session management operations. The Agent Privilege function enables the provisioning of session operations per agent.

See Also: Table 9–5, " Expanded OAM 11g and 10g Webgate Elements and Defaults" on page 9-16

You can configure single sign-on between Webgate and an access client that does not have the client IP address at authentication.

See Also: "About IP Address Validation for Webgates" on page 9-21

You can configure Webgate only settings to control the browser’s cache.

See Also: " Expanded OAM 11g and 10g Webgate Elements and Defaults" on page 9-16 for details about:

- Cache Pragma Header
- Cache Control Header

During Agent searches, if you do not know the exact name you can use a wild card (*) in the search string.

See Also: "Searching for an OAM Agent Registration" on page 9-25

Release 11g Release 1 (11.1.1)

See Chapter 2, "Introduction to This Book" for a full introduction, and the following topic for product and component name changes.

Product and Component Name Changes

The original product name, Oblix NetPoint, was changed to Oracle Access Manager and v7.x releases were available from Oracle as part of Oracle Application Server 10g Release 2 (10.1.2). Oracle Access Manager 10.1.4 provided some product and component name changes, with more in Oracle Access Manager 11g, 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 Webgate Access Client N/A N/A
Agents	Webgate AccessGate	Webgate (also OAM Agent) Access Client (also 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	Administrator N/A N/A N/A
Agent and partner application registration	N/A	Oracle Access Manager Console Remote registration tool provides automated Agent registration and application domain creation with default security policies
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	oamreg Remote registration of OAM Agents (10g and 11g Webgates and Access Clients), application domain, default policies for SSO.
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

	OAM 10g	OAM 11g
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))

Part I

Introduction to Oracle Access Manager with Oracle Security Token Service

This part of the book provides an introduction to Oracle Access Manager with Oracle Security Token Service.

Part I contains the following chapters

- Chapter 1, "Oracle Product Introduction"
- Chapter 2, "Introduction to This Book"

1 Oracle Product Introduction

This chapter provides a high-level overview of Oracle Access Manager 11g and Oracle Security Token Service with links to more information. This chapter contains the following sections:

- Introduction to Oracle Access Manager
- Introduction to Oracle Security Token Service

1.1 Introduction to Oracle Access Manager

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.

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 Oracle Access Manager 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. Custom access clients, created using the Access Manager SDK, can be used with non-Web applications. Unless explicitly stated, information in this book applies equally to both.

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.

The remainder of this section provides the following topics:

- Introduction to Oracle Access Manager Architecture

- Introduction to Oracle Access Manager Deployment Types and Installation
- Comparing Oracle Access Manager 11g, 10g, and OracleAS SSO 10g

There are several important differences between Oracle Access Manager 11g and Oracle Access Manager 10g,, and OSSO 10g, as described in "Comparing Oracle Access Manager 11g, 10g, and OracleAS SSO 10g" on page 1-5.

1.1.1 Introduction to Oracle Access Manager 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 the following earlier technologies:

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

Figure 1–1 illustrates the primary Oracle Access Manager 11g components and services. The Protocol Compatibility Framework interfaces with OAM Webgates, mod_osso agents, and custom Access Clients created using the Access Manager Software Developer Kit (SDK).

Figure 1–1 Oracle Access Manager 11g Components and Services

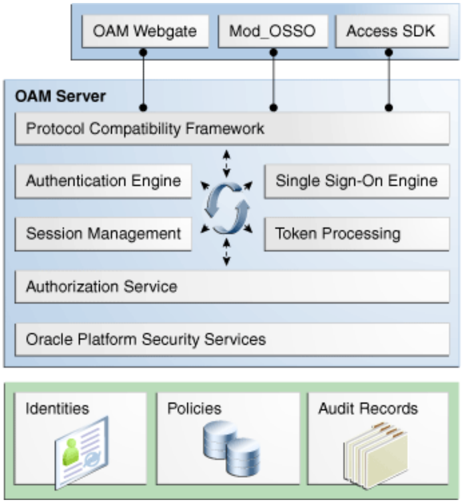
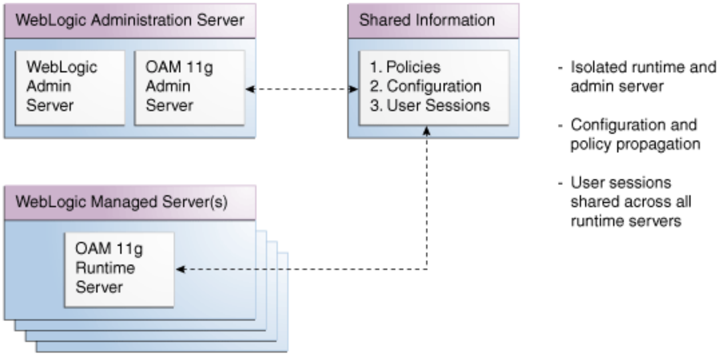


Figure 1–2 illustrates the distribution of Oracle Access Manager components.

Figure 1–2 Oracle Access Manager 11g Component Distribution


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

The Oracle Access Manager Console 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

1.1.2 Introduction to Oracle Access Manager Deployment Types and Installation

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

- About Deployment Types and OAM
- About Oracle Access Management Post-Installation Tasks

1.1.2.1 About Deployment Types and OAM

Table 1–1 describes the types of deployments you might have within your enterprise, even though these might be named differently in your enterprise.

Table 1–1 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

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
- A WebLogic Managed Server for Oracle Access Management
- 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 keystore is set up to be used for certificates for Simple or Certificate-based communication between OAM Servers and Webgates during authorization. The keystore bootstrap also occurs on the initial AdminServer startup after running the Configuration Wizard.

1.1.2.2 About Oracle Access Management Post-Installation Tasks

During initial deployment, the WebLogic Administrator userID and password are set for use when signing in to both the Oracle Access Manager Console and WebLogic Server Administration Console. A different administrator can be assigned for Oracle Access Management (Oracle Access Manager and Oracle Security Token Service, as described in "Introduction to Administrators" on page 3-2.

Oracle Access Management administrators can log in and use the Oracle Access Manager 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
- Oracle Security Token Service Settings and configuration as introduced in "Introduction to Oracle Security Token Service" on page 1-10.

1.1.2.3 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:**
>
> - Patching Oracle Identity and Access Management 11.1.1.3.0 to 11.1.1.5.0 in the Oracle Fusion Middleware Patching Guide
> - 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/Access Clients) 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 A, "Co-existence Overview: OAM 11g and OSSO 10g"

1.1.3 Comparing Oracle Access Manager 11g, 10g, and OracleAS SSO 10g

This section provides the following topics:

- Enhancements in Oracle Access Manager 11g
- Oracle Access Manager 10g Functionality Not Available with 11g
- Comparing Oracle Access Manager 11g, 10g, and OracleAS SSO 10g

1.1.3.1 Enhancements in Oracle Access Manager 11g

Oracle Access Manager 11g includes several important enhancements that were not available with Oracle Access Manager 10g. Enhancements are listed in Table 1–2.

> **See Also:** "What's New" on page xli

Table 1–2 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 Access Clients) and OracleAS 10g SSO agents (mod_osso) for SSO s provided. See Chapter 9, Chapter 10. ■ Upgrading and Co-existence: Utilities to upgrade 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 A.
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 IV. Per-agent-based shared secret key increases security and performance by moving cookie encryption and description to the agent. See Chapter 9 and Chapter 11.
Embedded LDAP for user and group information is described in Chapter 5. Integration with Oracle Entitlement Server MicroSM to enable database storage of policies. See Chapter 5.
■ 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 14.
Session Management functions are provided, as described in Chapter 7: ■ Webgate maximum user session timeout is now supported by Webgate through the host cookie See Table 1–4, " 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 24.
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*.
Extensibility framework required for building custom authentication plug-ins. For more information, see Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service. Complex policy constructs (AND, OR semantics for multiple rules) Impersonation support

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

1.1.3.2 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 1–3 provides an overview.

Table 1–3 OAM 10g Functionality Not Available with Oracle Access Manager 11g

Unavailable or Unsupported Functions
Extensibility framework required for building custom authorization plug-ins.
Application-domain-level delegated administration
LDAP filter-based authorization and response calculations

Table 1–3 (Cont.) OAM 10g Functionality Not Available with Oracle Access Manager 11g

Unavailable or Unsupported Functions
Authorization for mod_osso-protected resources
Identity Server, WebPass, Identity System Console, User Manager, Group Manager, Organization Manager. Replaced by Oracle Fusion Middleware Identity Manager.

1.1.3.3 Comparing Oracle Access Manager 11g, 10g, and OracleAS SSO 10g

This topic 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 Access Client.

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

Table 1–4 provides a comparison of Oracle Access Manager 11g, OAM 10g, and OracleAS SSO 10g. For a list of names that have changed with Oracle Access Manager 11g, see "Product and Component Name Changes" on page xlvi.

Table 1–4 Comparison: OAM 11g versus OAM 10g versus OSSO 10g

	OAM 11g	OAM 10g	OSSO 10g
Architecture Components	■ Agents: Webgate, Access Client, mod_osso, and IAMSuiteAgent ■ OAM Server ■ Oracle Access Manager Console (installed on WebLogic Administration Server) Note: Eight Administrator languages are supported.	■ Resource Webgate (RWG) ■ Authentication Webgate (AWG) ■ AccessGate ■ 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 inter-operability 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 1–4 (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 11g 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 1–4 (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 15.
		■ 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

1.2 Introduction to Oracle Security Token Service

Oracle Security Token Service is deployed with Oracle Access Manager and must be activated as a service.

See Also:

- Patching Oracle Identity and Access Management 11.1.1.3.0 to 11.1.1.5.0 in the Oracle Fusion Middleware Patching Guide
- "Enabling and Disabling Oracle Security Token Service" on page 17-8
- Part V, "Oracle Security Token Service"

Oracle Security Token Service provides the foundation to the current security infrastructure to facilitate a consistent and streamlined model for token acquisition, renewal, and cancellation that is protocol and security infrastructure agnostic.

Oracle Security Token Service is a Web Service (WS) Trust-based token service that allows for policy-driven trust brokering and secure identity propagation and token exchange between Web Services. Oracle Security Token Service can be deployed as a

Security and Identity Service needed to simplify the integration of distributed or federated Web services within an enterprise and its service providers.

Oracle Security Token Service is primarily based on the OASIS WS-Trust protocol. However, Oracle Security Token Service delegates the processing of other WS-* protocols present in the SOAP message.

Oracle Security Token Service brokers trust between a Web Service Consumer (WSC) and a Web Service Provider (WSP) and provides security token lifecycle management services to providers and consumers. Oracle Security Token Service can help simplify the effort needed to bridge access to various systems using a standardized set of interfaces.

Oracle Security Token Service (Oracle STS) augments Oracle Identity Federation (OIF), which facilitates Federated Single Sign-on (SSO) and Single Logout (SLO) for users coming through a Web browser to access resources across different security domains or across administrative boundaries through various federation protocols like SAML, WS-Federation, Liberty, or OpenID.

Security tokens contain claims or statements that are used to assert trust. To secure communication between a Web service client and a Web service, the two parties must exchange security credentials. These credentials can be obtained from a trusted Security Token Service (STS). To provide interoperable security tokens, the STS must be trusted by both the Web service client and the Web service.

Modern IT environments have numerous types of security tokens, most of them based on browser cookies to facilitate SSO and application session management for Web applications. Additional tokens include Kerberos (primarily for Windows Native Authentication), Security Assertion Markup Language (SAML) assertions, and even digital certificates.

For more information, see the following topics:

- Oracle Security Token Service Key Terms and Concepts
- About Oracle Security Token Service with Oracle Access Manager
- About Integrated Oracle Web Services Manager
- About Oracle Security Token Service Architecture
- About Oracle Security Token Service Deployments
- About Oracle Security Token Service Administration

1.2.1 Oracle Security Token Service Key Terms and Concepts

Table 1–5 identifies common Oracle Security Token Service terminology.

Table 1–5 Oracle Security Token Service Terms

Term	Description
Security Token	A security mechanism that protects messages using a token issued by a trusted Secure Token Service for message integrity and confidentiality protection. The issued tokens contain a key, which is encrypted for the server and which is used for deriving new keys for signing and encrypting. Service providers and consumers in potentially different managed environments can use a single Security Token Service to establish a chain of trust. The service does not trust the client directly, but instead trusts tokens issued by a designated Security Token Service. The Security Token Service is taking on the role of a second service with which the client must securely authenticate.

Table 1–5 (Cont.) Oracle Security Token Service Terms

Term	Description
Security Token Service	A trusted third party in an explicit trust relationship with the server (and a trust relationship with the client). Oracle Security Token Service is one example.
Secure Token Service	A shared Web service that provides a standards-based consolidated mechanism of trust brokerage between different identity domains and infrastructure tiers. The service implements the protocol defined in the WS-Trust specification by making assertions based on evidence that it trusts, to whoever trusts it (or to specific recipients). This protocol defines message formats and message exchange patterns for issuing, renewing, canceling, and validating security tokens. To communicate trust, a service requires something to prove knowledge of a security token or set of security tokens. An XML Signature binds the sender's identity (or "signing entity") to an XML document, for example. The document is signed using the sender's private key, the signature is verified using the sender's public key.
Request Security Token (RST)	Request for a security token.
Request Security Token Response (RSTR)	Response generated by Oracle Security Token Service in response to the Request for Security Tokens with claims for the requested user.
On Behalf Of (OBO)	An OBO Request Security Token (RST) is used when only the identity of the original client is important. An OBO RST indicates that the requestor wants a token containing claims about only one entity: ■ the external entity represented by the token in the OnBehalfOf element.
ActAs	An ActAs RST requires composite delegation. The final recipient of the issued token can inspect the entire delegation chain (not just the client). An ActAs RST indicates that the requestor wants a token that contains claims about distinct entities: ■ the requestor ■ an external entity represented by the token in the ActAs element
Token Exchange	The exchange of one security token for another. The requestor (in order to invoke a web service) requires a particular token. It uses Oracle Security Token Service to exchange the incoming token with a token required by the service.
WS-Security	Web Services Security (WS-Security) specifies SOAP security extensions that provide confidentiality using XML Encryption and data integrity using XML Signature. The most prevalent security tokens used with WS-Security are Username, X.509 Certificates, SAML assertions, and Kerberos tickets (all supported by Oracle Web Service Manager). WS-Security also includes profiles that specify how to insert different types of binary and XML security tokens in WS-Security headers for authentication and authorization purposes: WS-* specifications often depend on each other. For example, WS-Policy is used in conjunction with WS-Security. WS-* specifications also leverage non-WS-* specifications; for example, WS-Security uses XML Encryption and XML Signature. For WS-Security, only SAML assertions are used. The protocols and bindings are provided by the WS-Security framework. **Note**: WS-Security, WS-Trust, WS-Policy have been transferred over to standards bodies such as the Organization for the Advancement of Structured Information Standards (OASIS) or the World Wide Web Consortium (W3C).

Table 1–5 (Cont.) Oracle Security Token Service Terms

Term	Description
WS-Trust	Web Services Trust Language (WS-Trust) is a specification that uses the secure messaging mechanisms of WS-Security to facilitate trust relationships. WS-Trust defines a request and response protocol that enables applications to construct trusted SOAP message exchanges. Trust is represented through the exchange and brokering of security tokens. In a message exchange using WS-Security only, it is assumed that both parties involved in the exchange have a prior agreement on which type of security tokens they must use for sharing security information. However, there are cases where these parties do not have such an agreement, as a result trust must be established before exchanging messages. Trust between two parties exchanging SOAP / WS-Security-based messages is established by implementing the WS-Trust specification.
WS-Policy	Web Services Policy (WS-Policy). Together with WS-Security, WS-Policy is another key industry standard for Oracle Fusion Middleware security. WS-Policy is used in conjunction with WS-Security. A web service provider may define conditions (or policies) under which a service is to be provided. The WS-Policy framework enables one to specify policy information that can be processed by web services applications, such as Oracle Web Services Manager. A policy is expressed as one or more policy assertions representing a web service’s capabilities or requirements. For example, a policy assertion may stipulate that a request to a web service be encrypted. Likewise, a policy assertion can define the maximum message size that a web service can accept.
Certificates	The certificates used by Oracle Security Token Service are self signed. The subject and the issuer field are identical. Out of the box, the OAM Server hosting Oracle Security Token Service is uniquely identified:
Keystore	Oracle Security Token Service key stores include: ■ System Keystore ■ Trust Keystore ■ Partner Keystore See Also: Chapter 18, "Managing Oracle Security Token Service Certificates and Keys"
User Name Token (UNT)	Identifies the requestor by their username, and optionally using a password (or shared secret, or password equivalent) to authenticate that identity. When using a username token, the user must be configured in the Default User Identity Store.,
X.509 Certificates	A signed data structure designed to send a public key to a receiving party. A certificate includes standard fields such as certificate ID, issuer's Distinguished Name (DN), validity period, owner's DN, owner's public key, and so on. Certificates are issued by certificate authorities (CA), for example Verisign. A CA verifies an entity's identity and grants a certificate, signing it with the CA's private key. The CA publishes its own certificate which includes its public key. Each network entity has a list of the certificates of the CAs it trusts. Before communicating with another entity, a given entity uses this list to verify that the signature of the other entity's certificate is from a trusted CA.

Table 1–5 (Cont.) Oracle Security Token Service Terms

Term	Description
Security Assertion Markup Language (SAML) SAML Assertion	An open framework for sharing security information on the Internet through XML documents. SAML provides: ■ Assertions that define authentication and authorization information. ■ Protocols to ask (SAML Request) and get (SAML Response) the assertions you need. ■ Bindings that define how SAML Protocols ride on industry-standard transport (HTTP for instance) and messaging frameworks (SOAP for instance). ■ Profiles that define how SAML Protocols and Bindings combine to support specific use cases. For WS-Security, only SAML assertions are used. However, the protocols and bindings are provided by the WS-Security framework. SAML assertions can include three types of statements: ■ Authentication statement: issued by an authentication authority upon successful authentication of a subject. It asserts that Subject S was authenticated by Means M at Time T. ■ Attribute statement: issued by an attribute authority, based on policies. It asserts that Subject S is associated with Attributes A, B, etc. with values a, b, and so on. ■ Authorization decision statement (deprecated in SAML 2.0, now supported by XACML): issued by an authorization authority which decides whether to grant the request by Subject S, for Action A (read, write, and so on.), to Resource R (e.g., a file, an application, a web service), given Evidence E.
Kerberos	A cross-platform authentication and single sign-on system. The Kerberos protocol provides mutual authentication between two entities relying on a shared secret (symmetric keys). Kerberos authentication requires a client, a server, and a trusted party to mediate between them called the Key Distribution Center (KDC). Also required: ■ A Principal: An identity for a user (a user is assigned a principal), or an identity for an application offering Kerberos services. ■ A Realm is a Kerberos server environment, which can be a domain name such as EXAMPLE.COM (by convention expressed in uppercase). Each Kerberos realm has at least one Web Services Security KDC. The Kerberos Token profile of WS-Security allows business partners to use Kerberos tokens in service-oriented architectures (SOAs).

1.2.2 About Oracle Security Token Service with Oracle Access Manager

Oracle Security Token Service is compliant and co-exists with Oracle Access Manager (using Oracle Access Manager as the primary authenticator for Web clients requesting tokens).

Oracle Security Token Service is installed with Oracle Access Manager 11g on Managed Servers. Each Managed Server must be registered with Oracle Access Manager to open communication channels. All Oracle Security Token Service system configuration is done using the Oracle Access Manager Console. Oracle Security Token Service inter-operates with third party security token servers.

Oracle Security Token Service uses Oracle Web Services Manager Agents. Webgate is used as an Agent for identity propagation. The Webgate must be registered with Oracle Access Manager 11g to open a communication channel.

Oracle Security Token Service leverages the common infrastructure for shared services and the Oracle Access Manager 11g administration model. In addition, Oracle Security Token Service is integrated with the Oracle Access Manager Console to provide a unified and consistent administration experience.

Oracle Security Token Service adopts the same frameworks, guidelines, and practices for diagnostics, monitoring, auditing, and high availability used by Oracle Access Manager 11g. For more information, see Part VI, "Common Logging, Auditing, Performance Monitoring".

Oracle Security Token Service processing:

- Integrates with STS Audit events
- Publishes, in the Oracle Access Manager Console and WLST scripts, available Oracle Security Token Service methods to manage partner data
- Performs validation operations specific to the Oracle Security Token Service use cases and configuration model

The Oracle Security Token Service 11g infrastructure is described in Table 1–6.

Table 1–6 Oracle Security Token Service 11g Infrastructure

Component	Description
Default Trust Keystore	Oracle Security Token Service private keys used for Signing/Encryption are stored in the common keystore used with Oracle Access Manager. Oracle Security Token Service and Oracle Access Manager use the common infrastructure certification validation module. Trusted Certificates and Certificate Revocation Lists (CRLs) used during certificate validation are stored in Trust Keystore and CRL ZIP file. The Oracle Security Token Service configuration stores the OCSP/CDP settings. The token security key pair is populated to Oracle Access Manager/Oracle Security Token Service keystore. Note: When the Oracle WSM Agent is used as the WS_Trust client in the OSTS deployment, Oracle strongly recommends that the Oracle WSM Agent keystore and the OSTS/OAM keystore always be different. Do not merge the two. Otherwise, OAM/OSTS keys could be available to any modules authorized by OPSS to access the keystore and OAM keys might be accessed. See Also: "About Oracle Security Token Service Keystores" on page 5-5.
Default User Identity Store	Oracle Security Token Service authenticates and maps users against the User Identity stores configured through the Common Configuration section of System Configuration in the Oracle Access Manager Console. Oracle Security Token Service maps the incoming token to user records and attributes in the default User Identity Store, which operates with both Oracle Access Manager and Oracle Security Token Service. See Also: "About Setting the Default Store and System Store" on page 5-12.

Table 1–6 (Cont.) Oracle Security Token Service 11g Infrastructure

Component	Description
Certificates	The certificates used by Oracle Security Token Service are self signed. The subject and the issuer field are identical. Out of the box, the OAM Server hosting Oracle Security Token Service is uniquely identified: ■ The keys and certificates used in Oracle Security Token Service are generated during installation. The subject and issuer fields are linked to the host name. This applies to the signing and encryption keys and certificates used by Oracle Security Token Service, as well as the keys/certificates used by the OWSM Agent protecting OSTS. The OWSM Agent is the certified WS-Trust client that can be used to communicate with OSTS. ■ The SAML Issuer settings are configured to refer to the host name of the local computer. This ensures that two servers are not identical in terms of cryptographic materials and identifiers. The trust granted to one server by third-party modules is not granted to the other server because the identifiers and cryptographic keys differ. There are no identical keys, no identical identifiers, and authorization policies are in denial mode.
Oracle Coherence	Oracle Security Token Service integrates with the Oracle Coherence module to store and share run time WS-Trust data across all the physical instances of Oracle Security Token Service. The UserNameToken Nonce are stored in the Coherence store. This implementation supports the following requirements, which might be specific to Oracle Security Token Service: ■ Cleanup of timed out records ■ Existence of the records limited to several minutes (< 30)

1.2.3 About Integrated Oracle Web Services Manager

In the 11g release, Oracle Web Services Manager (WSM) security and management has been integrated into the Oracle WebLogic Server along with Oracle WSM Agent functionality. Table 1–7 describes these components.

See Also: "About Oracle Access Manager Security Keys and the Embedded Java Keystore"

Table 1–7 Integrated Oracle Web Services Manager

Component	Description
Java Keystore (JKS)	Required to store the signature and encryption keys required by the X.509 token on the client. JKS the proprietary keystore format defined by Sun Microsystems. Trusted certificates and public and private keys are stored in the keystore. To create and manage the keys and certificates in the JKS, use the keytool utility. Keys are used for a variety of purposes, including authentication and data integrity. If the client and Web service are in the same domain with access to the same keystore, they can share the same private/public key pair: ■ The client can use the private key `orakey` to endorse the signature of the request message and the public key `orakey` to encrypt the symmetric key. ■ The Web service in turn uses the public key `orakey` to verify the endorsement, and the private key `orakey` to decrypt the symmetric key.
Policy Interceptors	In Oracle Fusion Middleware 11g, Oracle WSM Agents are managed by the security and management policy interceptors. Policy Interceptors enforce policies, including reliable messaging, management, addressing, security, and Message Transmission Optimization Mechanism (MTOM). The Oracle WSM Agent manages the enforcement of policies using the Policy Interceptor Pipeline. For complete Oracle Web Services Manager details, including the differences between release 10g and 11g, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.
Oracle WSM Agent	The OWSM agent is the certified WS-Trust client that can be used to communicate with OSTS. The OWSM agent is *embedded* and used by Oracle Security Token Service for message protection only (to publish WS Policy and to enforce message protection on inbound and outbound WS messages). Oracle Security Token Service performs token validation/request authentication. ■ Oracle Security Token Service embedded Oracle WSM Agent is used in the mode of "Message Protection Only" with authentication functionality disabled. This way all aspects related to authentication of incoming token are performed by Oracle Security Token Service only. ■ Oracle WSM supports disabling of authentication using configuration overrides that Oracle Security Token Service must declare with each policy. Exception: The Kerberos token is handled by Oracle WSM and Oracle Security Token Service is involved in mapping only the identity. ■ The OWSM Agent is one of the certified WS-Trust clients that can be used to communicate with OSTS. Other 3rd party WS-Trust clients can be used to interact with OSTS. Note: Embedded means that the OWSM Agent is available as part of the JRF layer on the WebLogic Server that OSTS uses:

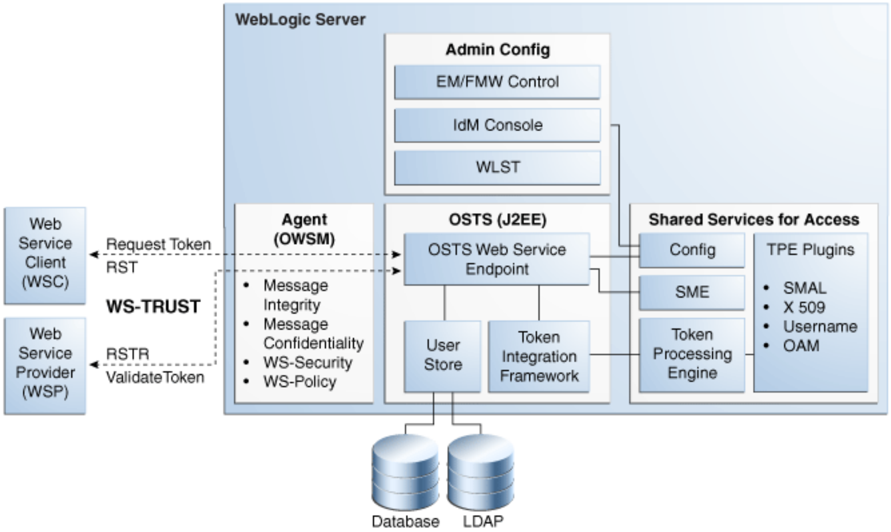
Table 1–7 (Cont.) Integrated Oracle Web Services Manager

Component	Description
Message/Token Protection	Oracle Security Token Service/Oracle Access Manager manages its own keystore and trust store. For Oracle WSM to enforce message protection for Oracle Security Token Service, the OWSM key store is seeded with its own self-signed certificate; passwords for its corresponding keys are stored in CSF. It does not work with OSTS keystore. Note: Conversely, Oracle WSM requires Oracle Access Manager/Oracle Security Token Service to store keys related to message protection in the OPSS Keystore. For cases where the client uses schemes such as SKI, Thumbprint, and so on to refer to its certificate, Oracle WSM requires that client certificate(s) are present in the OPSS Keystore.
Token Signing Key	Oracle Security Token Service has strong security requirements around its token signing key and uses the token signing key to broker trust between a client and a relying party. Therefore, this key must be stored in an exclusive partition that only Oracle Security Token Service can access.
Security Key Pairs	Oracle Security Token Service creates separate key pairs for issued token security and message security to provide security of token signing keys and eliminate the need for Oracle WSM agents to work with Oracle Access Manager/Oracle Security Token Service keystore: ▪ The message security key pair is populated to OPSS Keystore ▪ The token security key pair is populated to Oracle Access Manager/Oracle Security Token Service keystore
OPSS Keystore	The message security key pair is populated to OPSS Keystore. For special cases where clients use referencing schemes such as SKI (not a certificate token being received as part of the Web service request), Oracle Security Token Service populates OPSS Keystore with the requesting party's certificates. This is an uncommon scenario. Oracle Security Token Service can provide instructions on manually provisioning the keys to OPSS keystore to make it work.

1.2.4 About Oracle Security Token Service Architecture

Oracle STS, is a centralized token service that supports WS-Trust protocol, which defines extensions to the WS-Security specification for issuing and exchanging security tokens and establishing trust relationships. The Oracle STS is hosted as a web service endpoint and coordinates security based interactions between a WSC and a WSP. All communication with the Oracle STS occurs through a WS_Trust client, as shown in Figure 1–3.

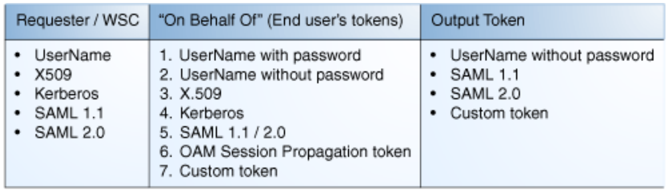
Figure 1–3 Oracle STS Architecture



When a WSC makes a call to the WSP, it gets the WS-Security policy that will indicate that a security token issued by Oracle STS should be presented. The policy will contain the location of Oracle STS, and the WSC will use that location to contact Oracle STS, and get the token expected by the WSP (Alternately, the WSP could register its acceptable security mechanisms with the Security Token Service and, before validating the incoming SOAP request, could check with the Security Token Service to determine its security mechanisms). When an authenticated WSC (carrying credentials that confirm either the identity of the end user or the application) requests a token for access to a WSP, the Security Token Service verifies the credentials and, in response, issues a security token that provides proof that the WSC has been authenticated. The WSC presents the security token to the WSP which verifies that the token was issued by a trusted Security Token Service.

Figure 1–4 shows the token support matrix for Oracle STS.

Figure 1–4 Oracle STS Token Support



Requester / WSC	"On Behalf Of" (End user's tokens)	Output Token
• UserName • X509 • Kerberos • SAML 1.1 • SAML 2.0	1. UserName with password 2. UserName without password 3. X.509 4. Kerberos 5. SAML 1.1 / 2.0 6. OAM Session Propagation token 7. Custom token	• UserName without password • SAML 1.1 • SAML 2.0 • Custom token

1.2.5 About Oracle Security Token Service Deployments

This section provides the following topics to introduce several different deployment options:

- Centralized Token Authority Deployment
- Tokens Behind a Firewall Deployment
- Web Services SSO Deployment

See Also: "Scenario: Identity Propagation with the OAM Token" on page 16-2

1.2.5.1 Centralized Token Authority Deployment

The need for a token exchange for security integration between Web SSO and Web service security tiers is in demand in a deployment where a Web application makes internal or external Web service calls.

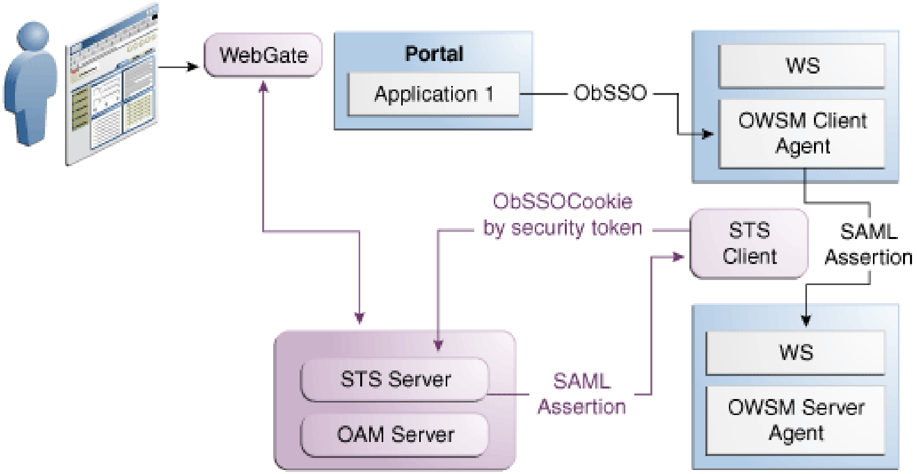
An example of such application is an intranet portal integration with external Web service provided by a partner or another organization within the same company. The portal needs a way to securely access the service.

The difficulty of security integration in this case stems from the fact that web SSO tier and WS tier use different methods of user authentication. In the Web SSO environment, the Web application can accept WAC-issued session tokens (SMSESSION, OBSSO), SAML assertions or proprietary tokens to authenticate the users.

The WS* security tier also uses a variety of tokes, standard and proprietary, and in most cases to achieve integration between the two tiers, local translation of token is required. In most cases, the WS performing the translation, needs to contact the authority by which the token was issued (Oracle Adaptive Access Manager) in order to decompose the token, before it can be translated. That requires every WS service to maintain trust with WAC systems. This is complex and not very secure because of multiple trust links that need to be maintained.

With the introduction of Oracle Security Token Service, the translation of tokens can be done at the centralized authority, as shown in Figure 1–5.

Figure 1–5 Token Translation at a Centralized Authority



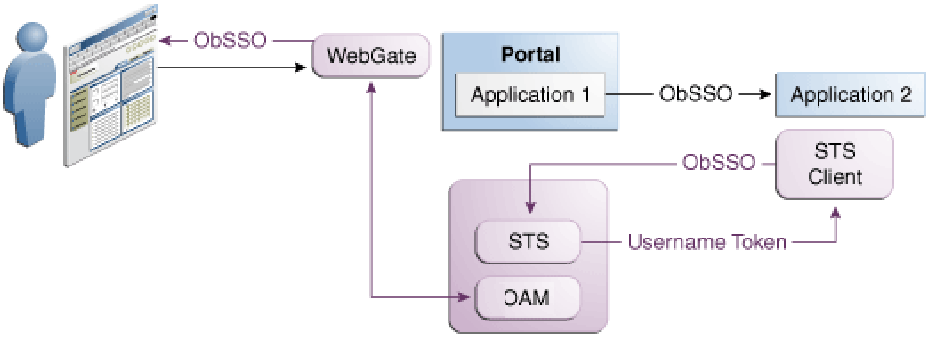
1.2.5.2 Tokens Behind a Firewall Deployment

The situation when applications rely on special form of credentials for their business logic is very common in deployments of Oracle access products. Integrations of WAC systems with both Oracle and custom applications almost always require extensive coding for (1) decomposing token issued by one token authority (such as OAM or SiteMinder) by calling a proprietary vendor API (SM agent API or ASDK) and (2) composing a new token format (PSFT, Siebel), that the application requires for its internal business logic.

Such translations are often handled through application coding, which introduces an element of risk of exposing user names and passwords when the code is deployed on multiple application instances in the DMZ.

Security administrators need an ability to control the translation process by externalizing it from the application. Introduction of Oracle Security Token Service provides significant relieve in this situation. Oracle Security Token Service plays a role of a centralized token authority, performing a translation of tokens behind the firewall, as shown in Figure 1–6.

Figure 1–6 Translating Tokens Behind a Firewall



Application 1 and Application 2 are protected by Oracle Access Manager. The Application 2 relies on a different type of token for its internal business logic. It has a client-side connector that contacts Oracle Security Token Service for exchanging the OBSSO token for a username token. The Oracle Security Token Service relies on Oracle Access Manager for decomposing the OBSSO token and generates a new token, required by Application 2.

This is more secure, because the same authority (Oracle Access Manager) performs both operations (composing and decomposing the OBSSO token). There is no need to decompose the token on the application side.

1.2.5.3 Web Services SSO Deployment

As in the Web SSO case, Web services SSO is a convenience feature. The difference is that in the case of Web SSO the party who benefits from the feature is a user. In the WS environment:

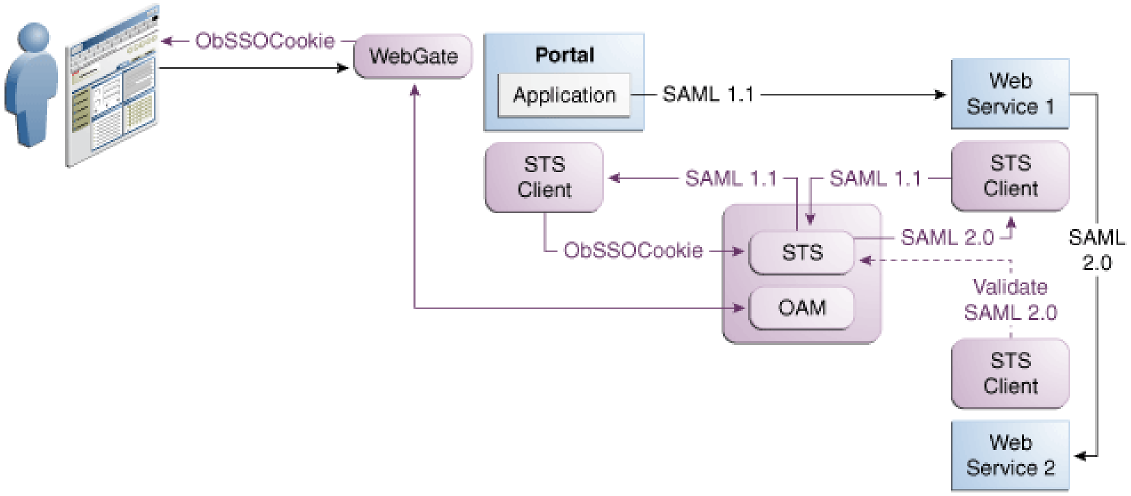
- Web SSO: The user benefits
- Web Services SSO: Security administrators benefit.

With Web services SSO different Web services have different token requirements, which change often. Externalizing the exchange to Oracle Security Token Service, however, enables the application to simply supply the target and the current token in its possession. Oracle Security Token Service takes charge of determining the token type for each requested service.

When one or more Web services change their authentication requirements, Oracle Security Token Service can seamlessly verify the token type submitted by the application. If the token is not of the requested type, the old token is revoked and the new one of the correct type is issued.

Figure 1–7 illustrates Web services SSO.

Figure 1–7 Web Services SSO



1.2.6 About Installation Options

This section provides the following topics as a brief overview to various installation options:

- Oracle Security Token Service Cluster in Single WLS Domain
- Endpoint Exposure through a Web Server Proxy
- Oracle Security Token Service Installation Overview
- Post-Installation Tasks: Oracle Security Token Service

1.2.6.1 Oracle Security Token Service Cluster in Single WLS Domain

You can leverage clustering across Oracle Security Token Service instances deployed in different managed servers within a single WebLogic domain. This deployment topology facilitates High Availability capabilities through a load balancer. By default, Oracle Access Manager co-exists on the same managed server as Oracle Security Token Service. However, Oracle Security Token Service is disabled by default and must be manually enabled before it can be used.

This deployment topology means that you:

- Deploy multiple instances of Oracle Security Token Service through the suite installer.
- Deploy a load balancer to support the High Availability and failover scenarios on the front of the Oracle Security Token Service cluster.

For more information, see the Oracle Fusion Middleware High Availability Guide.

1.2.6.2 Endpoint Exposure through a Web Server Proxy

This installation option provides inter-operability of Requester and Relying Party with Third-party STS Servers. At runtime, Oracle Security Token Service supports interoperability with Requesters and Relying Parties of third-party security token servers using the OPSS WS-Trust-Provider. For instance, a third-party security token service can create a valid SAML Assertion that can be consumed by Oracle Security Token Service.

1.2.6.3 Interoperability of Requester and Relying Party with Other Oracle WS-Trust based Clients

All run-time scenarios for Requesters and Relying Parties are supported by other Oracle WS-Trust Clients, including: WLSClient, MetroClient, and Oracle Web Services Manager (Oracle WSM).

All Web services clients are supported with Oracle Security Token Service only through the WS-Trust binding.

1.2.6.4 Oracle Security Token Service Installation Overview

Oracle Access Manager and Oracle Security Token Service are installed together from a single ear file. Oracle Access Manager and Oracle Security Token Service are deployed on the same managed server in a WebLogic domain.

The Oracle WSM Agent uses a keystore for various cryptographic operations. For those tasks, the Oracle WSM Agent uses the keystore configured for Oracle WSM tasks. During installation, if the Oracle WSM keystore service has not been configured, the installer:

- Creates a new keystore in the $*DOMAIN_HOME*/config/fmwconfig folder (default name is default-keystore.jks
- Creates a key entry with the corresponding certificate to be used by OWSM for signature and encryption operations. This key entry is stored in the OWSM Keystore under the `orakey` alias
- Stores the passwords of the key entry and of the keystore in CSF

Having access to the keystore is sometimes required, to:

- Extract the signing or encryption certificate to distribute to clients, if needed
- Update or replace the signing or encryption key entry
- Add trusted certificates

For more information, see the Oracle Fusion Middleware Installation Guide for Oracle Identity Management.

1.2.6.5 Post-Installation Tasks: Oracle Security Token Service

Any server hosting Oracle Access Manager with Oracle Security Token Service must be registered with Oracle Access Manager. This can occur automatically during installation, or manually after installation.

Elements in the Oracle Access Manager Console enable administrators to easily configure the Token Service to exchange WS Trust tokens with partners. Token Service elements provide for creation, viewing, modification, and removal of partners, endpoints, validation templates, issuance templates, and data store connections.

All Oracle Security Token Service system configuration is done using the Oracle Access Manager Console. This includes the following previously covered topics:

Look for information through out this manual. Pay close attention to Oracle Security Token Service details in Part V, "Oracle Security Token Service".

1.2.7 About Oracle Security Token Service Administration

A single default LDAP group, the WebLogic Server "Administrators" group, is set by default.

During initial deployment, using the Oracle Fusion Middleware Configuration Wizard, the administrator userID and password are set. Administrators can log in and use the Oracle Access Manager Console (and WebLogic Server Administration Console).

For more information, see Chapter 3, "Getting Started with Common Administration and Navigation".

2

Introduction to This Book

This chapter provides an introduction to this book and links to chapters where you can find more information. This chapter contains the following sections:

- Introduction to This Book
- Part I: Oracle Product Introduction
- Part II: Common Tasks
- Part III, Oracle Access Manager Settings
- Part IV, Single Sign-on, Oracle Access Manager Policies, and Testing
- Part V: Oracle Security Token Service
- Part VI: Common Logging, Auditing, Performance Monitoring
- Part VII: Using OAM 10g Webgates with OAM 11g
- Part VIII: Appendixes

2.1 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.

Following sections introduce each book part and chapter.

2.2 Part I: Oracle Product Introduction

The first part of this book introduces the products described in this book, and system management tasks common to both Oracle Access Manager and Oracle Security Token Service. The following chapters are provided:

- Chapter 1, "Oracle Product Introduction"
- Chapter 2, "Introduction to This Book"

2.3 Part II: Common Tasks

Oracle Access Manager 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 Oracle Access Manager and WebLogic. Initially, the same credentials are used for log in to both the Oracle Access Manager and the WebLogic Server Administration Console. The LDAP group for Oracle Access Manager administrators can be changed.

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

- Getting Started with Common Administration and Navigation
- Managing Services, Certificate Validation, and Common Settings
- Managing Common Data Sources
- Managing Common OAM Server Registration
- Managing Sessions

2.3.1 Getting Started with Common Administration and Navigation

Administrators use the:

- Oracle Access Manager Console to register and manage Oracle Access Manager system configurations and security elements and policies.

  For a quick tour of Oracle Access Manager Console and the most common functions and tasks, see Chapter 3, "Getting Started with Common Administration and Navigation".

  > **Note:** Custom Administrative command-line tools (WebLogic Scripting Tool, also known as WLST) provide an alternative to the Oracle Access Manager 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 Oracle Access Manager WebLogic Scripting Tool for command-line input
- Remote registration tool for registering agents and application domains

2.3.2 Managing Services, Certificate Validation, and Common Settings

Chapter 4 explains how to navigate to and configure properties that are common to both Oracle Access Manager and Oracle Security Token Service. This chapter includes the following topics:

- Introduction to Common Configuration Elements
- Enabling or Disabling Available Services

- Managing the Common Settings
- Managing Global Certificate Validation and Revocation

2.3.3 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 Oracle Access Manager 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 Oracle Access Manager 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, Oracle Access Manager 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. In this case, one store must be designated as the System Store that contains Administrator roles and users.

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

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

  Session Store: By default, Oracle Access Manager 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 5. For information about sessions, see Chapter 7.

  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 24, "Auditing Administrative and Run-time Events".

- A Java keystore is associated with Oracle Access Manager 11g and used to store security keys that are generated to encrypt agent traffic and session tokens. Every

Oracle Access Manager 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 Console. For more information, see Chapter 5, "Managing Common Data Sources".

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

2.3.4 OAM Server Instances and the Console

OAM Servers were known as Access Servers in Oracle Access Manager 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 Oracle Access Manager Console was previously known as Policy Manager in OAM release 10g. The Oracle Access Manager 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 Oracle Access Manager 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 services, which can be managed using the Oracle Access Manager Console. See Chapter 4, "Managing Services, Certificate Validation, and Common Settings".

You can use the Oracle Access Manager Console to manage server registrations, as described in Chapter 6, "Managing Common OAM Server Registration".

Settings that are specific to Access Manager operations are described in Chapter 8, "Configuring Access Manager Settings".

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 6, "Managing Common OAM Server Registration"
- The WebLogic Configuration Wizard
- Customized Oracle WebLogic Scripting Tool (WLST) commands for Oracle Access Manager

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

See Also: Appendix F, "Introduction to Custom WLST Commands for 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 "Single Sign-on Agents" on page 2-6.

2.3.5 Oracle Access Manager 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 7, "Managing Sessions".

2.4 Part III, Oracle Access Manager Settings

This section describes the content of Part III of this book:

- Access Manager Settings
- Single Sign-on Agents

2.4.1 Access Manager Settings

The Access Manager Setting section of the System Configuration tab provides a number of settings specific to Access Manager service operations.

Included in this chapter (Chapter 8) are the following Access Manager-specific settings:

- Introduction to Access Manager Settings
- Managing Access Manager Load Balancing Settings
- Managing SSO Tokens and IP Validation
- Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security
- Managing Run Time Policy Evaluation Caches
- Managing Authentication modules

2.4.2 Single Sign-on Agents

A single sign-on agent (also known as a policy-enforcement agent, or simply an 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 Oracle Access Manager 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, registered 10g Webgates communicate with Oracle Access Manager 11g services through a Java EE-based OAM proxy that acts as a bridge.

    IAMSuiteAgent: 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 Agent performs as an OAM 10g Agent to enforce Oracle Access Manager 11g policies. This agent replaces the earlier IDMDomainAgent

    IDMDomainAgent: This earlier Java agent is replaced by IAMSuiteAgent. After applying the Oracle Access Manager 11g patch set, IDMDomainAgent and its companion application domain are decommissioned.
  - **Access Client**: A custom programmatic access client created using the Access Manager software developer kit (SDK). Access Clients 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 Oracle Access Manager 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:

- **Oracle Access Manager Console**: Register and manage OAM and OSSO agents as described in Chapter 9
- **Remote Registration**: Use the Oracle-provided command-line tool as described in Chapter 10.
- **Programmatic Agent Registration**: This is the same as registering OAM and OSSO agents using the console or remote registration tool. See Chapter 9 and Chapter 10.

From an existing 10g Oracle Access Manager or OSSO deployment you can:

- Provision 10g Webgates with Oracle Access Manager 11g, as described in Chapter 27.
- Upgrade OracleAS 10g SSO (OSSO) as described in the Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management. Read about co-existence with Oracle Access Manager 11g Servers in Appendix A.

> **See Also:** "Co-existence: OAM 11g SSO versus OAM 10g SSO with OracleAS SSO 10g" on page 2-13

2.5 Part IV, Single Sign-on, Oracle Access Manager Policies, and Testing

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

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

2.5.1 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 Console, and other Oracle Identity Management consoles deployed in a WebLogic container, is enabled using the pre-registered IAMSuiteAgent and companion application domain (IAMSuite). 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 Oracle Access Manager 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 Oracle Access Manager 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 Oracle Access Manager 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 Oracle Access Manager 11g and OAM 10g Agents (out of the box Webgates and programmatic access clients, and OSSO Agents (mod_osso 10g), which can be used in any combination.

See Also:

- "OAM Server Instances and the Console" on page 2-4
- "Single Sign-on Agents" on page 2-6
- Chapter 11, "Introduction to the OAM Policy Model, Single Sign-On"

2.5.2 Oracle Access Manager 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 12, "Managing Policy Components".

2.5.3 Oracle Access Manager 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.

> **Note:** To enhance security, Oracle Access Manager 11g default behavior is to deny access when a resource is not protected by a policy that explicitly allows access. In contrast, Oracle Access Manager 10g default behavior allowed access when a resource was not protected by a rule or policy that explicitly denied access.
>
> Oracle Access Manager 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:** Oracle Access Manager 10g enables authorization actions to be taken depending on the evaluation of the administrator-defined authorization expression contained one or more authorization rules.

- Token Issuance Policies and Constraints for Specific Resources

  A Token Issuance Policy defines the rules under which a token can be issued for a resource (Relying Party Partner) based on the client's identity, with the client either being a Requester Partner or an end user.

  For details, see Chapter 19, "Managing Templates, Endpoints, and Policies".

For more information about the policy model and application domains, see Chapter 13, "Managing Policies to Protect Resources and Enable SSO".

2.5.4 Connectivity and Policy Testing

Oracle provides a portable, stand-alone Java application that replaces the Oracle Access Manager 10g Access Tester function. The Oracle Access Manager 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 14, "Validating Connectivity and Policies Using the Access Tester".

2.5.5 Centralized Logout for Oracle Access Manager 11g

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

For details, see Chapter 15, "Configuring Centralized Logout for OAM 11g".

2.6 Part V: Oracle Security Token Service

This section provides information to help administrators manage the Oracle Security Token Services available with Oracle Access Manager

- Chapter 16, "Oracle Security Token Service Implementation Scenarios"
- Chapter 17, "Managing Oracle Security Token Service Settings and Set Up"
- Chapter 18, "Managing Oracle Security Token Service Certificates and Keys"
- Chapter 19, "Managing Templates, Endpoints, and Policies"
- Chapter 20, "Managing Token Service Partners and Partner Profiles"

- Chapter 21, "Troubleshooting Oracle Security Token Services"

2.7 Part VI: Common Logging, Auditing, Performance Monitoring

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

- Component Event Message Logging
- Webgate Event Message Logging
- Common Audit Framework
- Performance Metrics in the Oracle Access Manager Console
- Performance Metrics in Fusion Middleware Control

2.7.1 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 22, "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 Administrators" introduces custom WLST commands to change OAM logging levels

2.7.2 Webgate Event Message Logging

Each Webgate instance (both 10g and 11g Webgates) can write information about its processes and states to a log file. The logs can be configured to provide information at various levels of granularity. For example, you can record errors, errors plus state information, or errors, states, and other information to the level of a debug trace. You can also eliminate sensitive information from the logs.

For more information, see Chapter 23, "Logging Webgate Event Messages".

2.7.3 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 adherence 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 Oracle Access Manager Console.

Administrators can control and specify certain auditing parameters using the Oracle Access Manager 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 Oracle Access Manager 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 accessible 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 24, "Auditing Administrative and Run-time Events".

2.7.4 Performance Metrics in the Oracle Access Manager 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.

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

For more information, see Chapter 25, "Monitoring Performance by Using Oracle Access Manager Console".

2.7.5 Performance Metrics in Fusion Middleware Control

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

For more information, see Chapter 26, "Monitoring Performance and Logs with Fusion Middleware Control".

2.8 Part VII: 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

2.8.1 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 27, "Managing OAM 10g Webgates with OAM 11g".

2.8.2 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 28, "Configuring Apache, OHS, IHS for 10g Webgates".

2.8.3 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 29, "Configuring the IIS Web Server for 10g Webgates".

2.8.4 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 30, "Configuring the ISA Server for 10g Webgates".

2.8.5 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 31, "Configuring Lotus Domino Web Servers for 10g Webgates".

2.9 Part VIII: 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

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

Table 2–1 outlines several ways to use OAM 11g when you have various starting points.

Table 2–1 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 A.
Web Servers other than Oracle HTTP Server	See Chapter 27 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 A for an introduction to post-upgrade co-existence between OAM 11g and OSSO 10g Servers.

2.9.2 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 B, "Transitioning OAM 11g from a Test to a Production Environment".

2.9.3 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".

2.9.4 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.

2.9.5 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 (OAP) 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: Also known as the OAP 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 for Oracle Access Manager 11g". See also:

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

2.9.6 Custom WebLogic Scripting Tool Commands for OAM

Only 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 Administrators".

2.9.7 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".

2.9.8 Creating Deployment-Specific Pages

Oracle Application Server Single Sign-On provides a framework for integrating deployment-specific login, change password, and single sign-off pages with the single sign-on server. This means that you can tailor these pages to your UI look and feel and globalization requirements.

For more information, see Appendix H, "Creating Deployment-Specific Pages"

2.9.9 Troubleshooting

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

Part II

Using the Console for Common Tasks

Part II provides information about managing common system configuration details for Oracle Access Manager and Oracle Security Token Service.

Part II contains the following chapters:

- Chapter 3, "Getting Started with Common Administration and Navigation"
- Chapter 4, "Managing Services, Certificate Validation, and Common Settings"
- Chapter 5, "Managing Common Data Sources"
- Chapter 6, "Managing Common OAM Server Registration"
- Chapter 7, "Managing Sessions"

3

Getting Started with Common Administration and Navigation

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

- Prerequisites
- Introduction to Administrators
- Logging In to and Signing Out of Oracle Access Manager Console
- Introduction to the Oracle Access Manager Console and Controls
- Introduction to Policy Configuration and System Configuration Tabs
- Viewing Configuration Details in the Console
- Conducting Searches Using the Console
- Using Online Help
- Command-Line Tools
- Logging, Auditing, Monitoring Performance

WARNING: Starting the AdminServer the first time can take an unusually long time: 12-15 minutes, for example. This process must not be interrupted or terminated. If the startWebLogic.cmd (Windows; startWebLogic.sh) is stopped for any reason (whether accidently or a system crash or a reboot, for example), policy data might be corrupted. This would require removal and recreation of the domain and rerunning the RCU to create the OAM schema.

3.1 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 Console when the WebLogic Administration Server is running. If the Oracle Access Manager Console is protected by a Webgate, the OAM Server must be running.

Before you begin tasks in this chapter:

- Learn about the Oracle Access Manager Console as described in "Introduction to the Oracle Access Manager Console and Controls" on page 3-4.
- Verify the administrative LDAP group defined in the primary user identity store.

Note: The default LDAP group, "Administrators", is set during initial deployment using the Oracle Fusion Middleware Configuration Wizard, as described in "Introduction to Administrators".

3.2 Introduction to Administrators

A single default LDAP group, the WebLogic Server "Administrators" group, is set in the Default User Identity Store (Embeded LDAP). During initial deployment, using the Oracle Fusion Middleware Configuration Wizard, the administrator userID and password are set. These credentials grant access to:

- WebLogic Server Administration Console
- Oracle Access Manager (with Oracle Security Token Service)
- Remote registration administrative tool
- WebLogic Scripting Tool (WLST)
- Customized Oracle Access Manager WLST commands

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

Table 3–1 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 common Default System User Identity Store.

Table 3–1 Role Mapping from an LDAP Group to Administrator

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

Initially, administrative users must log in to the Oracle Access Manager Console using the WebLogic Administrator credentials set during initial configuration. However, your enterprise might require independent sets of administrators: one set of users responsible for Oracle Access Manager with Oracle Security Token Service administration and a different set for WebLogic administration. For more information, see "Managing the Administrators Role" on page 5-14.

Note: Concurrent configuration updates are not supported. Only one administrator should be allowed to modify the system configuration at any given time. Administrators performing updates concurrently will result in an inconsistent state within the system configuration of the Oracle Access Manager Console.

3.3 Logging In to and Signing Out of Oracle Access Manager Console

The Oracle Access Manager Console provides administrative access to Oracle Access Manager with Oracle Security Token Service. This section describes how to log in to and sign out of the Oracle Access Manager 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*.

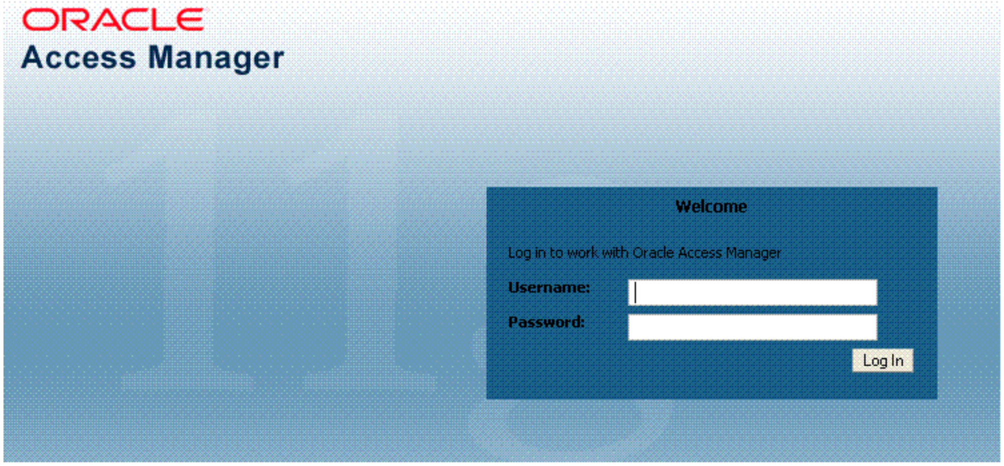
This section provides the following topics:

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

3.3.1 Logging In to the Oracle Access Manager Console

The Oracle Access Manager Console log in page is shown in Figure 3–1.

Figure 3–1 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 Administrator is the same as the LDAP group defined for the WebLogic Server Administration Console ("Administrators") and the common Default System User Identity Store store is the WebLogic Embedded LDAP.

To log in to Oracle Access Manager Console

1. In a browser window, enter the URL to the Oracle Access Manager Console 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 Console
- *port* refers to the designated bind port for the Oracle Access Manager Console (this is the same as the bind port for the WebLogic Server Administration Console)
- /oamconsole/refers to the Oracle Access Manager Console Log In page

2. On the Log In page, enter the Oracle Access Manager Console Administrator credentials. For example:

   Username: *Admin_login_id*

   Password: *Admin_password*

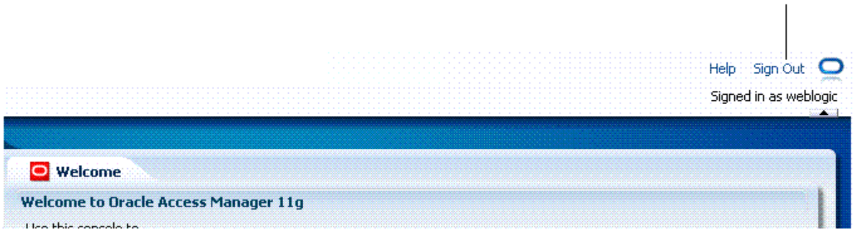
3. Click the Log In button or press the Enter key.
4. Proceed as follows:
   - **Successful**: Policy Configuration and System Configuration tabs appear on the left; Welcome page is on the right. Tour the console, as described in "Introduction to Policy Configuration and System Configuration Tabs" on page 3-14 or start performing tasks on your own.
   - **Not Successful**: Log in again and ensure that you enter information exactly as specified for the Administrator in the common Default System User Identity Store.

   See Also: "Introduction to Administrators" on page 3-2

3.3.2 Signing Out of Oracle Access Manager Console

The Sign Out link appears in the upper-right corner of the Oracle Access Manager Console, as shown in Figure 3–2. You select the Sign Out link to conclude your session. Oracle recommends that you also close the browser window after signing out.

Figure 3–2 Sign Out Link, Upper-right Corner



Signed in as weblogic

Welcome to Oracle Access Manager 11g

To sign out of Oracle Access Manager Console

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

3.4 Introduction to the Oracle Access Manager Console and Controls

The Oracle Access Manager 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 Console.

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

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

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

> **Note:** Concurrent configuration updates are not supported. Only one administrator should be allowed to modify the system configuration at any given time. Administrators performing updates concurrently will result in an inconsistent state within the system configuration of the Oracle Access Manager Console Console.

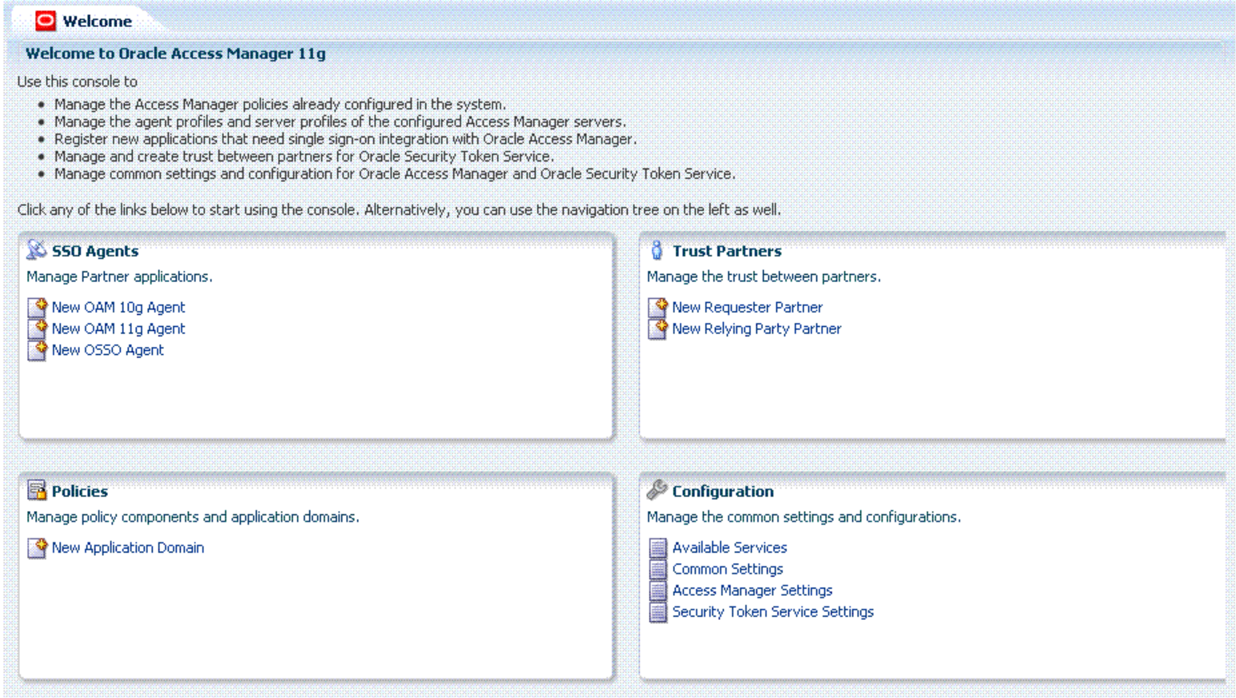
This section provides a quick introduction to orient you to the Oracle Access Manager Console.

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

3.4.1 Console Layout and Controls

Figure 3–3 provides a look at the Oracle Access Manager Console as it appears immediately after log in.

Figure 3–3 Oracle Access Manager Console Welcome 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.
- Manage and create trust between partners for Oracle Security Token Service.
- Manage common settings and configuration for Oracle Access Manager and Oracle Security Token Service.

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

SSO Agents

Manage Partner applications.

Trust Partners

Manage the trust between partners.

Policies

Manage policy components and application domains.

Configuration

Manage the common settings and configurations.

The Oracle Access Manager 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:

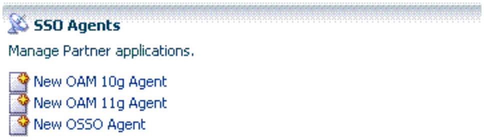
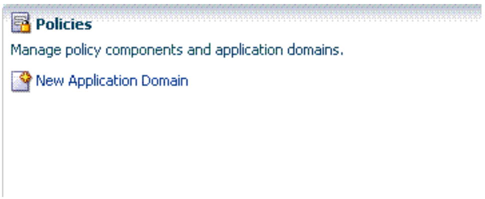
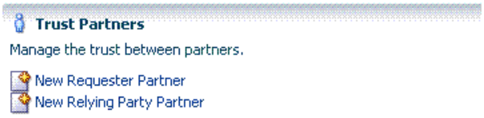
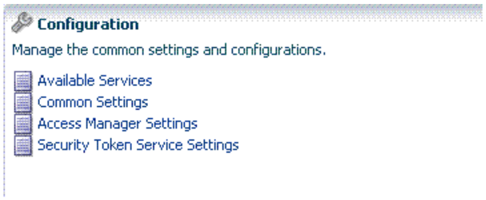
- Welcome Page and Shortcuts
- Function-Level Tabs and Controls
- Content Pages and Page Controls

See Also: "Selecting Controls in the Console" on page 3-13

3.4.1.1 Welcome Page and Shortcuts

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 3–2.

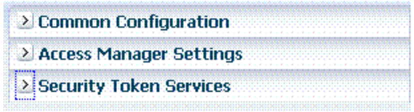
Table 3–2 Welcome Page and Shortcuts

Welcome Page Shortcut Section	Description
 SSO Agents Manage Partner applications. New OAM 10g Agent New OAM 11g Agent New OSSO Agent	Click **New ..** to launch a fresh Create Agent page. See "About the System Configuration Tab" on page 3-14 for more information.
 Policies Manage policy components and application domains. New Application Domain	Click **New Application Domain** to launch a fresh Application Domains page. See "About the Policy Configuration Tab" on page 3-15 for more information.
 Trust Partners Manage the trust between partners. New Requester Partner New Relying Party Partner	Click **New ...** to launch a fresh page where you can enter appropriate details. See Part V, "Oracle Security Token Service"for more information.
 Configuration Manage the common settings and configurations. Available Services Common Settings Access Manager Settings Security Token Service Settings	See the following topics for more information on each of the choices on the Configuration Shortcuts panel: ■ "Enabling or Disabling Available Services" on page 4-2 ■ "Managing the Common Settings" on page 4-3 ■ Chapter 8, "Configuring Access Manager Settings" ■ Chapter 17, "Managing Oracle Security Token Service Settings and Set Up"

3.4.1.2 Function-Level Tabs and Controls

Table 3–3 introduces the function-level tabs in the Oracle Access Manager Console.

Table 3–3 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 3-15 for more information.
System Configuration Note: This is not the active tab when you enter the console. 	Provides access to system-level definitions as shown here.  Common Configuration Access Manager Settings Security Token Services See "About the System Configuration Tab" on page 3-14 for more information.

The following topics provide more information about specific controls:

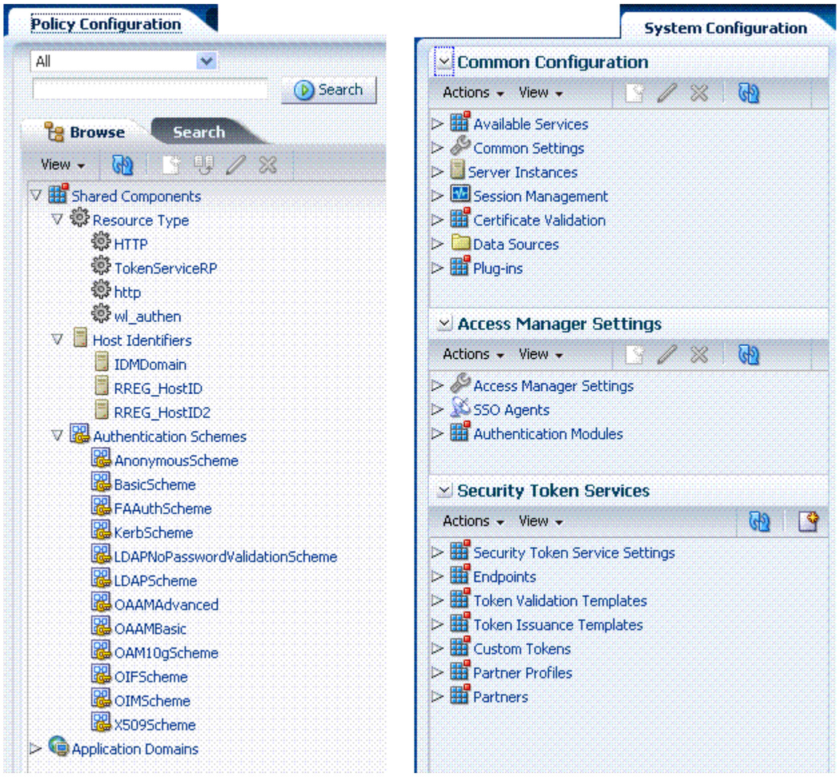
- Navigation Tree
- Menu and Tool Bar
- View Menu
- Actions Menu

> **See Also:** "Selecting Controls in the Console" on page 3-13

3.4.1.2.1 Navigation Tree A navigation tree for the active configuration tab is provided on the left side of the console. 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 3–4. Notice the menu and tool bars above each navigation tree.

Figure 3–4 Sample Navigation Trees with Menu and Tool Bars



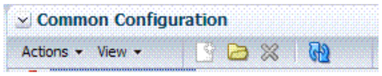
See Also:

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

3.4.1.2.2 Menu and Tool Bar 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 a tool bar for quick access. A menu and tool bar appear above each navigation tree in the console.

A different collection of menus and command buttons is available, depending upon the tab or subtab you are viewing. Additionally, certain configuration pages within the console include tables that provide a menu and tool bar. Figure 3–5.

Figure 3–5 Menu and Tool Bar Above Common Configuration Navigation Tree



Command buttons appear in full color when the related function is available for use. When a function cannot be used, the command button (or menu item) appears grey. For instance, you can open a node and edit or delete a selected registration or definition. Table 3–4 provides a description of each command button in the tool bar. Some buttons are only available

Table 3–4 Command Buttons in the Tool Bar

Button	Definition	Description
🗋	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 3–6.
📂	Open	Opens the selected instance in the navigation tree.
✎	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 3–6.
🔄	Refresh	Revives the navigation tree, in the same way a Web browser refreshes a Web page.
⧉	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. The number of characters a supported URL is based on browser version. Ensure that your applications do not use URLs that exceed the length that Oracle Access Manager and the browser can handle.
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.

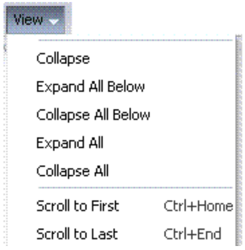
Most commands available as buttons in the tool bar are also available on a menu.

See Also:

- View Menu
- Actions Menu

3.4.1.2.3 View Menu Figure 3–6 illustrates the View menu, which is available for use with both the Policy Configuration and System Configuration tabs.

Figure 3–6 View Menu



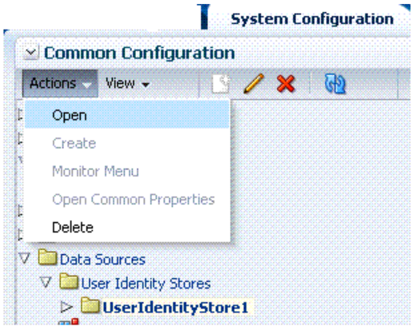
Unavailable items (those that cannot be used on the selection in the navigation tree) appear in grey. View menu command descriptions are provided in Table 3–5.

Table 3–5 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.

3.4.1.2.4 Actions Menu This menu is available only when the System Configuration tab is active. Figure 3–6 illustrates the Actions menu, which provides appropriate commands for the selected instance in the navigation tree. For example, if you have an identity store instance selected in the navigation tree one of the commands on the Actions menu enables you to open the Common Properties page for viewing or editing.

Figure 3–7 Actions Menu



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. Actions menu command descriptions are provided in Table 3–6.

Table 3–6 System Configuration, Actions 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 3–4.
Monitor Menu	Displays the monitoring page for the Agent selected in the navigation tree. For more information, see Chapter 25.
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 a system instance 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 3–4.

3.4.1.3 Content Pages and Page Controls

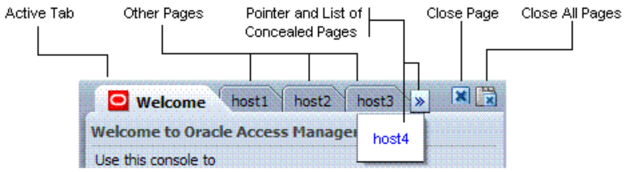
Like the Welcome page, any open content pages appear on the right side of the console.

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 3–8 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 3–8 Tabs of Open Pages, and Page Controls



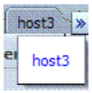
Each page appears only once. No warning is issued if you attempt to open the same page multiple times. However, the page is only one time.

> **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 3–7.

See Also: "Selecting Controls in the Console" on page 3-13

Table 3–7 Controls for Open Pages

Page Control	Definition	Description
 host3 » host3	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.

3.4.2 Elements on a Page

Pages in the console contain one or more graphical user interface elements as described in Table 3–8. For an example of each element n the console, see Figure 3–8 or log into the console and have a look.

Table 3–8 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 3–2 for an example.
Page controls	Enables you to close one or more pages. See Table 3–7.
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.

Table 3–8 (Cont.) Page Elements and Descriptions

Page Element	Description
Command buttons for tables ✚ ✖	Enables you to: Add a fresh row or definition to the table. Remove the selected row or definition from the table.
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.

3.4.3 Selecting Controls in the Console

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

Table 3–9 describes selections and controls.

Table 3–9 Selection Tasks and Controls

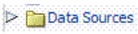
Task	Control	Description
Expand a node	 ▷ Data Sources	Click the Expand button (>) beside the desired node in the navigation tree to reveal nodes or instances within it.
Collapse a node	▽ Agents	Click the Collapse button (V)beside the desired node in the navigation tree to conceal nodes or instances 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.
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: Resources and Responses, for instance. ■ Named Page tab to reveal (activate) the page ■ Text field to enter information on a page ■ Page Control (close or close all buttons as described in Table 3–7)
Open	Double click mouse button or the Open Folder icon	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

Table 3–9 (Cont.) Selection Tasks and Controls

Task	Control	Description
Highlight	Drag cursor	Drag the cursor across text in a box to highlight its content.
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 3–4) Close page buttons (Table 3–7) ■ Command Button on a Page or Table: Apply: Submits additions and changes on the active page. Table or section buttons (Table 3–8) ✚ Add a new row. ✖ Remove the selected row. ■ Links: Help, and Sign Out are examples

3.5 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

3.5.1 About the System Configuration Tab

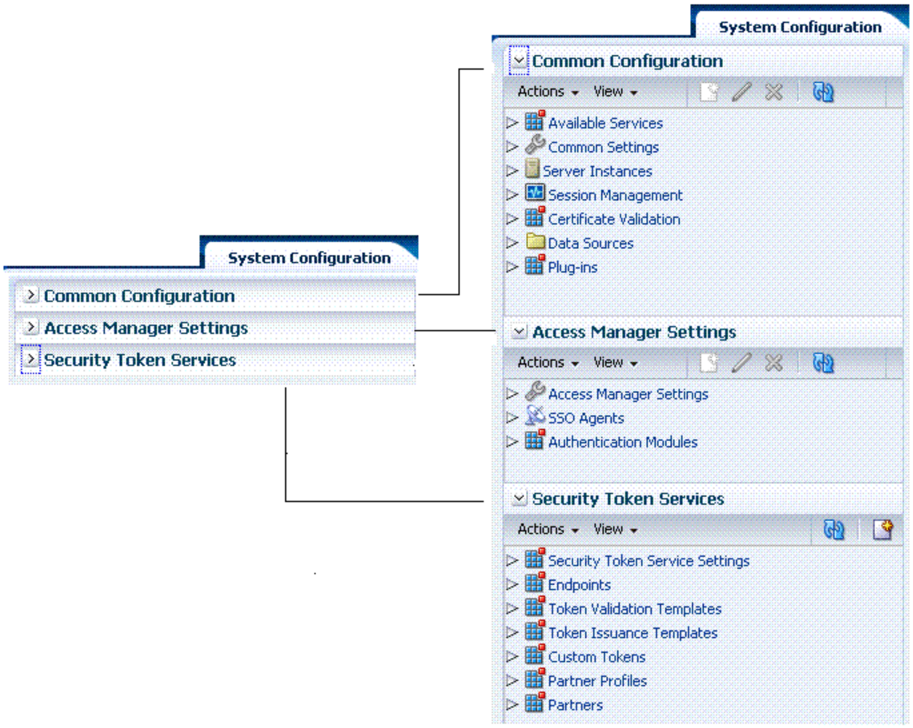
In the Oracle Access Manager Console, the white tab is active. This section describes the System Configuration tab, which provides three independent sections:

- Common Configuration
- Access Manager Settings
- Security Token Services

Each section can be viewed and used independently because each section provides an independent menu and tool bar above the related 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. Figure 3–9 shows the System Configuration tab in both fully collapsed and fully expanded form.

Figure 3–9 System Configuration Tab (Collapsed and Fully Expanded)



For more information about system configuration, see:

- Part II, "Using the Console for Common Tasks"
- Part III, "Oracle Access Manager Settings Management"
- Part V, "Oracle Security Token Service"

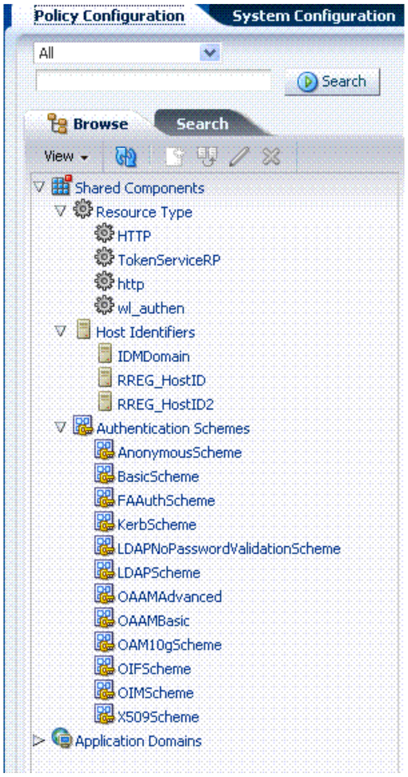
3.5.2 About the Policy Configuration Tab

The Policy Configuration tab in the Oracle Access Manager Console gives administrators access to authentication and authorization policies and shared SSO components.

The Browse subtab provides a view of the navigation tree. The Search subtab provides a view of your search results for policy elements.

The view in Figure 3–10 lists first-level items beneath Shared Components and Authentication Schemes.

Figure 3–10 Policy Configuration, Shared Components, Collapsed Application Domains



- Shared Components
  - Resource Type
    - HTTP
    - TokenServiceRP
    - http
    - wl_authen
  - Host Identifiers
    - IDMDomain
    - RREG_HostID
    - RREG_HostID2
  - Authentication Schemes
    - AnonymousScheme
    - BasicScheme
    - FAAuthScheme
    - KerbScheme
    - LDAPNoPasswordValidationScheme
    - LDAPScheme
    - OAAMAdvanced
    - OAAMBasic
    - OAM10gScheme
    - OIFScheme
    - OIMScheme
    - X509Scheme
- Application Domains

Table 3–10 describes the Policy Configuration subtabs.

Table 3–10 Policy Configuration Subtabs

Subtab	Description
Browse Browse	On the Policy Configuration tab, the Browse subtab displays 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 Console.
Search Results Search Results	On the Policy Configuration tab, the Search subtab 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 Using the Console" on page 3-17.

See "Console Layout and Controls" on page 3-5 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 VI, "Common Logging, Auditing, Performance Monitoring"

3.6 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 Oracle Access Manager 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 Administrators"

To view configuration details using the console

1. Go to the Oracle Access Manager 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 Console; *port* refers to the HTTP port number on which the console host listens; /oamconsole identifies the Oracle Access Manager Console.

2. System Configuration:

   a. Click the System Configuration tab.

   b. In the navigation tree, click the desired section name from those listed:

      Common Configuration
      Access Manager Settings
      Security Token Service Settings

   c. Expand: Click the expansion icon besided the desired node (or from the View menu, click Expand All).

   d. Open Instance: Double click the desired instance name to view its configuration.

3. Policy Configuration:

   a. Click the Policy Configuration tab (available when you enter the console by default).

   b. Expand: Click the expansion icon besided the desired node (or from the View menu, click Expand All).

   c. Open Instance: Double click the desired instance name to view its configuration.

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

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

3.7 Conducting Searches Using the Console

The Oracle Access Manager Console provides separate search controls for policy elements versus system instances (agents, for instance). This section describes only policy configuration search controls.

- Conducting Policy Element Searches Using the Console

- Refining Searches for System Configuration Elements

3.7.1 Conducting Policy Element Searches Using the Console

This section provides the following topics:

- About Policy Configuration Search Controls
- Searching for Policy Elements

3.7.1.1 About Policy Configuration Search Controls

When searching for a policy configuration element, you can use a wild card in the search string if you do not know the exact name of the item you are trying to locate. From the search results table, you can choose an item to open and view or edit.

> **Note:** You can use a wildcard (*) character if you do not know the exact name you seek. However, capitalization in your search criteria must match capitalization of the item you seek.

You cannot save your policy configuration search criteria. When you select the System Configuration tab and return to the Policy Configuration tab, the search field and results table are empty.

Policy Configuration search controls are shown and described in Table 3–11.

Table 3–11 Policy Configuration Search Controls

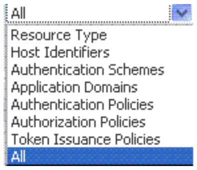
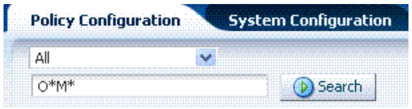
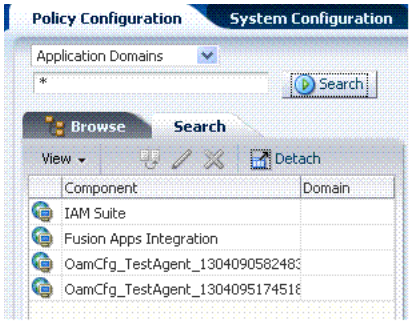
Search Control	Description
 All Resource Type Host Identifiers Authentication Schemes Application Domains Authentication Policies Authorization Policies Token Issuance Policies All	From the Policy Configuration Search menu, choose an item to define your search or simply select All (the default).
 Policy Configuration System Configuration All SearchO*M*	In the text field, enter the name of the instance you want to find (or enter a partial name with a wild card (*) in the search string). Click the Search button to initiate the operation.

Table 3–11 (Cont.) Policy Configuration Search Controls

Search Control	Description
 Policy Configuration System Configuration Application Domains * Search Browse Search View Detach Component | Domain IAM Suite Fusion Apps Integration OamCfg_TestAgent_130409058248 OamCfg_TestAgent_130409517451	Click the Search subtab to reveal the results of your search. Double-click a name in the results table to open the item to view or edit. Alternatively, click a name in the results table and then click a command button in the tool bar. 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. **Note**: Clicking the System Configuration tab clears the Policy Configuration search field and results table.

3.7.1.2 Searching for Policy Elements

This topic describes how to perform a search for a policy element using the Oracle Access Manager Console.

> **Note:** Wild cards (*) are allowed in the search string.

In thfollowing example, a search is conducted for a specific host identifier.

To search for a policy element

1. Activate the Policy Configuration tab.
2. From the search type list, choose a type to define your search. For example:

   Host Identifiers

3. In the text field, enter the name of the instance you want to find (or partial name with wild card). For example:

   *my**

4. Click the Search button to initiate the operation.
5. From the Search Results tab, click a name in 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.

3.7.2 Refining Searches for System Configuration Elements

This section introduces some of the search controls you will encounter as you create or manage various System Configuration elements using the Oracle Access Manager Console:

- OAM Agents
- OSSO Agents
- STS Templates
- STS Partners
- STS Profiles

Table 3–12 describes the common controls available to refine a such a search. Though the type of control is common, actul selections in lists will apply to the specifi item you are seeking.

> **See Also:** Individual search topics that appear with related component management topics this guide.

Table 3–12 Common System Configuration Search Controls

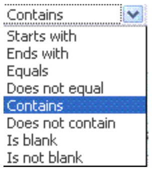
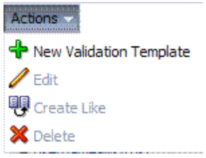
Element	Description
Match	Choose All to search for a template that matches all your specifications. Choose Any to search for a template that matches at least one of your specifications.
Search Operations List	A list of operations from which you choose one to help refine your search.  Contains Starts with Ends with Equals Does not equal Contains Does not contain Is blank Is not blank
Description	Refine your search using the optional description field.
Search	Initiates the Search function using criteria in the form.
Reset	Resets the Search form with defaults only.
Add Fields *Shown: For STS Templates*	A list of additional items you can add as search criteria. This is not available for all components.
Search Results Table	Itemizes the results of your search based on the component you were searching for and the choices in the View menu, described later in this table.

Table 3–12 (Cont.) Common System Configuration Search Controls

Element	Description
Actions menu *Shown: Actions for Validation Template*	Provides the following functions that can be performed on a selection in the results table:  Actions New Validation Template Edit Create Like Delete Note: Actions menu functions mirror command buttons above the results table. For example: ■ New ... Template: Click the New ... Template button at the top of the Search page, or select New ... Template from the menu, or click the **+** button above the table. ■ Edit: Click a name in the Results Table, or select Edit from the Actions menu, or click the Edit (pencil) command button above the Results Table. ■ Create Like: Select the desired row in the table and either select Create Like from the Actions menu, or click the Create Like command button above the table ■ Remove: Select the desired row in the Results Table and either select Delete from the Actions menu, or click the Delete (**X**) command button above the table.
View menu	A list from which you can identify which information to display in the results table.
	Controls you can choose to define the order of items listed in the results table: ■ Ascending ■ Descending

To refine your search for a System Configuration target

1. Activate the System Configuration tab and expand the desired section:
   - Common Configuration
   - Access Manager Settings
   - Security Token Service Settings
2. Within the section, expand the desired node, and open the terminal node to display the Search page.
3. **Search**: Choose the desired controls and enter your search criteria, then click the Search button.
4. In the Search Results table:
   - **Edit or View:** Click the Edit command button in the tool bar to display the editable configuration page.
   - **Delete:** Click the Delete (X) button above the table 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.

- **Reconfigure Table**: Select a View menu item (Table 3–12) to alter the appearance of the results table.

5. Apply any changes (or dismiss the page) when you finish.

3.8 Using Online Help

At any time while using the Oracle Access Manager 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 Oracle Access Manager 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 Oracle Access Manager Console, click a tab or named node in the navigation tree.
2. Click Help in the upper-right corner of the 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.

3.9 Command-Line Tools

Several command-line tools are available to perform various tasks using the keyboard rather than the console. After using these commands, configurations will be available in the 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 10, "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 Oracle Access Manager Console.

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

3.10 Logging, Auditing, Monitoring Performance

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.

In Oracle Access Manager, Oracle Security Token Service, and Oracle Fusion Middleware, auditing provides a measure of accountability and answers to the "who has done what and when" types of questions.

Oracle Access Manager uses the Oracle Dynamic Monitoring Systems (DMS) to measure application-specific performance information for OAM Servers and registered OAM Agents.

Administrators can monitor performance and log messages for Oracle Access Manager and Oracle Security Token Service using Oracle Fusion Middleware Control.

For more information, see Part VI, "Common Logging, Auditing, Performance Monitoring".

- Chapter 22, "Logging Component Event Messages"
- Chapter 23, "Logging Webgate Event Messages"
- Chapter 24, "Auditing Administrative and Run-time Events"
- Chapter 25, "Monitoring Performance by Using Oracle Access Manager Console"
- Chapter 26, "Monitoring Performance and Logs with Fusion Middleware Control"

4

Managing Services, Certificate Validation, and Common Settings

This chapter explains how to navigate to and configure properties that are common to both Oracle Access Manager and Oracle Security Token Service. This chapter includes the following topics:

- Introduction to Common Configuration Elements
- Enabling or Disabling Available Services
- Managing the Common Settings
- Managing Global Certificate Validation and Revocation

4.1 Prerequisites

Chapter 3, "Getting Started with Common Administration and Navigation"

4.2 Introduction to Common Configuration Elements

This section introduces the common System Configuration elements, shared by all OAM Servers and services in the domain. Figure 4–1 shows the Common Configuration section of the System Configuration navigation tree.

Figure 4–1 Common Configuration Nodes in Navigation Tree

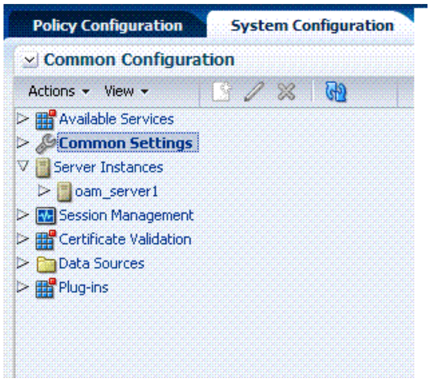


Table 4–1 introduces the common configuration elements that apply to all services in the suite, and where you can find more information on each one.

Table 4–1 Common Configuration Nodes in Navigation Tree

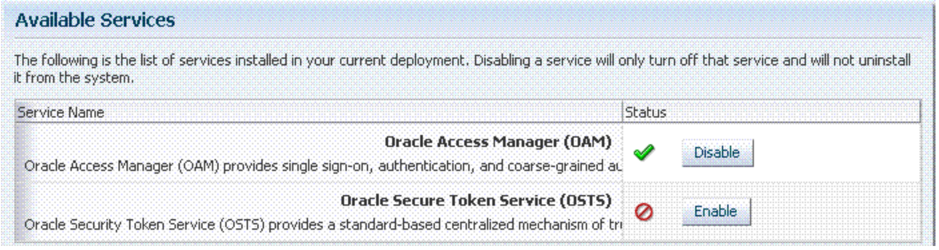
Node	Description
Available Services	Provides access to all services. See: "Enabling or Disabling Available Services" on page 4-2.
Common Settings	Provides properties and settings that apply to all services in the suite, including Session lifetime, Oracle Coherence, Auditing configuration, and Default and System Identity Stores. See: "Managing the Common Settings" on page 4-3.
Server Instances	Provides access to all registered OAM Server instances. See: Chapter 6, "Managing Common OAM Server Registration"
Session Management	Provides access to user session management operations. See: Chapter 7, "Managing Sessions"
Certificate Validation	Provides access to the certificate revocation list and OCSP/CDP settings. See: "Managing Global Certificate Validation and Revocation" on page 4-6.
Data Sources	Provides access to registered user identity stores for Oracle Access Manager and Oracle Security Token Service. See: Chapter 5, "Managing Common Data Sources"
Plug-Ins	Provides access to custom plug-ins to extend authentication functionality for Oracle Access Manager with Oracle Security Token Service. See: Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service

4.3 Enabling or Disabling Available Services

By default, Oracle Security Token Service is disabled at installation time, and must be enabled as described here before using Oracle Security Token Service.

Figure 4–2 shows the Available Services page of the Common Configuration section. This page shows the status of services and provides controls to enable or disable a service.

Figure 4–2 Available Services Page



Available Services

The following is the list of services installed in your current deployment. Disabling a service will only turn off that service and will not uninstall it from the system.

Service Name	Status
Oracle Access Manager (OAM) Oracle Access Manager (OAM) provides single sign-on, authentication, and coarse-grained au	✓
Oracle Secure Token Service (OSTS) Oracle Security Token Service (OSTS) provides a standard-based centralized mechanism of tr	⊘

A green check mark in the Status field beside the service name indicates the service is enabled. A red circle with a line through it indicates that the corresponding service is disabled.

Oracle Access Manager (OAM) must be enabled, whether Oracle Security Token Service is enabled or disabled. Oracle Access Manager does not require Oracle Security

Token Service. However, Oracle Access Manager must be enabled to use Oracle Security Token Service.

Prerequisites

AdminServer must be running.

Logging In to and Signing Out of Oracle Access Manager Console

To enable or disable an available service

1. Log in to the Oracle Access Manager Console, as usual

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

2. From the System Configuration tab, Common Configuration section, click Available Services.
3. **Enable Service**: Click Enable beside the desired service name (or confirm that the Status check mark is green).
4. **Disable Service**: Click Disable beside the desired service name (or confirm that the Status check mark is red).

4.4 Managing the Common Settings

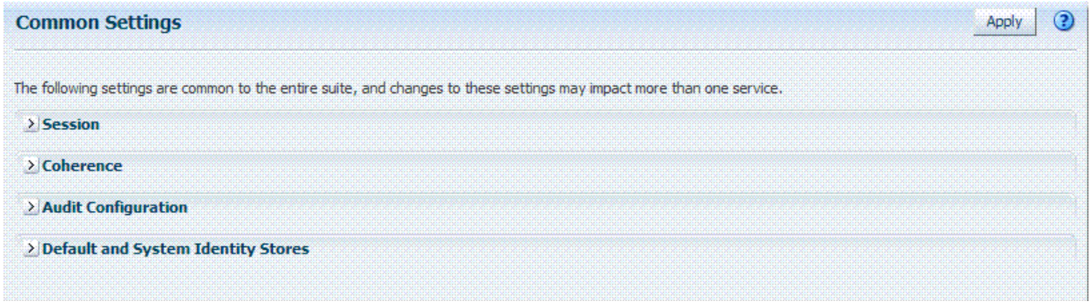
The Common Settings apply to all OAM Server instances and services. This section provides the following topics:

- About Common Settings Pages
- Managing Common Settings

4.4.1 About Common Settings Pages

Common Settings apply to all services within the suite. Figure 4–3 shows the named sections on the Common Settings page, which can be expanded to reveal related elements and values.

Figure 4–3 Common Settings Page (Collapsed View)



Common Settings

The following settings are common to the entire suite, and changes to these settings may impact more than one service.

OAM Administrators can control and specify parameters used by the entire suite, not just a single service, as introduced in Table 4–2.

Table 4–2 Common Settings

Tab Name	Description
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. See Also: "Managing Common Settings" on page 4-4.
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: "Managing Common Settings" on page 4-4.
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. Oracle Access Manager auditing configuration is recorded in `oam-config.xml`. See Also: "Managing Common Settings" on page 4-4.
Default and System Identity Stores	This section identifies the default identity and system stores, which can be one in the same (or different). See Also: "Managing Common Settings" on page 4-4.

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

- Chapter 22, "Logging Component Event Messages"
- Chapter 25, "Monitoring Performance by Using Oracle Access Manager Console"

4.4.2 Managing Common Settings

Users with valid OAM Administrator credentials can perform the following task to display the Common Settings page and perform changes. Included in each main step is a reference to more information elsewhere in this book.

Prerequisites

The OAM Server must be running.

To manage common settings

1. From the System Configuration tab, Common Configuration section, double-click Common Settings in the navigation tree.
2. **Session**:
   a. On the Common Settings page, expand the Session section.
   b. Click the arrow keys beside each list to increase or decrease session lifecycle settings as needed:

      Session Lifetime
      Idle Timeout
      Maximum Number of Sessions per User

   c. Check the box to enable Database Persistence for Active Sessions (or clear it to disable Database Persistence).
   d. Click Apply to submit your changes.

   e. See Also: Chapter 7, "Managing Sessions".

3. **Coherence**: See "Viewing Common Coherence Settings" on page 4-5.
4. **Audit Configuration**:
   a. Open the Audit Configuration section.
   b. In the Audit Configuration section, enter appropriate details for your environment:

      Maximum Log directory size
      Maximum Log file size
      Filter settings to include specific users from the audit (click + button above the Users table and enter a value in the field

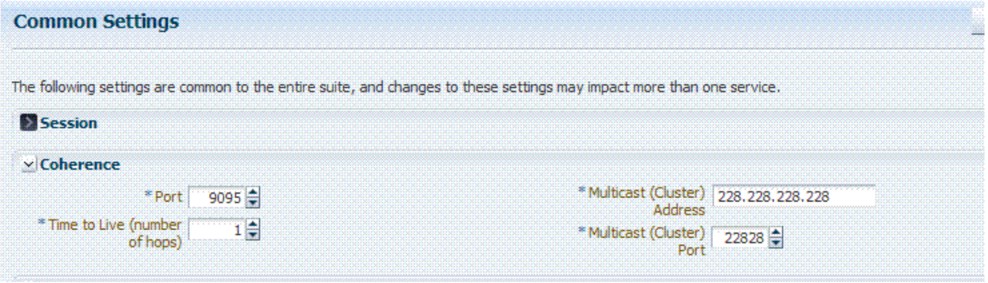
   c. Click Apply to submit the Audit Configuration (or close the page without applying changes).
   d. See Also: Chapter 24, "Auditing Administrative and Run-time Events".
5. **Default and System Identity Stores**:
   a. Expand the Default and System Identity Stores section.
   b. Click the name of the System Store (or Default Store) to display the configuration page.
   c. See "Setting the Default Store and System Store" on page 5-12 for more information.

4.4.3 Viewing Common Coherence Settings

Figure 4–4 shows the Common Settings page with the coherence section expanded.

> **Note:** Oracle strongly recommends that you do not alter these settings without the assistance of Oracle Support.

Figure 4–4 Common Coherence Settings



Common Settings

The following settings are common to the entire suite, and changes to these settings may impact more than one service.

Session

Coherence

* Port 9095

* Time to Live (number of hops) 1

* Multicast (Cluster) Address 228.228.228.228

* Multicast (Cluster) Port 22828

Table 4–3 describes these settings.

Table 4–3 Common Coherence Settings

Element	Description
Port	Value between 1 and 65535 is supported.
Cluster Address	Value between 224.1.255.0 to 239.255.255.255 is allowed.
Time to Live	Value between 0 and 255 is supported.

Table 4–3 (Cont.) Common Coherence Settings

Element	Description
Cluster Port	Value between 1 and 65535 is supported.

To view Common Coherence settings

1. From the System Configuration tab, expand the Common Configurations section, and double-click Common Settings.
2. On the Common Settings page, expand the Coherence section.
3. Close the page when you finish.

4.5 Managing Global Certificate Validation and Revocation

This section provides the following topics:

- About Certificate Validation and Revocation Lists
- Managing Certificate Revocation Lists (CLRs)
- Managing Certificate Validation
- Configuring CDP

4.5.1 About Certificate Validation and Revocation Lists

Oracle Access Manager uses the Online Certificate Status Protocol (OCSP) 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 between the server containing the certificate status and the client application that is informed of that status.

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.

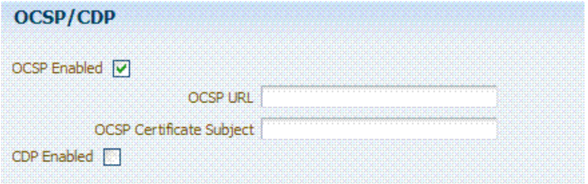
The certificate validation module is used by OSTS to validate X.509 tokens and to verify if needed whether or not the certificates are revoked, by using

- Certificate Revocation Lists (CRLs)
- Online Certificate Status Protocol (OCSP)
- CRL Distribution Point extensions (CDP extensions)

A Certificate Revocation List (CRL) is a common way to maintain access to servers in a network when using a public key infrastructure. The CLR is a list of subscribers paired with their digital certificate status. Revoked certificates are listed with a reason. The dates of certificate issue, and the entities that issued them, are also included. In addition, each list contains a proposed date for the next release. When a potential user attempts to access a server, the server allows or denies access based on the CRL entry for the particular user.

Figure 4–5 shows OCSP/CDP settings for global certificate validation in the console.

Figure 4–5 OCSP/CDP Settings for Global Certificate Validation



OCSP/CDP

OCSP Enabled ☑

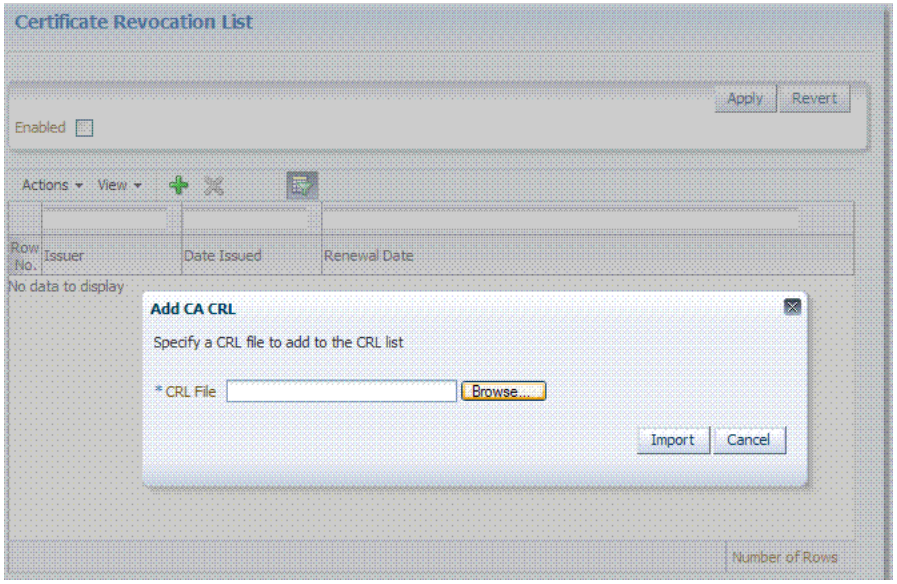
OCSP URL

OCSP Certificate Subject

CDP Enabled □

Figure 4–6 shows adding a CA CRL using the console.

Figure 4–6 Certificate Revocation List Dialog Box



Certificate Revocation List

Enabled □

Row No.	Issuer	Date Issued	Renewal Date

No data to display

Add CA CRL

Specify a CRL file to add to the CRL list

* CRL File

4.5.2 Managing Certificate Revocation Lists (CLRs)

Users with OAM Administrator credentials can use the following procedure to maintain the security of a server and other network resources. This is accomplished by enabling continuous data protection and importing current CA Certificate Revocation Lists.

> **See Also:** "About Certificate Validation and Revocation Lists"

Prerequisites

Have your CA Certificate Revocation List (CA CRL) ready to import.

To manage certificate revocation lists

1. From the Oracle Access Manager Console System Configuration tab, Common Configuration section, select Certificate Validation.
2. Open the Certificate Revocation List node and:
   a. Confirm that the Enabled box is checked.
   b. Add: Click the Add button, browse for the CRL file and select it, click Import.
   c. Remove: Click the name of the list in the table, click the Delete (x) button, and confirm when asked.

   d. Save the configuration.

3. Search for CRLs:

   a. Review the table.

   b. Enable Query by Example and enter the filter strings in the header fields of the table.

4. Proceed to "Managing Certificate Validation".

4.5.3 Managing Certificate Validation

Users with OAM Administrator credentials can use the following procedure to maintain the security of a server and other network resources. This is accomplished by enabling the Online Certificate Status Protocol.

See Also: "About Certificate Validation and Revocation Lists"

Prerequisites

Have your CA Certificate Revocation List (CA CRL) ready to import.

To manage certificate validation

1. From the Oracle Access Manager Console System Configuration tab, Common Configuration section, select Certificate Validation.

2. Open the Certificate Revocation List node:

   a. Confirm that the Enabled box is checked.

   b. Save the configuration.

3. Open the OCSP/CDP node and:

   a. Enable OCSP

   b. Enter the URL of the OCSP Service

   c. Enter the Subject DN of the OCSP Service

   d. Save this configuration.

   e. Proceed to "Configuring CDP".

4.5.4 Configuring CDP

Users with OAM Administrator credentials can use the following procedure to maintain the security of a server and other network resources.

See Also: "About Certificate Validation and Revocation Lists"

To configure CDP

1. From the Oracle Access Manager Console System Configuration tab, Common Configuration section, select Certificate Validation.

2. Open the Certificate Revocation List node:

   a. Confirm that the Enabled box is checked.

   b. Save the configuration.

3. Open the OCSP/CDP node and:

   a. Enable CDP.

b. Save this configuration.

5 Managing Common Data Sources

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

> **Note:** Unless explicitly stated, information in this chapter is the same whether you are using Oracle Access Manager alone or with Oracle Security Token Service.

- Prerequisites
- Introduction to Managing Common Data Sources
- Managing User Identity Stores
- Setting the Default Store and System Store
- Managing the Policy Database by Using the Console

5.1 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 Common Data Sources
- Managing User Identity Stores

5.2 Introduction to Managing Common Data Sources

Various types of data must be managed for Oracle Access Manager with Oracle Security Token Service, as described in following topics:

- About User Identity Stores
- About the Policy and Session Database Store
- About the Oracle Access Manager Configuration Data File
- About Oracle Access Manager Security Keys and the Embedded Java Keystore
- About Oracle Security Token Service Keystores

See Also:

- Chapter 7 for details about session data stored in-memory using Oracle Coherence and propagated to Oracle Database
- Chapter 24 for details about Audit data stored within audit files or a separate Oracle Database (not the policy store)

5.2.1 About User Identity Stores

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 the User Identity Store designated as the System Store is used to authenticate Administrators signing in to use the Oracle Access Manager Console, remote registration, and custom administrative commands in WLST.
- Users attempting to access an OAM-protected resource can be authenticated against any store, not necessarily the only one marked as Default User Identity Store.
- Oracle Security Token Service uses only the Default User Identity Store. When adding User constraints to a Token Issuance Policy, for instance, the identity store from which the users are to be chosen must be Default User Identity Store.

Note: Administrators using the Oracle Access Manager Console must be in the System Store. Users attempting to access an OAM-protected resource can be authenticated against any store, not only the designated Default Store. Oracle Security Token Service uses only the Default User Identity Store

Once you define a remote User Store as the System Store, you must change the OAMAdminConsoleScheme to use an LDAP Authentication Module that references the same remote user store (the System Store).

In the Oracle Access Manager Console, User Identity Store registrations are organized under the Data Sources node (System Configuration tab, Common Configuration section). Administrators can register, view, modify, and delete User Identity Store registrations using either the Oracle Access Manager 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`" seeded as the default administrator.

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

5.2.1.1 Multiple Identity Stores

Administrators can install multiple user identity stores for Oracle Access Manager with Oracle Security Token Service. Each identity store can rely on a different LDAP provider. However, Administrator logins occur against the System Identity Store only. When more than one identity store is registered, administrators must define:

- The System Store for Administrative logins
- The default user store, which comes into play during migration and when using Oracle Security Token Service

External LDAP repositories can provide user, role, and group membership information to be used:

- When evaluating policies during authentication
- When evaluating identities for authorization constraints in a policy
- When using LDAP to search for identities for constraints per authorization policy

Registering user identity stores with Oracle Access Manager is required to provide connectivity with OAM Servers. After registering the identity store, administrators can reference it in one or more authentication modules that form the basis for authentication schemes.

Oracle Access Manager will address each user population and directory as identity domains. Each identity domain simply maps to configured Identity store name.

In the original Oracle Access Manager 11g release, users were identified using a simple user name/id field both internally as well as externally. Support for multiple identity realms requires cross-realm representation of a user or a group or any entity that resides within the identity store. This representation, referred to as a canonical identifier, serves as a unique identifier to various run time and administrative components of Oracle Access Manager:

- **External Representation**: Qualifies the simple user name with identity domain information.

  For instance, in Oracle Access Manager Console a table that lists user names includes a column that displays the identity domain of the respective user. Identity domains map to identity store names. All functional components (the console, Policies, Responses, Logging, Session management, Auditing, and so on) that display user information will begin to qualify the same with the identity domain information

- **Internal Representation**: To support disambiguation, OAM stores and uses the fully-qualified name (or uses both fields, as-is, to form a composite key).

  For instance, The Session Management Engine does this to eliminate the need to store composite). In any case, the fully-qualified name is not visible.

Authorization Policy Administration

Authorization policy administration allows authoring of grants to users or groups. Administrators can search within specific identity stores, selecting certain users or groups and granting or denying them access. Search results provide canonical identifiers for users and groups such that those values are stored as principals of the Identity Constraint component of the Oracle Access Manager Authorization policy. The console displays the names and the Identity Store of origin.

Run Time

Authentication and Authorization relies on the Policy run time component. OAMIdentity is the runtime representation of the authenticated user and any groups that the user is a member of (if any). During policy evaluation, information present within the OAMIdentity is matched with what is stored as part of authorization policy's Identity Constraint. The domain is asserted as a Name Qualifier within the token.

For OAM Proxy, in addition to the existing OAM_REMOTE_USER header, a second OAM_IDENTITY_DOMAIN header is set on every request for an authenticated user, such that a consuming application can disambiguate the user if needed.

Sessions

Session Management searches inform Administrators as to the user Identity Store, which is listed in the search results table

Auditing and Logging

The user Identity Store against which the user has been authenticated is accounted for during auditing and logging.

See Also:

- "About the User Identity Store Registration Page"
- "Managing the Administrators Role" on page 5-14
- "Setting the Default Store and System Store" on page 5-12
- Appendix F, "Introduction to Custom WLST Commands for Administrators"

5.2.2 About the Policy and Session Database 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.

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.

See Also: "Managing the Policy Database by Using the Console" on page 5-16

5.2.3 About the Oracle Access Manager 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 Oracle Access Manager 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 sync 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.

5.2.4 About Oracle Access Manager Security Keys and the Embedded Java Keystore

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

- Every Oracle Access Manager and OSSO Agent has a secret key that other agents cannot read.
- There is 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 5–1 compares the cryptographic keys generated by Oracle Access Manager 11g, 10g, and OSSO 10g, as well as a brief description of there each is stored.

Table 5–1 Oracle Access Manager 11g, 10g, and OSSO 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 keystore is not available through the console and cannot be viewed, managed, or modified.

5.2.5 About Oracle Security Token Service Keystores

Following is a brief summary of several types of keystores for Oracle Access Manager with Oracle Security Token Service:

- System Keystore for keys and certificates associated with OAM Server instances
- Trust Keystore for keys and certificates that are used to establish trust in entities that are interacting with the OAM Server instances
- Partner Keystore for keys and certificates that are used to establish trust with partners, clients, and agents. The partner keys and certificates are stored in .oamkeystore with sensitive information encrypted.
- Certificate Revocation Lists (CRL) are used by the Oracle Access Manager/Oracle Security Token Service server instances when performing CRL-based certificate revocation checking

The following files are distributed across all Managed Servers in the domain by the JMX framework:

- System Keystore: .oamkeystore
- Trust Keystore: amtruststore
- Partner Keystore: .oamkeystore
- CRL: amcrl.jar

Oracle WSM Agent Keystore: Oracle WSM Agent functionality is available to Oracle Security Token Service to publish WS Policies and enforce message protection on inbound and outbound WS messages. Oracle WSM requires a separate Keystore of type JKS to contain System and Partner keys and certificates. The default name is default-keystore.jks, which is specified in jps-config.xml.

Note: Oracle strongly recommends that the Oracle WSM Agent keystore and the OAM/OSTS keystore always be different. Otherwise, keys could be available to any modules authorized by OPSS to access the keystore and Oracle Access Manager keys might be accessed.

See Also: Chapter 18, "Managing Oracle Security Token Service Certificates and Keys"

5.3 Managing User Identity Stores

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
- Registering a New User Identity Store
- Viewing or Editing a User Identity Store Registration
- Deleting a User Identity Store Registration

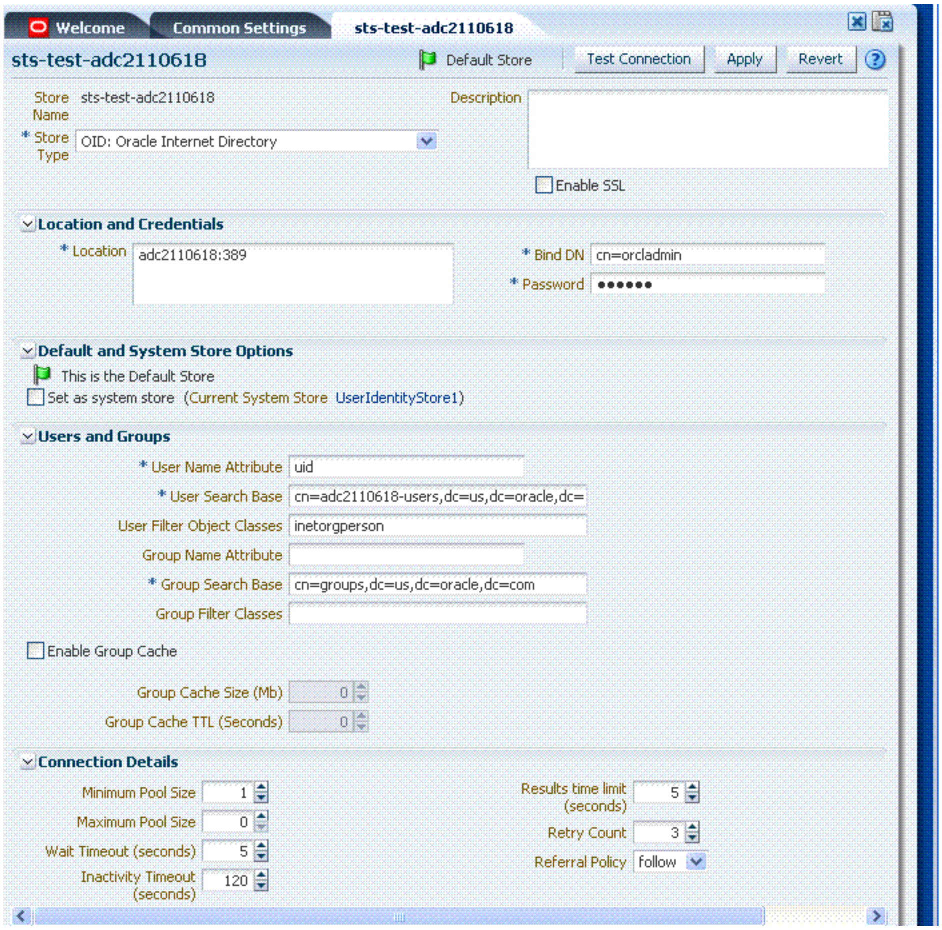
See Also: "Setting the Default Store and System Store" on page 5-12

5.3.1 About the User Identity Store Registration Page

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

Figure 5–1 illustrates a completed registration page. The Create User Identity Store Page is similar, though empty except for default Connection Details values. The Store Type drop-down list provides supported choices.

Figure 5–1 Completed Registration for the Default Store



sts-test-adc2110618

Store Name: sts-test-adc2110618

* Store Type: OID: Oracle Internet Directory

Description:

□ Enable SSL

Location and Credentials

* Location: adc2110618:389

* Bind DN: cn=orcladmin

* Password: ●●●●●●

Default and System Store Options

This is the Default Store

□ Set as system store (Current System Store UserIdentityStore1)

Users and Groups

* User Name Attribute: uid

* User Search Base: cn=adc2110618-users,dc=us,dc=oracle,dc=

User Filter Object Classes: inetorgperson

Group Name Attribute:

* Group Search Base: cn=groups,dc=us,dc=oracle,dc=com

Group Filter Classes:

□ Enable Group Cache

Group Cache Size (Mb): 0

Group Cache TTL (Seconds): 0

Connection Details

Minimum Pool Size: 1

Maximum Pool Size: 0

Wait Timeout (seconds): 5

Inactivity Timeout (seconds): 120

Results time limit (seconds): 5

Retry Count: 3

Referral Policy: follow

Required settings are identified by the asterisk (*) on the page. Table 5–2 describes each element and is organized by element types.

Table 5–2 User Identity Store Elements

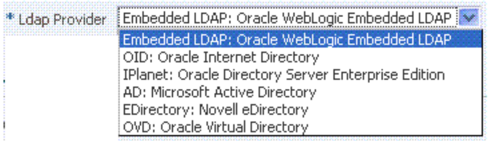
Elements	Description
Store Name	A unique name for this registration. Use up to 30 characters for the name.
Store Type	A list of all supported LDAP providers from which you can choose.  * Ldap Provider: Embedded LDAP: Oracle WebLogic Embedded LDAP Embedded LDAP: Oracle WebLogic Embedded LDAP OID: Oracle Internet Directory IPlanet: Oracle Directory Server Enterprise Edition AD: Microsoft Active Directory EDirectory: Novell eDirectory OVD: Oracle Virtual Directory
Description	Optional.
Enable SSL	Click to check this box and indicate that SSL is enabled between the directory server and OAM Server.
Location and Credentials	

Table 5–2 (Cont.) User Identity Store Elements

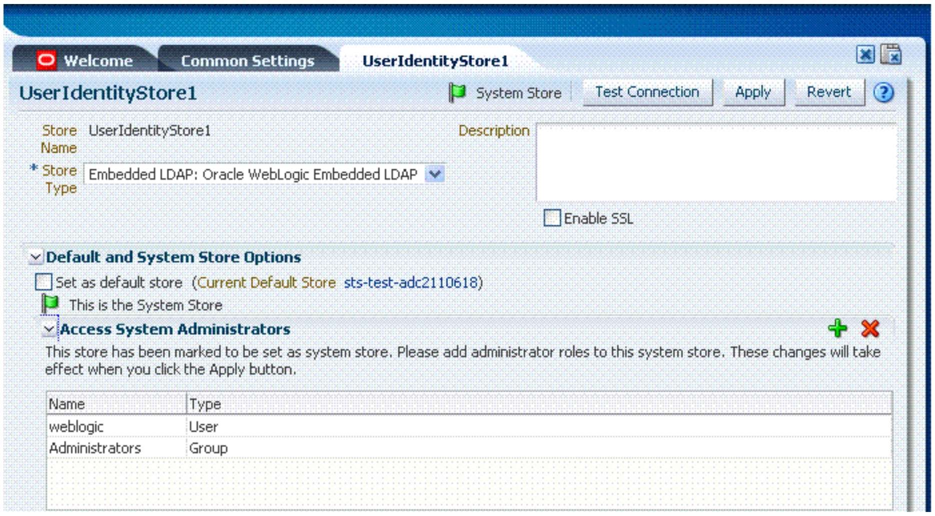
Elements	Description
Location	The URL for the LDAP host, including the port number. Oracle Access Manager 11g has support for multiple LDAP URLs with failover capability. The Identity Assertion Provider fails over to the next LDAP URL based on the order in which these appear. There is no need to specify ldap:// or ldaps://(which supports SSL_NO_AUTH) while specifying the URL value within the Location field. For example, enter: `localhost:7001` **Note**: The number of characters a supported URL can have is based on the browser version. Ensure that your applications do not use URLs that exceed the length that Oracle Access Manager and the browser can handle.
Bind DN	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`
Password	The password of the Principal, which is encrypted for security.
Users and Groups	
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.
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 Classes	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.
Enable Group Cache (size)	Boolean value for group cache: true or false. Default: true
Group Cache Size	Integer for the group cache size. Default: 10000
Group Cache TTL (seconds)	Integer (in seconds) for Time to Live for group cache elements. Default: 0
Connection Details	
Minimum Pool Size	The smallest size set for the connection pool. Default: 10
Maximum Pool Size	The greatest size set for the connection pool. Default: 50

Table 5–2 (Cont.) User Identity Store Elements

Elements	Description
Wait Timeout	The number (in seconds) that connection requests can wait before timing out in the event of a fully utilized pool. Default: 120
Inactivity Timeout	The number (in seconds) that connection requests can be inactive before timing out in the event of a fully utilized pool.
Results Time Limit (seconds)	The time limit (in seconds) for LDAP searches and bind operations on the connection pool. Default: 0
Retry Count	The number of time that the connection is retried when there is a connection failure. Default: 3
Referral Policy	One of these values: ■ follow: Follows referrals during an LDAP search (Default) ■ ignore: Ignores referral entries during an LDAP search ■ throw: Results in a ReferralException, which can be caught by the component user.

Figure 5–2 shows a completed registration page for the System Store, as it looks when viewing it. Notice the Access System Administrators roles are listed. You can add or remove administrator roles only within the defined System Store and the store itself.

Figure 5–2 Completed Registration for System Store



Store Name UserIdentityStore1

* Store Type Embedded LDAP: Oracle WebLogic Embedded LDAP

Description

□ Enable SSL

Default and System Store Options

□ Set as default store (Current Default Store sts-test-adc2110618)

This is the System Store

Access System Administrators

This store has been marked to be set as system store. Please add administrator roles to this system store. These changes will take effect when you click the Apply button.

Name	Type
weblogic	User
Administrators	Group

See Also: Details about classifying users in Chapter 13, "Managing Policies to Protect Resources and Enable SSO"

5.3.2 Registering a New User Identity Store

Users with valid 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. You can also reference it in authorization constraints in access policies.

See Also:

- "About the User Identity Store Registration Page" on page 5-7
- "Setting the Default Store and System Store" on page 5-12

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, Common Configuration section, expand the Data Source nodes.
2. Click User Identity Stores and then click the Create command button in the tool bar.
3. Fill in the form with appropriate values for your deployment (Table 5–2), then click Apply to submit the registration.
4. **Test Connection**: Click the Test Connection button to confirm connectivity, then close the Confirmation window.
5. Close the registration page and proceed as follows:
6. **Add Administrators**: See "Managing the Administrators Role" on page 5-14.
   a. In the navigation tree, double-click the store name to open the registration page.
   b. In the Access System Administrators section, click the + above the table.
   c. Fill in the Add System Administrator Roles dialog box (...).
   d. Click Apply.
7. **Set Default Store:** See "Setting the Default Store and System Store" on page 5-12.
8. Click Apply to submit the registration and close the page.
9. Configure one or more authentication modules to use this store, as described in "Managing Authentication Modules" on page 8-10.

5.3.3 Viewing or Editing a User Identity Store Registration

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

Prerequisites

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

To view or modify a user identity store registration

1. From the System Configuration tab, Common Configuration section, 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 5–2).
5. Click Apply to update the registration (or close the page without applying changes).

6. **Test Connection**: Click Test Connection button to confirm connectivity, then close the Confirmation window.
7. **Set as Default Store**: See "Setting the Default Store and System Store".
8. **Manage Administrator Roles**: See "Managing the Administrators Role".
9. Close the page when you finish.

5.3.4 Deleting a User Identity Store Registration

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

> **Note:** You cannot delete the Default User Identity Store or System 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 within the module).
2. From the System Configuration tab, Common Configuration section, 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 not a Default), 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.

5.4 Setting the Default Store and System Store

This section includes the following topics:

- About Setting the Default Store and System Store
- Defining a Default Store and System Store

5.4.1 About Setting the Default Store and System Store

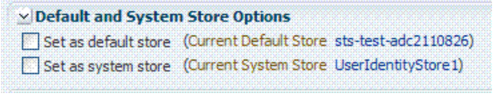
After identity store registration, you can designate the new store as either the Default Store or the System Store, or both:

- Default Store: Used by Oracle Security Token Service, and for migration purposes when patching.
- System Store: Contains Groups and or users for Access System Administrator roles for the entire Identity Management Domain, to which the LDAP Authentication Module used by the OAMAdminConsoleScheme points.

Note: Administrator login works only when the LDAP Authentication Module used by the OAMAdminConsoleScheme also uses the System Store. Changing the System Store impacts the entire identity management domain. If you set another store as a remote store, ensure that the OAMAdminConsoleScheme is also modified to avoid a lockout.

Figure 5–4 shows the Default Store and System Store options that appear immediately after adding a new User Identity Store registration.

Figure 5–3 Fresh Default Store and System Store Options



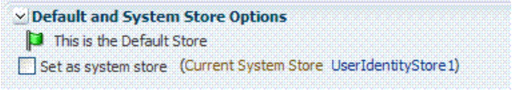
Default and System Store Options

□ Set as default store (Current Default Store sts-test-adc2110826)

□ Set as system store (Current System Store UserIdentityStore1)

Figure 5–4 shows the registration page when you view the Default Store.

Figure 5–4 Default Store Designation



Default and System Store Options

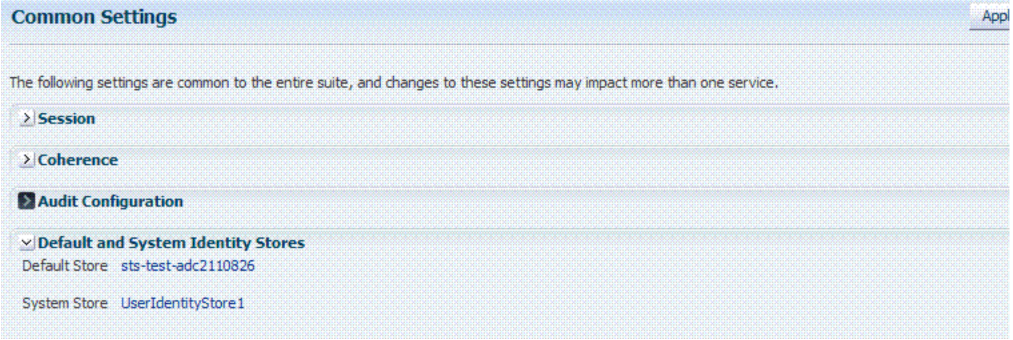
This is the Default Store

□ Set as system store (Current System Store UserIdentityStore1)

As soon as you apply the System Store designation, you are immediately asked to add Access System Administrator roles to it, as described in "Managing the Administrators Role" on page 5-14. Also, you must ensure that the LDAP module used by OAMAdminConsoleScheme refers to the System store.

You can also access the Default and System Identity Stores from the Common Settings page, as shown in Figure 5–5.

Figure 5–5 Common Settings Page: Default and System Identity Stores



Common Settings

The following settings are common to the entire suite, and changes to these settings may impact more than one service.

Default Store sts-test-adc2110826

System Store UserIdentityStore1

5.4.2 Defining a Default Store and System Store

Users with Administrator credentials can use the following procedure to define or change Default Store and System Store designations.

See Also: Managing User Identity Stores

To define a Default Store and System Store

1. From the Oracle Access Manager Console, open the.

   System Configuration tab
     Common Configuration section
       Data Source node
         User Identity Stores node

2. **Set the System Store**: Administrator roles and credentials must reside in this store.

   a. Double-click the name of the store to become the System Store to open the registration page.

   b. Check the box beside Set as system store (for domain wide authentication and authorization operations).

   c. Click Apply to submit the designation.

   d. **Add Administrators**: See "Managing the Administrators Role" on page 5-14.

   e. **Authentication Module**: Set the LDAP Authentication Module used by the OAMAdminConsoleScheme (authentication scheme) to use this System Store. See "Managing Authentication Modules" on page 8-10.

3. **Set Default Store**: This store is for migration purposes only when patching.

   a. Double-click the name of the store to become the Default Store to open the registration page.

   b. Check the box beside Set as default store to set this LDAP store (for migration).

   c. **Authentication Module**: Confirm that the LDAP module for OAMAdminConsoleScheme does not refer to this store. See "Managing Authentication Modules" on page 8-10.

4. Close the registration page and proceed as follows:

5.5 Managing the Administrators Role

This section provides the following topics:

- About Managing the Administrator Role
- Managing Administrator Roles

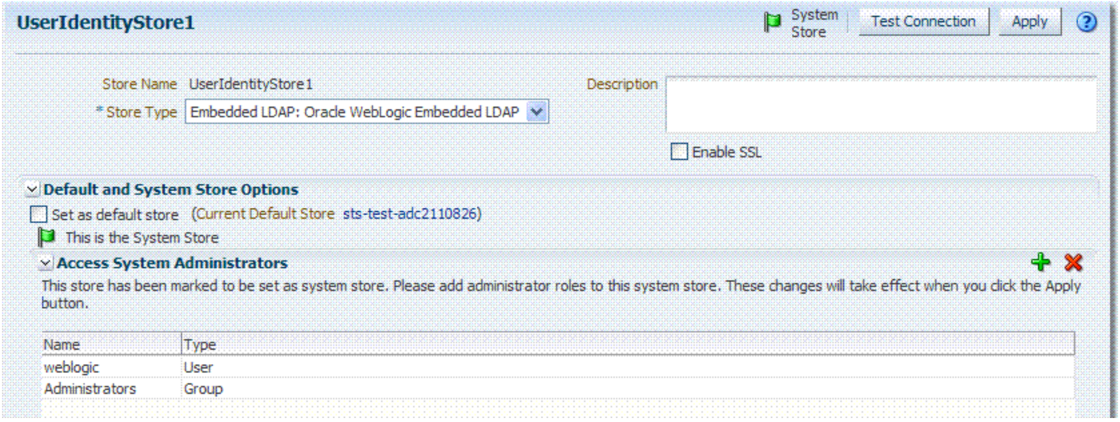
5.5.1 About Managing the Administrator Role

Administrator login works only when the Authentication Scheme (and assigned Authentication Module) used by the IAMSuiteAgent, also uses the System Store.

By default, the Administrators role for Oracle Access Manager with Oracle Security Token Service is the same as the WebLogic Administrators role (Administrators). However, your enterprise might require independent sets of administrators: one set of users responsible for Oracle Access Manager and another for Oracle Security Token Service.

All Administrator roles, users, and groups must be stored in the System Store. If the System Store changes, appropriate Administrator roles must be added to the new System Store. If, when editing an Identity Store registration, you designate a store as the System Store the Access System Administrator section opens on the page as shown in Figure 5–6.

Figure 5–6 System Store Registration with Access System Administrators Section



UserIdentityStore1

Store Name UserIdentityStore1

* Store Type Embedded LDAP: Oracle WebLogic Embedded LDAP

Description

□ Enable SSL

Default and System Store Options

□ Set as default store (Current Default Store sts-test-adc2110826)

This is the System Store

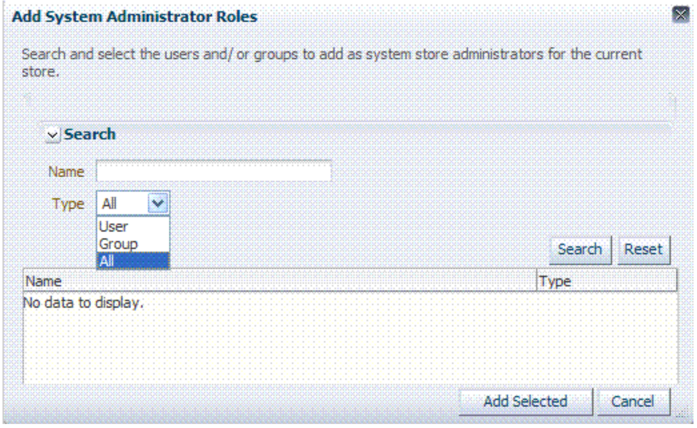
Access System Administrators

This store has been marked to be set as system store. Please add administrator roles to this system store. These changes will take effect when you click the Apply button.

Name	Type
weblogic	User
Administrators	Group

You can add new Administrator roles when adding or editing a User Identity Store registration. Add System Administrator RolesFigure 5–7 shows the page and controls to use.

Figure 5–7 Add System Administrator Roles



Add System Administrator Roles

Search and select the users and/ or groups to add as system store administrators for the current store.

Search

Name

Type All

Name	Type
No data to display.	

5.5.2 Managing Administrator Roles

The following procedure explains how to define or remove administrator roles which must be stored in the User Identity Store designated as the System Store.

Prerequisites

Setting the Default Store and System Store

To add or remove an Administrator role from the System Store

1. In the designated System LDAP Store for OAM:
   a. Define the desired LDAP group to use for Administrators.
   b. Ensure that your Administrators group is available in the group search base.
2. **Locate the System Store Registration**: Perform the following steps (or find a different System Store in the Data Sources node to designate as the System Store).

   a. From the Oracle Access Manager Console, Welcome page, click Common Settings in the Configuration panel.
   b. Scroll and expand the Default and System Identity Stores … section, as needed.
   c. Click the name of the System Store to display the configuration page.
3. **Add User Roles**:
   a. Click the Add (+) button above the Access System Administrators table to display the Add System Administrator Roles dialog box.
   b. Select User in the Type list and click the Search button.
   c. In the results list, click the desired User and then click the Add Selected button.
   d. Repeat as need to add desired administrator User roles.
   e. Click Apply to submit user roles.
4. **Add Group Roles**:
   a. Click the Add (+) button above the Access System Administrators table to display the Add System Administrator Roles dialog box.
   b. Select Group in the Type list and click the Search button.
   c. In the results list, click the desired Group and then click the Add Selected button.
   d. Repeat as need to add desired administrator Group roles.
   e. Click Apply to submit Group roles.
5. **Remove Administrator Roles**:
   a. In the Access System Administrators table, click the row containing the user or group to remove.
   b. Click the Delete (x) button above the table.
   c. Confirm removal when asked.
   d. Click Apply to submit the removal.
6. Correct any authentication modules that use the System Store (if this is a new store), as described in "Managing Authentication Modules" on page 8-10.
7. **Test the New Role**: Close the browser window, then re-open it.
   a. Sign out of the Oracle Access Manager Console and close the browser window.
   b. Start up the Oracle Access Manager Console and attempt to log in using the previous Administrator role to confirm that this attempt fails.
   c. Log in using the new Administrator role to confirm that this attempt is successful.

5.6 Managing the Policy Database by Using the Console

This section includes the following topics:

- About Database Deployment for Oracle Access Manager
- Configuring a Separate Database for Session Data

5.6.1 About Database Deployment for Oracle Access Manager

Oracle requires a single database as the policy store, 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 Oracle Access Manager Console 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 sync operations.

5.6.2 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 Oracle Access Manager Console.

Note: In this rare instance, Oracle recommends that you carefully edit oam-config.xml as described in Step 2f.

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.

6 Managing Common OAM Server Registration

This chapter describes how to provision and manage OAM Server instance registrations using the Oracle Access Manager Console.

The following topics are included:

- Prerequisites
- Introduction to OAM Server Registration and Management
- Managing Individual OAM Server Registrations

6.1 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".

> **Note:** Unless explicitly stated, information is the same whether you are using Oracle Access Manager alone or with Oracle Security Token Service.

6.2 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

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

Table 6–1 summarizes server-side differences between Oracle Access Manager 11g, OAM 10g, and OracleAS SSO 10g (extracted from the overall comparison in Table 1–2).

Table 6–1 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) Oracle Security Token Service runs on OAM Server ■ Oracle Access Manager 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 ■ Oracle Security Token Service	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

6.2.2 About Individual OAM Server Registrations

Administrators can add one or more Managed Servers to the WebLogic Server domain for Oracle Access Manager with Oracle Security Token Service.

When using the WebLogic Configuration Wizard, the OAM Server is automatically registered. However, if the configuration wizard was not used, the OAM Server must be registered manually to open a communication channel.

> **Note:** There is no difference in server registration for Oracle Access Manager or Oracle Security Token Service.

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

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

Only OAM Servers are registered with Oracle Access Manager 11g. The Oracle Access Manager 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 Oracle Access Manager Console, OAM Server registration details within the Oracle Access Manager 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 Console. For more information, see "About the OAM Server Registration Page" on page 6-5.

6.2.3 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 Access Clients) 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 6-6.

> **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 9 and Chapter 10.

6.2.4 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 6-6.

6.2.5 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 6-5.

Simple and Cert modes also require:

- Security passwords that are common to all OAM Servers and Webgates, as described in "Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security" on page 8-5.
- Appropriately signed X.509 digital certificates, as described in Appendix E, "Securing Communication for Oracle Access Manager 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 for Oracle Access Manager 11g"

6.3 Managing Individual OAM Server Registrations

This section describes how to register and manage OAM Server instances using the Oracle Access Manager 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
- Deleting an Individual Server Registration

6.3.1 About the OAM Server Registration Page

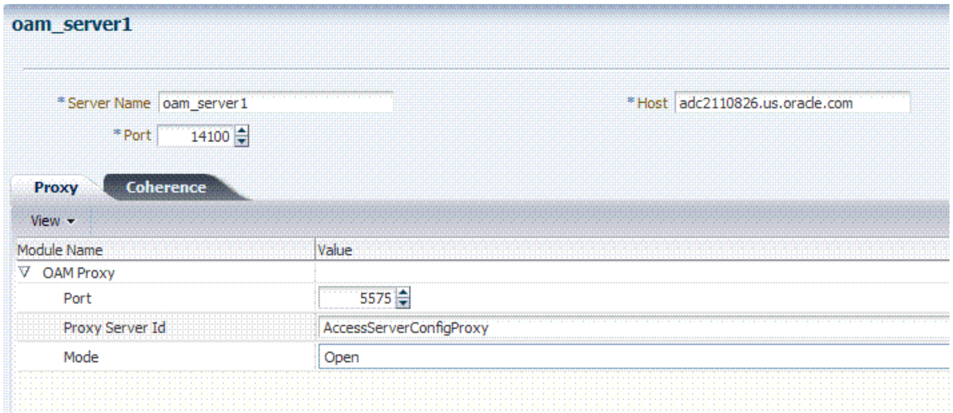
Users with valid Administrator credentials can register a freshly installed Managed Server (OAM Server instance) or modify an existing OAM Server registration using the Oracle Access Manager Console.

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

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

Figure 6–1 illustrates a typical OAM Server registration page when viewed within the Oracle Access Manager Console.

Figure 6–1 OAM Server Registration Page with Proxy Tab Displayed



oam_server1

*Server Name oam_server1

*Host adc2110826.us.oracle.com

*Port 14100

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

This screen illustrates the Server Registration page. The Proxy and Coherence tabs provide additional elements to help configure your environment.

****

Individual server registration settings are described in Table 6–2.

Table 6–2 OAM Server Instance Settings

Element	Definition
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.domain.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 for Oracle Access Manager 11g"
Proxy	See "OAM Proxy Page" on page 6-6
Coherence	See "Coherence Page for Individual Servers" on page 6-7

> **See Also:** "Managing Individual OAM Server Registrations" on page 6-4

6.3.1.1 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 Access Clients, OAM 11g provides authentication and authorization functionality only. Policy modification through Access Clients is not supported.

OAM Proxy settings consist of the details in Table 6–3.

Table 6–3 OAM Proxy Settings for an Individual OAM Server

OAM Proxy Setting	Type	Value
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 6–3 (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 third 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 the Access Protocol for OAM Proxy Simple and Cert Mode Security" on page 8-5. **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 24. However, OAM Proxy uses Apache log4j for logging.

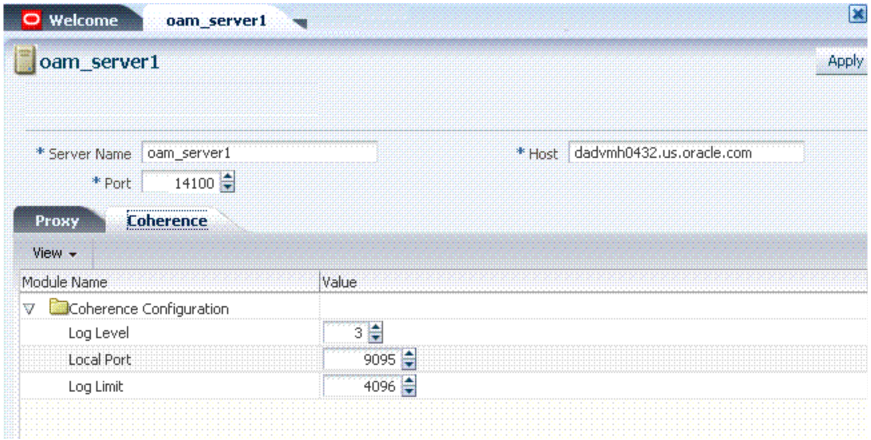
6.3.1.2 Coherence Page for Individual Servers

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 6–2.

Figure 6–2 Coherence Page and Values for an Individual OAM Server



oam_server1

* 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 6–4 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 22.

6.3.2 Registering a Fresh OAM Server Instance

Users with valid Administrator credentials can perform the following task to register a new Managed Server (OAM Server) instance using the Oracle Access Manager 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 6-5

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 Oracle Access Manager Console as usual.
3. From the Server Configuration tab, Common Configuration section, 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 6–2:
   - Server name
   - Host
   - Port
5. Proxy: Enter or select details for this OAM Proxy instance, as described in Table 6–3:
   - 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 Coherence" on page I-22

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.

6.3.3 Viewing or Editing Individual OAM Server and Proxy Settings

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

Changes are immediately visible in the Oracle Access Manager Console and propagated to all OAM Servers in the cluster.

See Also:

- "About the OAM Server Registration Page" on page 6-5
- Appendix F, "Introduction to Custom WLST Commands for Administrators"
- Moving Identity Management to a New Production Environment in the Oracle Fusion Middleware Administrator's Guide

To view or modify a server instance registration

1. From the System Configuration tab, Common Configuration section, 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 OAM Server page, change details for your instance, as described in Table 6–2.
4. **Proxy**: Change details for this OAM Proxy instance, as described in Table 6–3.

   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 Coherence" on page I-22

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

6.3.4 Deleting an Individual Server Registration

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

Prerequisites

Registering a Fresh OAM Server Instance

To delete a server registration

1. From the System Configuration tab, Common Configuration section, 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.

7 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 Operations
- Security

7.1 Prerequisites

The requirements for tasks in this chapter include:

- Reviewing "Introduction to User Sessions and Session Management" on page 7-1
- Getting familiar with Chapter 6, "Managing Common OAM Server Registration"
- Getting familiar with Chapter 3, "Getting Started with Common Administration and Navigation"

7.2 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 7-3, 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 7-4.

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 7-6.

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 7-8.

Oracle Access Manager 11g uses Tangosol Coherence to replicate session states within a distributed installation. Coherence is used to communicate state changes between the Oracle Access Manager Console and OAM Servers. Coherence relies on User Datagram Protocol (UDP) for cluster discovery and heartbeat. If a firewall exists between certain components of OAM 11g, then the corresponding UDP ports used by Coherence must be open. Otherwise, OAM 11g might not work correctly. For more information, see "Using Coherence" on page I-22.

7.2.1 About the User Session Lifecycle

User session lifecycle settings can be defined using the Oracle Access Manager 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 11-16.

User session lifecycle settings for OAM Agents can be defined using the Oracle Access Manager Console. The WebLogic Scripting Tool does not include options for session management.

7.2.2 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.

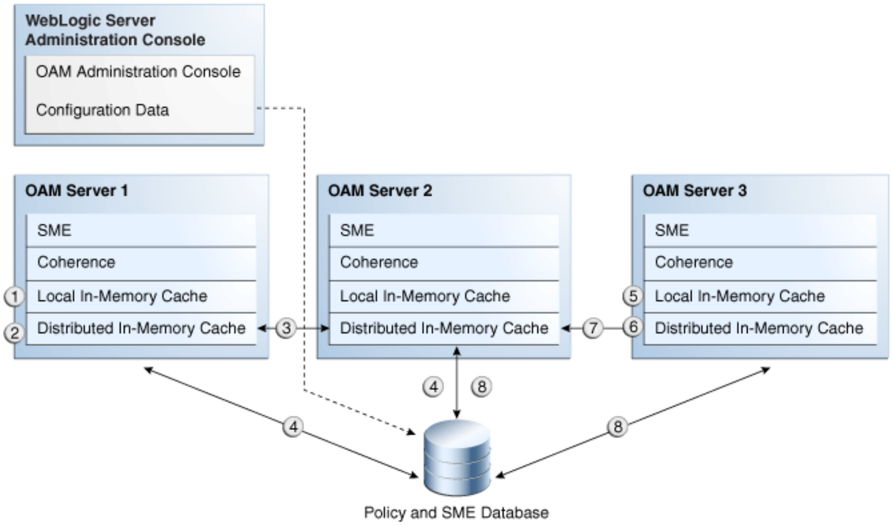
Note: Oracle Coherence traffic is automatically encrypted.

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 7–1 illustrates the storage of session data that occurs using embedded Oracle Coherence. A description follows the diagram.

Note: The Oracle Access Manager 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 Oracle Access Manager Console, an OAM Server must be running.

Figure 7–1 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.

7.3 Configuring User Session Lifecycle Settings

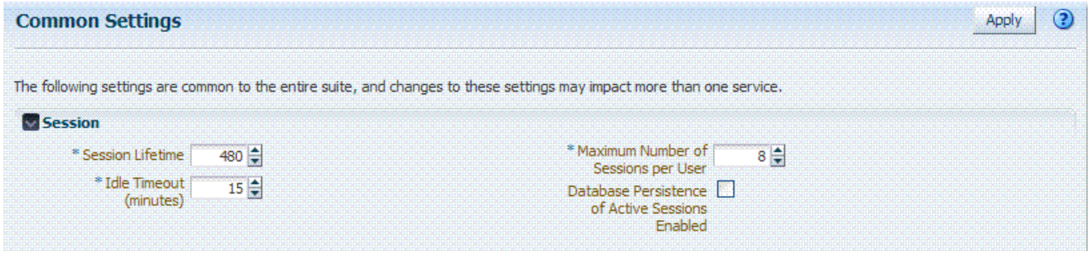
This section provides the following topics:

- About Common Session Lifecycle Setting Page
- Viewing or Modifying Common Session Lifecycle Settings

7.3.1 About Common Session Lifecycle Setting Page

User-session lifecycle settings are part of the Common Settings shared by all OAM Servers. Figure 7–2 shows the lifecycle attributes that you can configure on the Common Settings page.

Figure 7–2 Session Details: Common Settings Page



The following settings are common to the entire suite, and changes to these settings may impact more than one service.

* Session Lifetime 480

* Idle Timeout (minutes) 15

* Maximum Number of Sessions per User 8

Database Persistence of Active Sessions Enabled □

Table 7–1 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 7–1 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. Any value between -2147483648 and 2147483647 is allowed. 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. Any value between -2147483648 and 2147483647 is allowed. 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. Any value between 0 and 2147483647 is allowed.
Database Persistence for Active Sessions Enabled	Persists active sessions to the configured database session store, in addition to the local and distributed caches. Sessions are retained even if all managed servers die off. Default = Enabled (checked) If this is overkill for your environment, or you want to perform deployment sizing to take into account the database, you can clear the checkbox and restart all OAM Servers to disable this function.

7.3.2 Viewing or Modifying Common Session Lifecycle Settings

Users with valid Administrator credentials can use the following procedure to modify common session lifecycle settings using the Oracle Access Manager Console.

> **See Also:** "About Common Session Lifecycle Setting Page" on page 7-6

To view or modify common session lifecycle settings

1. From the System Configuration tab, expand the Common Configurations section, and double-click Common Settings.
2. On the Common Settings page, expand the Session section.
3. Click the arrow keys beside each list to increase or decrease session lifecycle settings as needed (Table 7–1):
   - Session Lifetime
   - Idle Timeout
   - Maximum Number of Sessions per User
4. Check the box to enable Database Persistence for Active Sessions.
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".

7.4 Managing Active User Sessions

The Session Management page provides Search controls that enable Administrators to create a query based on filter conditions, save their Search Criteria for use later, and add fields to the query form to further refine the search.

In the database store configuration, the session might exist in the database but not in the cache. Session searches are based on the system time stamp. The database is queried for sessions updated earlier than the time stamp (minus the write delay). The cache is queried for sessions updated later than this time stamp. Resulting data found in the cache and the database is merged. If duplicate results exist, cache data prevails. Detailed performance metrics are generated for search operations.

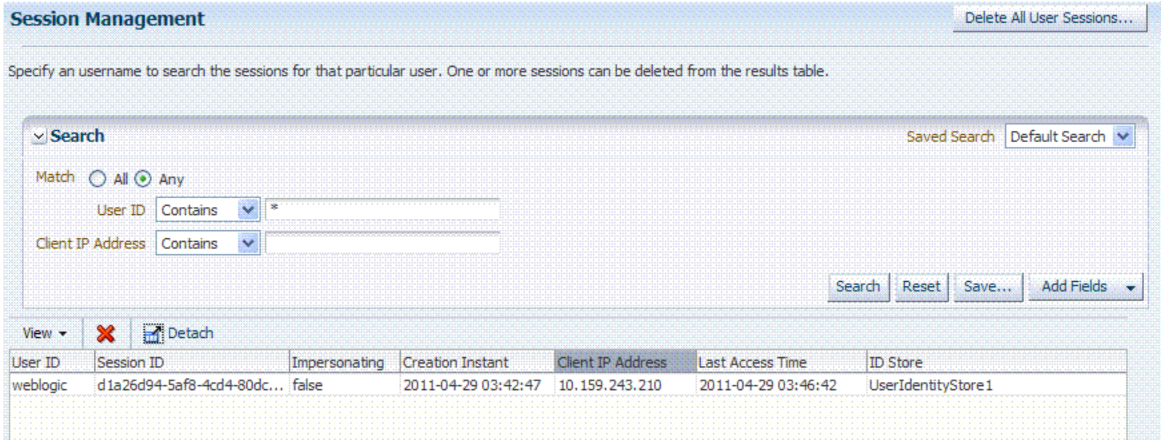
This section describes how to locate and delete one or more sessions for a single user, or for all users. It provides the following information:

- About the Session Management Page
- Managing Active User Sessions

7.4.1 About the Session Management Page

Figure 7–3 illustrates the Session Management page, under the System Configuration tab, Common Configuration section. Additional details follow the figure.

Figure 7–3 Common Configuration: Session Management Page



Specify an username to search the sessions for that particular user. One or more sessions can be deleted from the results table.

Match ○ All ● Any

User ID Contains *

Client IP Address Contains

User ID	Session ID	Impersonating	Creation Instant	Client IP Address	Last Access Time	ID Store
weblogic	d1a26d94-5af8-4cd4-80dc...	false	2011-04-29 03:42:47	10.159.243.210	2011-04-29 03:46:42	UserIdentityStore1

Table 7–2 describes Session Management page and Search controls that enable you to create a query that is based on filter conditions.

Table 7–2 Session Management Controls and the Results Table

Name	Description
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.

Table 7–2 (Cont.) Session Management Controls and the Results Table

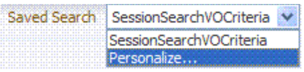
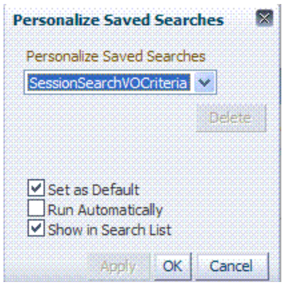
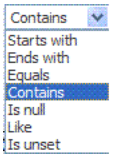
Name	Description
Saved Search	Lists any search criteria saved previously for reuse. A list like the following is made available whenever you save search criteria.  Saved Search SessionSearchVOCriteria SessionSearchVOCriteria Personalize... If you choose Personalize, you can change the behavior of the saved search criteria by making new choices in the following window.  Personalize Saved Searches Personalize Saved Searches SessionSearchVOCriteria Delete Apply OK Cancel ☑ Set as Default □ Run Automatically ☑ Show in Search List
Match All Any	Enables you to match either any of the criteria you have specified or match all of the criteria you have specified during the search. **Note**: When a resource is protected by AnonymousScheme, it is not displayed in a session search.
Userid	Enter a specific userID in the field and then click the Search button to display all active sessions for this user. Incomplete strings and wild cards are allowed. The following list is available to assist your search:  Contains Starts with Ends with Equals Contains Is null Like Is unset
Client IP Address	Enter a Client IP Address and then click the Search button to display all active sessions for this user. Incomplete strings and wild cards are allowed. The same list is available to assist your Userid search and your Client IP Address.
Search	Click this button to initiate a search based on criteria in the form.
Reset	Click this button to clear the form of all criteria.

Table 7–2 (Cont.) Session Management Controls and the Results Table

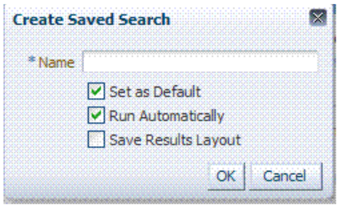
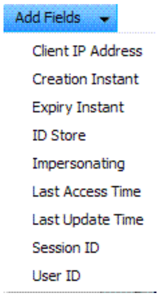
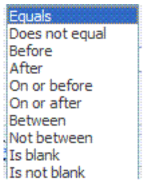
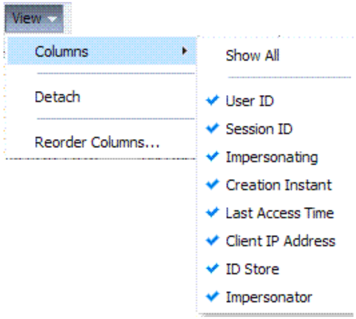
Name	Description
Save	Click this button to initiate a save operation that enables reuse of your search criteria. The following window opens.  Create Saved Search * Name ☑ Set as Default ☑ Run Automatically □ Save Results Layout 1. Enter a name, which will appear in the Saved Search list for later selection. 2. Set this search as the Default (or clear the check box). 3. Set this search to Run Automatically (or clear the check box). 4. Save the Results Layout (or clear the check box). 5. Click OK.
Add Fields	You can add different fields to your search form. The following list is available to assist.  Add Fields Client IP Address Creation Instant Expiry Instant ID Store Impersonating Last Access Time Last Update Time Session ID User ID 1. Click the Add Fields button. 2. Click items in the list to add them to the form and click Save. After adding an item, notice that a list is available to assist with the search. For example: Employment and time-based selections provide the following list.  Equals Does not equal Before After On or before On or after Between Not between Is blank Is not blank

Table 7–2 (Cont.) Session Management Controls and the Results Table

Name	Description
View	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:  View Columns Detach Reorder Columns... Show All ✓ User ID ✓ Session ID ✓ Impersonating ✓ Creation Instant ✓ Last Access Time ✓ Client IP Address ✓ ID Store ✓ Impersonator ■ 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.
✖ Delete	Choose this command button after selecting items in the results table to delete. **Note**: When session search criteria is generic (using just a wild card (*), for example), there is a limitation on deleting a session from a large list of sessions. Oracle recommends that your session search criteria is fine-grained enough to obtain a relatively small set of results (ideally 20 or less). **Also**: 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.
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. ■ User ID: ■ Impersonating: ■ Creation Time: The day and time the session was created. ■ Last Accessed: The day and time the session was last accessed ■ Client IP: The IP address of the specified user. ■ ID Store ■ Impersonator

7.4.2 Managing Active User Sessions

Users with valid 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.

When a resource is protected by AnonymousScheme, it is not displayed in a session search.

> **See Also:** "About the Session Management Page" on page 7-8

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 System Configuration tab, Common Configuration section, open the Session Management node.

   The Session Management Search page appears with the Username field and a results table.

2. **Add Fields**: From the Add Fields list, choose the desired field name (Table 7–2).
3. **Choose Operators**: Open the list of operators for the chosen search field, and choose the desired function.
4. **Find sessions**:
   a. In the desired query field, enter your criteria (with or without a wild card (*)).
   b. Click the Search button to locate sessions that match either any or all your criteria.
   c. Review the results table.
   d. Repeat if needed to further refine your search.
5. **Configure the Results Table**: Use functions on the View menu to create the desired results table.
6. **Delete sessions**:
   a. In the results table, click one or more sessions to remove.
   b. Click the Delete (x) 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 Operations".

7.5 Verifying Session Management Operations

Use the following procedure to verify session management operations.

To validate session management

1. Access a resource from a browser.

2. In the Oracle Access Manager Console, verify that a user session exists, as described in "Managing Active User Sessions" on page 7-11.
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 Oracle Access Manager Console, delete all user sessions, (Step 7 of "Managing Active User Sessions" on page 7-11) 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.

7.6 Security

This section discusses session security for Oracle Access Manager 11g:

- Secure HTTPS Protocol
- Coherence
- Database Persistence

7.6.1 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.

7.6.2 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.

7.6.3 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.

Database Persistence for Active Sessions Enabled appears in the Oracle Access Manager Console, as described in Table 7–1.

Part III

Oracle Access Manager Settings Management

Part III provides information about managing low-level Oracle Access Manager configuration.

Part III contains the following chapters:

- Chapter 8, "Configuring Access Manager Settings"
- Chapter 9, "Registering Partners (Agents and Applications) by Using the Console"
- Chapter 10, "Registering Partners (Agents and Applications) Remotely"

8

Configuring Access Manager Settings

This chapter describes Access Manager-specific settings. It provides the following topics:

- Prerequisites
- Introduction to Access Manager Settings
- Managing Access Manager Load Balancing and Secure Error Modes
- Managing SSO Tokens and IP Validation
- Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security
- Managing Run Time Policy Evaluation Caches
- Managing Authentication Modules
- Creating Custom Authentication Modules

8.1 Prerequisites

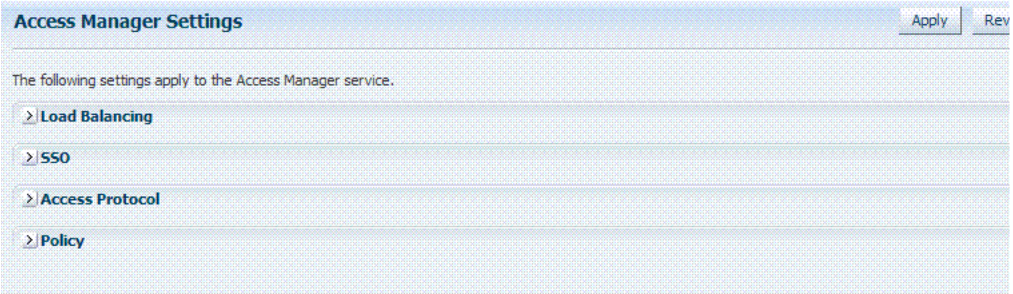
This section identifies requirements for tasks in this chapter. Before you begin tasks in this chapter, be sure to review the following topics:

- Chapter 3, "Getting Started with Common Administration and Navigation"
- Chapter 6, "Managing Common OAM Server Registration"

8.2 Introduction to Access Manager Settings

The Access Manager Setting section of the System Configuration tab provides a number of settings specific to Access Manager service operations.

Figure 8–1 Access Manager Settings



Access Manager Settings

The following settings apply to the Access Manager service.

Table 8–1 Access Manager Settings

Setting	Described in ...
Load Balancing	Managing Access Manager Load Balancing and Secure Error Modes
SSO	Managing SSO Tokens and IP Validation
Access Protocol	Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security
Policy	Managing Run Time Policy Evaluation Caches

8.3 Managing Access Manager Load Balancing and Secure Error Modes

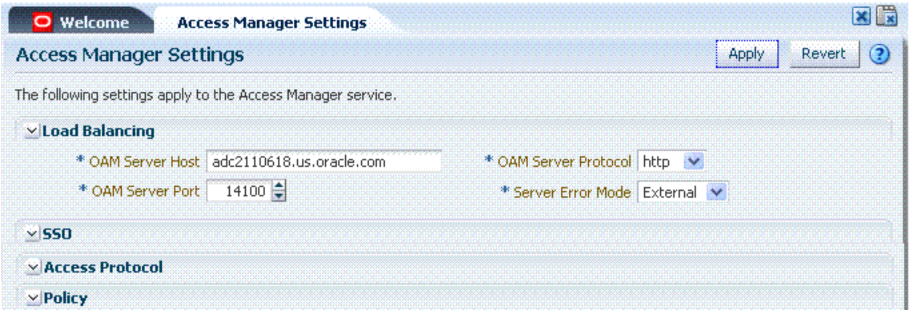
This section provides the following topics:

- About Access Manager Load Balancing Settings and Secure Error Modes
- Managing OAM Server Load Balancing and Secure Error Modes

8.3.1 About Access Manager Load Balancing Settings and Secure Error Modes

Figure 8–3 shows the Load Balancing Settings section of the Access Manager Settings page. These were previously part of the SSO Engine settings. SSO Engine is the controller for user sessions; settings are global and common to all OAM Servers in the WebLogic administration domain. Table 8–4 describes each element and how it is used.

Figure 8–2 Access Manager Settings: Load Balancer



Access Manager Settings

The following settings apply to the Access Manager service.

Load Balancing

* OAM Server Host: adc2110618.us.oracle.com

* OAM Server Port: 14100

* OAM Server Protocol: http

* Server Error Mode: External

Table 8–2 Access Manager Settings: Load Balancer

Element	Description
OAM Server Host	The name of the computer on which OAM is installed.
OAM Server Port	The port on which the host is listening. Value between 1 and 65535 is supported
OAM Server Protocol	Either HTTP or HTTPS. See Also: "About Security Modes and X509Scheme Authentication" on page E-3

Table 8–2 (Cont.) Access Manager Settings: Load Balancer

Element	Description
Server Error Mode	The setting you choose determines the nature of error messages and error codes returned by the OAM Server when an operation fails (because of an invalid username or password, for example, or a server error (connection to the LDAP Server is down)). Choose one of the following settings to configure error messages with varying degrees of security for your custom login pages: ■ SECURE: Most secure. Provides generic error messages that barely give any hint of the internal reason for the error. ■ EXTERNAL: Recommended level. ■ INTERNAL: Least secure level. ■ OSSO10g: Compatible with OSSO 10g. Might be required in upgraded environments for consistency.

Table 8–3 identifies the error codes, trigger conditions, and recommended messages. These error codes are based on the Server Error Mode and are not exposed.

Table 8–3 External Error Codes, Trigger Conditions, and Recommended Messages

External Error Code	Trigger Condition	Recommended Display Message
OAM-1	Invalid login attempts less than the allowed count.	An incorrect Username or Password was specified
OAM-2	Invalid login attempts less than the allowed count.	An incorrect Username or Password was specified
OAM-3	Processing submitted credentials fails for some reason. For example: in WNA mode, the SPENGO token is not received.	Internal Error.
OAM-4	An authentication exception is raised for some reason.	System error. Please contact the System Administrator.
OAM-5	The user account gets locked because of certain conditions (exceeded invalid attempts, for instance). OIM Integration. The Error page appears with contact details after the password is validated.	The user account is locked or disabled. Please contact the System Administrator.
OAM-5	The user account gets locked because of certain conditions (exceeded invalid attempts, for instance). OID Without OIM Integration: The Error page appears with contact details after the password is validated.	The user account is locked or disabled. Please contact the System Administrator.
OAM-5	The user account is disabled.	The user account is locked or disabled. Please contact the System Administrator.
OAM-6	The user has exceeded the maximum number of allowed sessions, which is a configurable attribute.	The user has already reached the maximum allowed number of sessions. Please close one of the existing sessions before trying to login again.

Table 8–3 (Cont.) External Error Codes, Trigger Conditions, and Recommended

External Error Code	Trigger Condition	Recommended Display Message
OAM-7	Failure could be due to multiple reasons; the exact reason is not propagated to the user level for security reasons. For instance: ▪ The request ID could have been lost ▪ The certificate is not retrieved correctly The default error message is displayed when no other specific messages are propagated up.	System error. Please re-try your action. If you continue to get this error, please contact the Administrator.

8.3.2 Managing OAM Server Load Balancing and Secure Error Modes

Users with valid Administrator credentials can perform the following task to modify Access Manager load balancing settings using the Oracle Access Manager Console.

> **See Also:** "About Access Manager Load Balancing Settings and Secure Error Modes" on page 8-2

To view or edit common load balancing specifications

1. From the System Configuration tab, Access Manager Settings section, open the Access Manager Settings to display the page.
2. On the Access Manager Settings page, expand the load balancing section:
   - View Only: Close the page when you finish.
   - Modify: Perform remaining steps to edit the configuration.
3. Edit settings as needed for your deployment, based on details in Table 8–2.
4. Click Apply to submit the changes (or close the page without applying changes).
5. Dismiss the Confirmation window.
6. Proceed to "Managing SSO Tokens and IP Validation".

8.4 Managing SSO Tokens and IP Validation

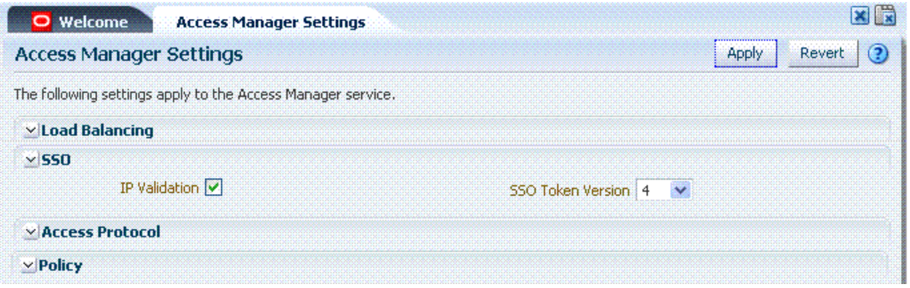
This section provides the following topics:

- About Access Manager SSO Tokens and IP Validation Settings
- Managing SSO Tokens and IP Validation

8.4.1 About Access Manager SSO Tokens and IP Validation Settings

Figure 8–3 shows the single-sign on (SSO) portion of the Access Manager Settings page. Table 8–4 describes each element and how it is used.

Figure 8–3 Access Manager Settings: SSO



The following settings apply to the Access Manager service.

IP Validation ☑

SSO Token Version 4

Table 8–4 Access Manager Settings: SSO

Element	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. Check the box to enable IP Validation. Clear the box the disable IP Validation.
SSO Token Version	Select your SSO token version from the list.

8.4.2 Managing SSO Tokens and IP Validation

Users with valid Administrator credentials can perform the following task to modify Access Manager load balancing settings using the Oracle Access Manager Console.

> **See Also:** "About Access Manager Load Balancing Settings and Secure Error Modes" on page 8-2

To view or edit Access Manager SSO specifications

1. From the System Configuration tab, Access Manager Settings section, open Access Manager Settings to display the page.
2. On the Access Manager Settings page, expand the SSO section:
   - View Only: Close the page when you finish.
   - Modify: Perform remaining steps to edit the configuration.
3. Edit settings as needed for your deployment, based on details in Table 8–4.
4. Click Apply to submit the changes (or close the page without applying changes).
5. Dismiss the Confirmation window.
6. Proceed to "Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security".

8.5 Managing the Access Protocol for 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

8.5.1 About Simple and Cert Mode Transport Security

Table 8–5 outlines the similarities between Simple and Cert modes.

> **See Also:** Appendix E, "Securing Communication for Oracle Access Manager 11g"

Table 8–5 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 ▪ Key KEYSTOREStore Alias Password	N/A
During OAM Agent or OAM Server registration, the communication mode is propagated to the Oracle Access Manager 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 in Cert Security mode (see Table 9–4).	N/A

Table 8–5 (Cont.) Summary: Simple and Cert Mode

Artifact or Process	Simple Mode	Cert Mode	Open Mode
During Agent registration, ObAccessClient.xml is generated in: $DOMAIN_HOME/output/$*Agent_Name*/	ObAccessClient.xml **Copy to**: **11g Webgate**: *11gWebgate_instance_dir*/config/OHS/ohs1/webgate/config If: 11gWebgate_instance_dir=Oracle_Home/instance/instance1 **10g Webgate**: $*Webgate_install_dir*/oblix/lib	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

8.5.2 About the Common OAM Proxy Page for Secure Server Communications

Table 8–6 describes the settings required for Simple or Cert mode configurations.

Table 8–6 Server Common OAM Proxy Secure Communication Settings

Mode	Description
Simple Mode Configuration	The global passphrase for 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 Key KEYSTOREStore where the Cert mode X.509 certificates signed by an outside Certificate Authority reside: ▪ PEM Keystore Alias ▪ PEM 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 them, as described in"Viewing or Editing Simple or Cert Settings for OAM Proxy".

8.5.3 Viewing or Editing Simple or Cert Settings for OAM Proxy

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

See Also:

- "Registering and Managing OAM Agents Using the Console" on page 9-10
- Appendix E, "Securing Communication for Oracle Access Manager 11g"

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

1. From the System Configuration tab, Access Manager Settings section, open the Access Manager Settings page.
2. Expand the Access Protocol section of the page, if needed.
3. Simple Mode: Add or alter a Global Passphrase if you are using OAM-signed X.509 certificates.
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).
6. Update Agent registration pages as needed to regenerate artifacts, and then replace the earlier artifacts as described in Chapter 9 or Chapter 10.

8.6 Managing Run Time Policy Evaluation Caches

This section explains:

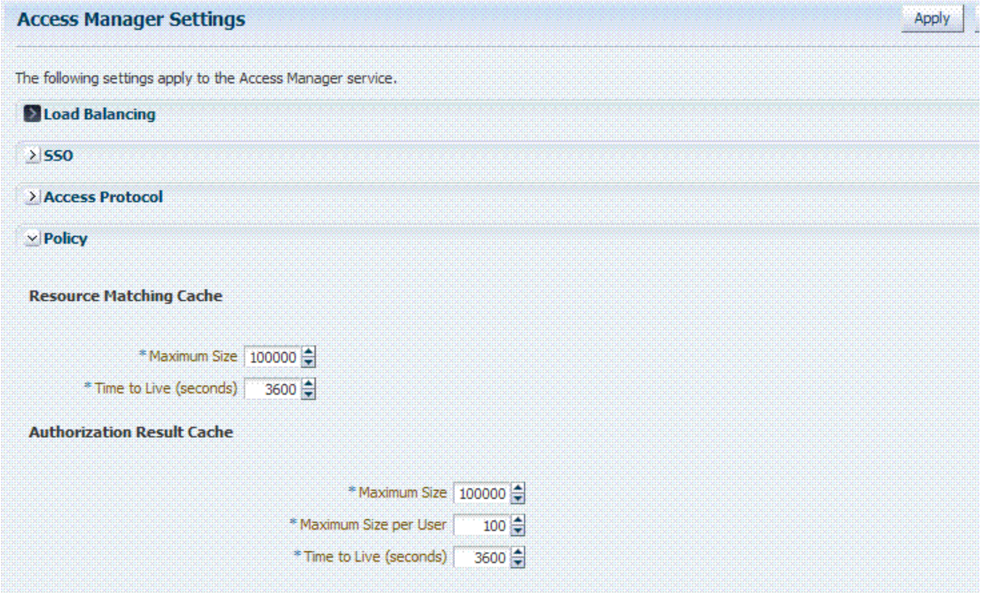
- About Run Time Policy Evaluation Caches
- Managing Run Time Policy Evaluation Caches

See Also: "About Run Time Resource Evaluation" on page 13-17

8.6.1 About Run Time Policy Evaluation Caches

Figure 8–4 illustrates the Policy section of the Access Manager Settings page. This section provides settings for the Resource Matching Cache and the Authorization Result Cache, which come into play during policy evaluation at run time.

Figure 8–4 Common Policy Evaluation Caches



Access Manager Settings

The following settings apply to the Access Manager service.

Load Balancing

SSO

Access Protocol

Policy

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 8–7 outlines these global settings that apply to all servers and requests.

Table 8–7 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 See Also: "Tuning 10g and 11g Webgate Caches" on page 9-31

8.6.2 Managing Run Time Policy Evaluation Caches

Administrators can use this procedure to manage the Access Manager policy evaluation caches.

> **See Also:** "Changing the Request Cache Type in a High Availability Environment" on page F-8

To manage common run time policy evaluation cache settings

1. From the System Configuration tab, Access Manager Settings section, double-click Access Manager Settings to open the page.
2. On the Access Manager Settings page, expand the Policy section.
3. **Resource Matching Cache**: Specify details and click apply (Table 8–7).
4. **Authorization Result Cache**: Specify details and click apply (Table 8–7).

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

8.7 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 and Pages
- Creating a New Authentication Module of an Existing Type
- Viewing or Editing Authentication Modules
- Deleting an Authentication Module

8.7.1 About Default Authentication Modules and Pages

In the Oracle Access Manager 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

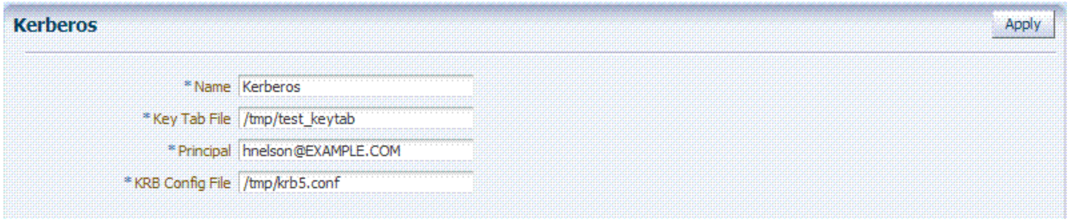
See Also:

- "About the Custom Authentication Module Plug-ins" on page 8-18
- "About Challenge Methods" on page 12-20
- "About Authentication Modules" on page 12-24
- Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service for details about custom authentication modules and plug-ins

8.7.1.1 Kerberos Authentication Module

The pre-configured Kerberos authentication module is illustrated in Figure 8–5. Additional details follow the figure.

Figure 8–5 Pre-configured Kerberos Authentication Module



* Name	Kerberos
* Key Tab File	/tmp/test_keytab
* Principal	hnelson@EXAMPLE.COM
* KRB Config File	/tmp/krb5.conf

Table 8–8 describes the definition of the Kerberos authentication module. You can use the existing, pre-configured Kerberos authentication module or create one of your own.

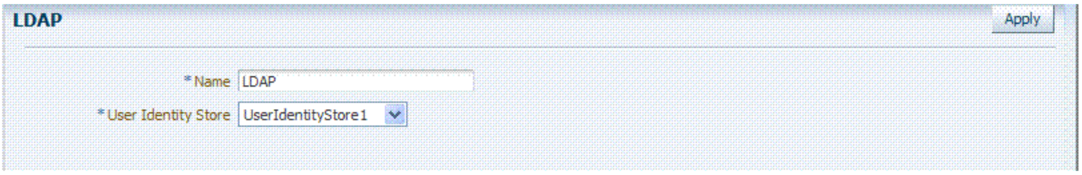
Table 8–8 Kerberos Authentication Module Definition

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).

8.7.1.2 LDAP Authentication Modules

The pre-configured LDAP authentication module is illustrated in Figure 8–5. Additional details follow the figure.

Figure 8–6 Pre-Configured LDAP Authentication Module



LDAP

Apply

* Name LDAP

* User Identity Store UserIdentityStore1

Table 8–9 describes the elements in an LDAP authentication module. The same elements and values are also used in LDAPNoPasswordAuthnModule.

Table 8–9 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 Stores" on page 5-7. Upon installation, there is only one User Identity Store and it is also the System Store. If you add more identity stores and designate a different store as the System Store, be sure to change the LDAP module to point to the System Store. OAMAdminConsoleScheme (authentication scheme) relies on the LDAP module for Administrator Roles and credentials. If you change See Also: "Setting the Default Store and System Store" on page 5-12.

8.7.1.3 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.

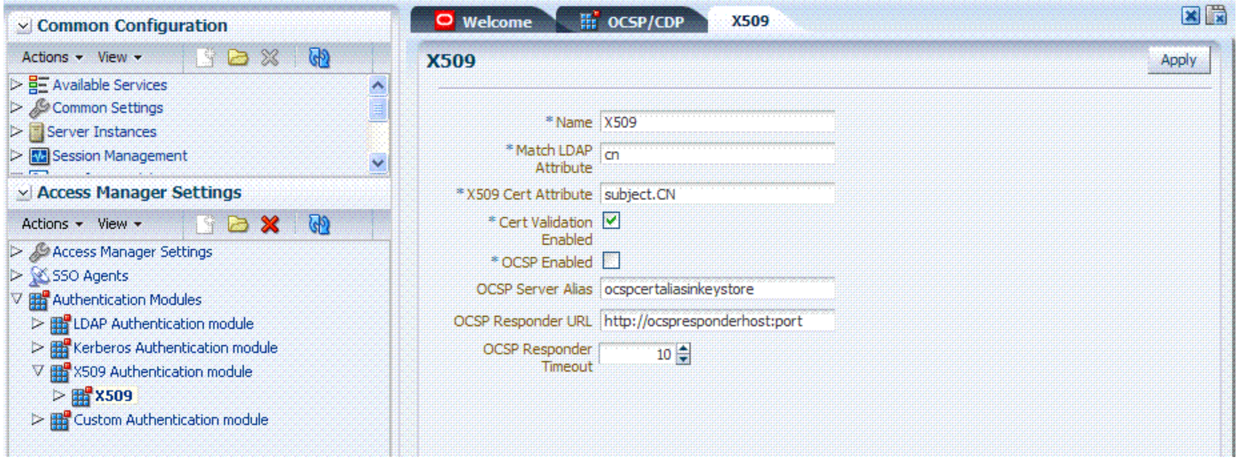
Oracle Access Manager uses the Online Certificate Status Protocol (OCSP) Internet protocol to maintain the security of a server and other network resources. OCSP is used for obtaining the revocation status of an X.509 digital certificate. OCSP specifies the communication syntax used between the server 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 Oracle Access Manager 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 8–7 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 8–10 describes the requirements of the X509 authentication module.

Table 8–10 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.

8.7.2 Creating a New Authentication Module of an Existing Type

Users with valid 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:** Authentication modules are a core component of Authentication Schemes in access policies. Ensure that each Authentication Module points to the appropriate Identity Store.
>
> If you change the System Store, you must also change the LDAP Authentication Module to reference the newly designated System Store.

Prerequisites

About Default Authentication Modules and Pages

> **See Also:** Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service if you want to create an authentication module for a custom plug-in.

To create a new authentication module of an existing type

1. From System Configuration tab, Access Manager Settings section, expand the Authentication Modules node.
2. From the Authentication Modules node, click the desired module type:
   - LDAP Authentication module
   - Kerberos Authentication module
   - X509 Authentication module
3. Click the Create button in the tool bar.
4. Add details for the new authentication module:

- LDAP: See Table 8–9
- Kerberos: See Table 8–8
- X509: See Table 8–10

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 "Managing Authentication Schemes" on page 12-15.

8.7.3 Viewing or Editing Authentication Modules

Users with valid 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 and Pages

To find, 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, Access Manager Settings section, 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 8–8
   - LDAP Module: See Table 8–9
   - X509 Module: See Table 8–10
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 "Managing Authentication Schemes" on page 12-15.

8.7.4 Deleting an Authentication Module

Users with valid Administrator credentials can use the following procedure to delete an authentication module.

The following procedure is the same whether you are deleting a custom authentication module or a standard one.

Prerequisites

In each authentication scheme that references the module to be deleted, 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, Access Manager Settings section, 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).

8.8 Creating Custom Authentication Modules

This section provides the following topics:

- About Creating Custom Authentication Modules
- About the Custom Authentication Module Plug-ins
- Creating a Custom Authentication Module

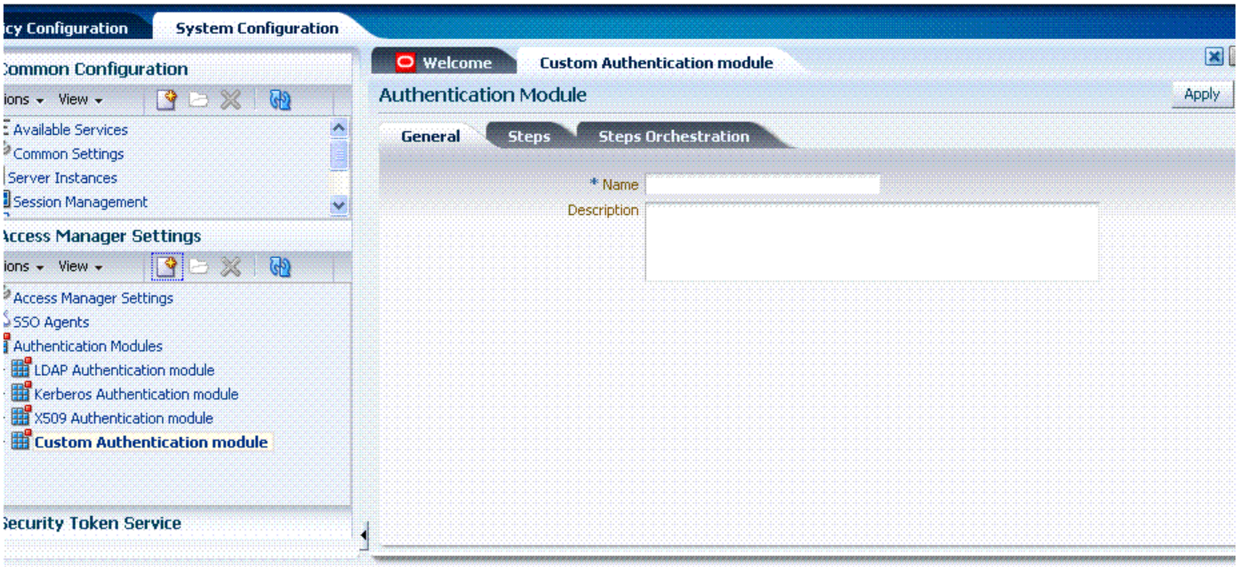
8.8.1 About Creating Custom Authentication Modules

Each custom authentication module requires the following types of information:

- General
- Steps
- Step Orchestration

Figure 8–8 shows the Custom Authentication Module within the Access Manager Settings section of the System Configuration tree. You can also see three subtabs where you enter information for the module.

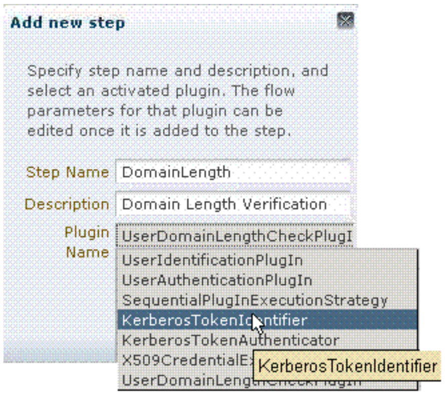
Figure 8–8 Custom Authentication Modules Node and General Subtab



The General subtab provides space for the module Name and an optional description. The name can be up to 60 characters. The optional description can be up to 250 characters.

When you add a new Step, the following dialog box appears. Information that you enter is used to populate the table and Details sections of the page.

Figure 8–9 Adding a Step and Associating a Plug-in


Add new step

Specify step name and description, and select an activated plugin. The flow parameters for that plugin can be edited once it is added to the step.

Step Name: DomainLength

Description: Domain Length Verification

Plugin Name: UserDomainLengthCheckPlugI

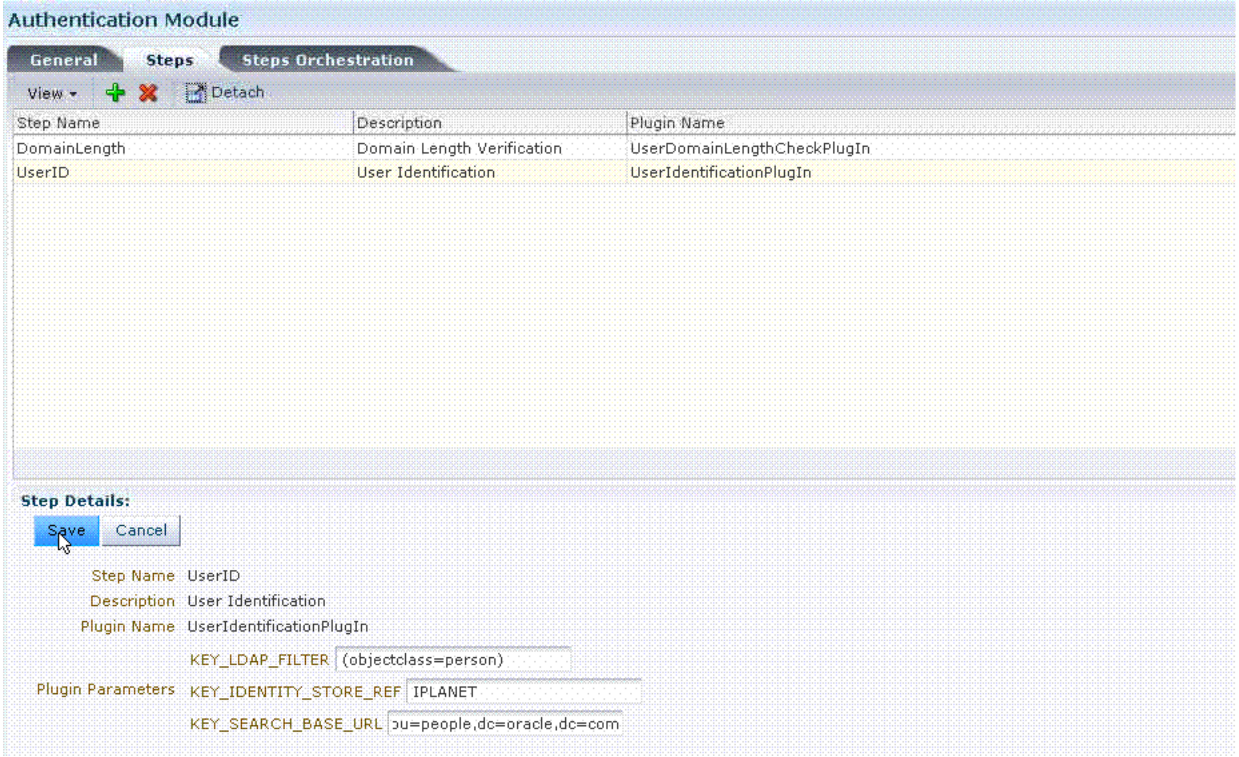
Table 8–11 describes the information required for adding a new step.

Table 8–11 Add New Step Entries, Steps Results Table, and Details Section

Element	Description
Step Name	The unique name you enter when adding the step.
Description	The optional description for this step, entered when adding the step.
Plugin Name	The plug-in name that you select when adding the step.
Step Details	Details of the selected step in the results table, and Plug-in configuration details that are set when the plug-in is added. These will differ depending on the selected plug-in.

Figure 8–10 illustrates the Steps subtab and Details section for a custom authentication module. When you are adding Steps, there is no data to display in the table. However, once you add one or more Steps the table and Details sections are populated.

Figure 8–10 Custom Authentication Module Steps Subtab and Details Section

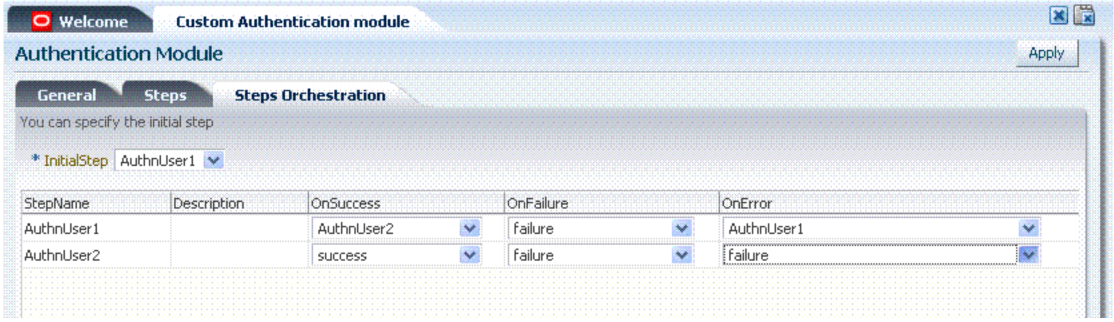


Step Name	Description	Plugin Name
DomainLength	Domain Length Verification	UserDomainLengthCheckPlugIn
UserID	User Identification	UserIdentificationPlugIn

Step Name UserID
Description User Identification
Plugin Name UserIdentificationPlugIn
Plugin Parameters
KEY_LDAP_FILTER (objectclass=person)
KEY_IDENTITY_STORE_REF IPLANET
KEY_SEARCH_BASE_URL ɔu=people,dc=oracle,dc=com

Figure 8–11 illustrates the Steps Orchestration subtab of a custom authentication module, which is populated by information for each defined step (and the action you choose for each operational condition).

Figure 8–11 Custom Authentication Module Steps Orchestration Subtab



You can specify the initial step

* InitialStep AuthnUser1

StepName	Description	OnSuccess	OnFailure	OnError
AuthnUser1		AuthnUser2	failure	AuthnUser1
AuthnUser2		success	failure	failure

Table 8–12 describes the elements on the Steps Orchestration subtab. The lists available for OnSuccess, OnFailure, and OnError include the following choices:

- success
- failure

- *StepName* (any step in the module can be selected as the action for an operational condition)

Table 8–12 Steps Orchestration Subtab

Element	Description
Initial Step	Choose the starting step from those listed. The list includes only those steps defined for this module.
Name	Each step that has been added is listed by the name that was entered when the step was added.
Description	The optional description for this step, entered when this step was added.
OnSuccess	The action selected for successful operation of this step. A list provides actions you can choose: ■ Success ■ Failure ■ *StepName*
OnFailure	The action selected for failure of this step. A list provides actions you can choose: ■ Success ■ Failure ■ *StepName*
OnError	The action selected for an error when executing this step. A list provides actions you can choose: ■ Success ■ Failure ■ *StepName*

8.8.2 About the Custom Authentication Module Plug-ins

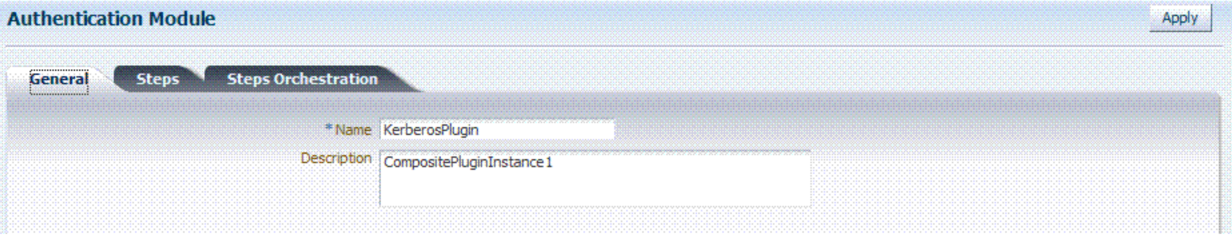
Oracle Access Manager 11g provides several Custom Authentication Module plug-ins that you can use to compose your own custom authentication modules and organize these into a multi-stepped authentication module that you can assign to authentication schemes. Each module can point to an independent user identity store.

- KerberosPlugin
- LDAPPlugin
- X509Plugin

KerberosPlugin

Figure 8–12 shows the KerberosPlugin that is bundled with Oracle Access Manager 11g. This is a credential mapping module that matches the credentials (username and password) of the user who requests a resource to the encrypted "ticket".

Figure 8–12 KerberosPlugin

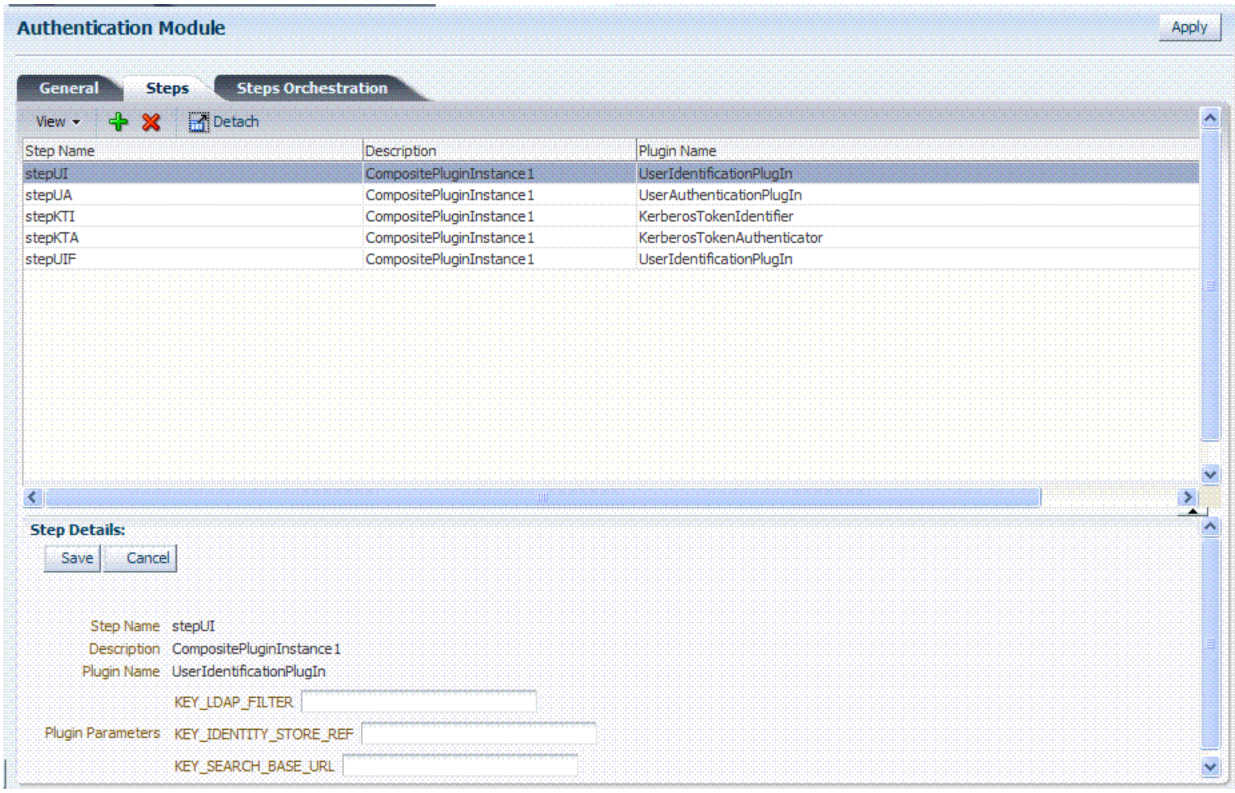


* Name: KerberosPlugin

Description: CompositePluginInstance1

Figure 8–13 shows the default steps and details. Figure 8–14 shows the orchestration of the steps and conditions.

Figure 8–13 Default KerberosPlugin Steps and Details



Step Name	Description	Plugin Name
stepUI	CompositePluginInstance1	UserIdentificationPlugIn
stepUA	CompositePluginInstance1	UserAuthenticationPlugIn
stepKTI	CompositePluginInstance1	KerberosTokenIdentifier
stepKTA	CompositePluginInstance1	KerberosTokenAuthenticator
stepUIF	CompositePluginInstance1	UserIdentificationPlugIn

Step Details:

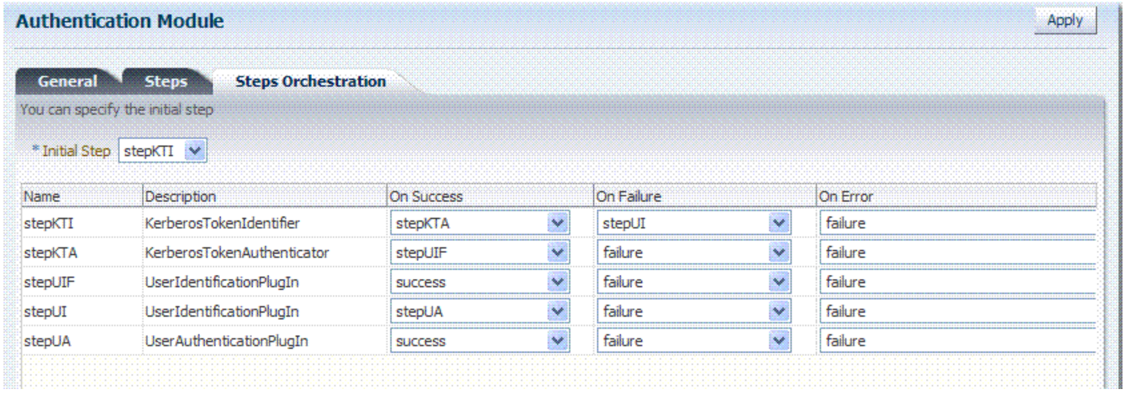
Step Name: stepUI

Description: CompositePluginInstance1

Plugin Name: UserIdentificationPlugIn

Plugin Parameters:
KEY_LDAP_FILTER
KEY_IDENTITY_STORE_REF
KEY_SEARCH_BASE_URL

Figure 8–14 Default KerberosPlugin Steps and Orchestration



You can specify the initial step

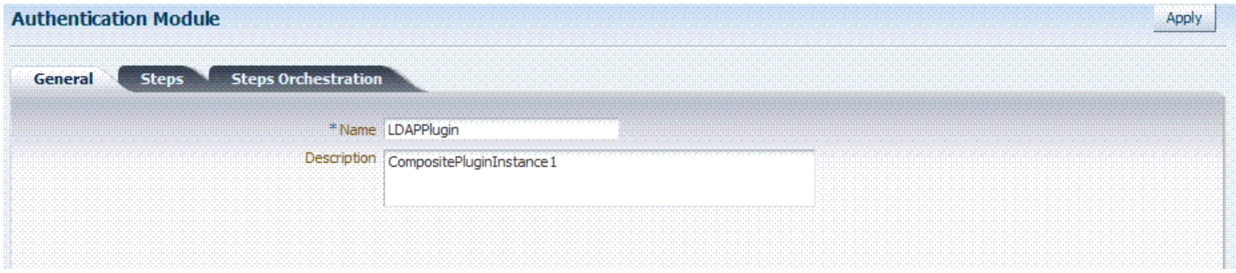
* Initial Step: stepKTI

Name	Description	On Success	On Failure	On Error
stepKTI	KerberosTokenIdentifier	stepKTA	stepUI	failure
stepKTA	KerberosTokenAuthenticator	stepUIF	failure	failure
stepUIF	UserIdentificationPlugIn	success	failure	failure
stepUI	UserIdentificationPlugIn	stepUA	failure	failure
stepUA	UserAuthenticationPlugIn	success	failure	failure

LDAPPlugin

Figure 8–15 shows the LDAPPlugin that is bundled with Oracle Access Manager 11g. By default, LDAPPlugin has 2 steps, shown in Figure 8–16. Figure 8–17 shows the default orchestration of steps for LDAPplugin.

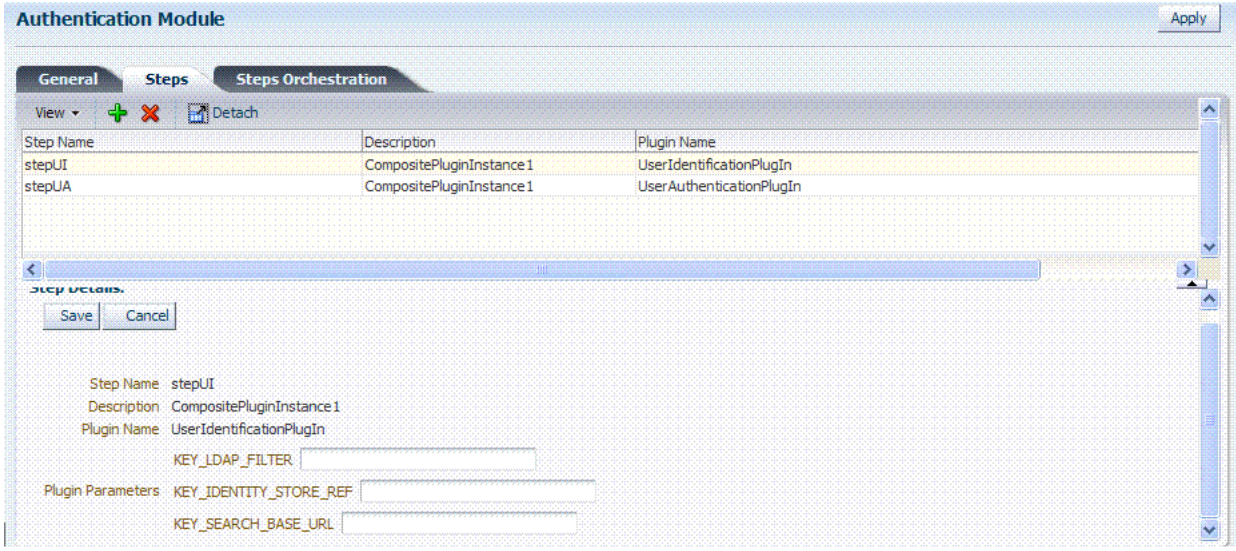
Figure 8–15 LDAPPlugin



* Name: LDAPPlugin

Description: CompositePluginInstance1

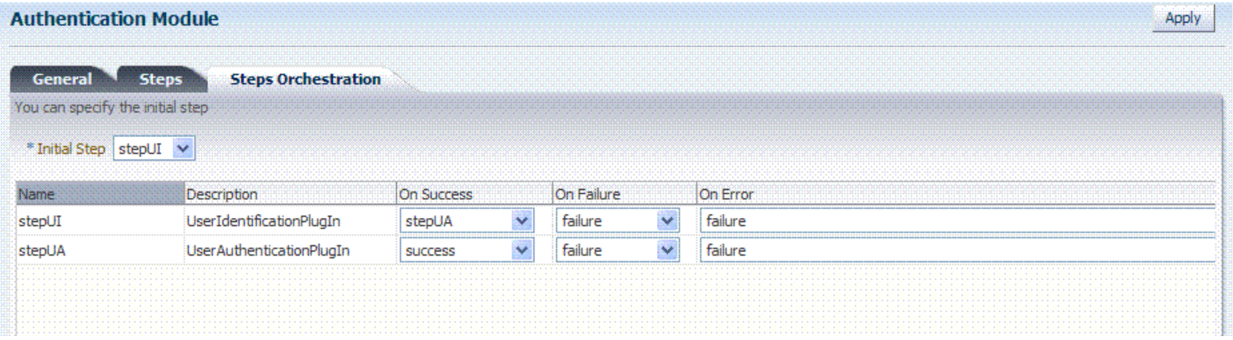
Figure 8–16 Default LDAPPlugin Steps and Details



Step Name	Description	Plugin Name
stepUI	CompositePluginInstance1	UserIdentificationPlugIn
stepUA	CompositePluginInstance1	UserAuthenticationPlugIn

Step Name: stepUI
Description: CompositePluginInstance1
Plugin Name: UserIdentificationPlugIn
Plugin Parameters: KEY_LDAP_FILTER
KEY_IDENTITY_STORE_REF
KEY_SEARCH_BASE_URL

Figure 8–17 Default Orchestration of Steps for LDAPplugin



Authentication Module

You can specify the initial step

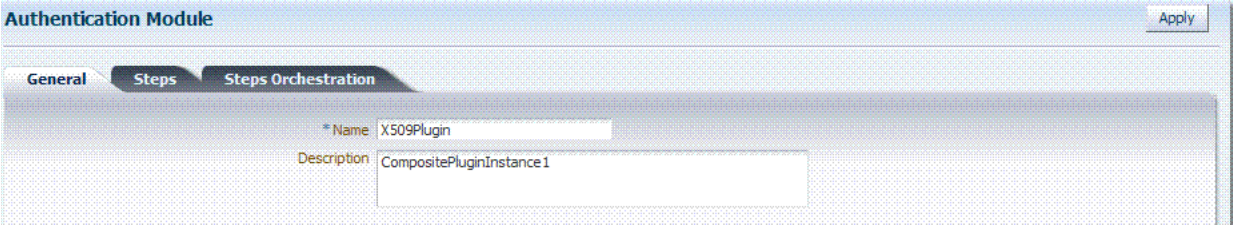
* Initial Step: stepUI

Name	Description	On Success	On Failure	On Error
stepUI	UserIdentificationPlugIn	stepUA	failure	failure
stepUA	UserAuthenticationPlugIn	success	failure	failure

X509Plugin

Figure 8–18 shows the X509Plugin that is bundled with Oracle Access Manager 11g. The X509Plugin is similar to the LDAPPlugin with additional properties that indicate which attribute of the client's X.509 certificate should be validated against the user attribute in LDAP. Figure 8–19 shows default steps and details for this plug-in. Figure 8–20 shows the default orchestration of steps for the X509Plugin.

Figure 8–18 X509Plugin

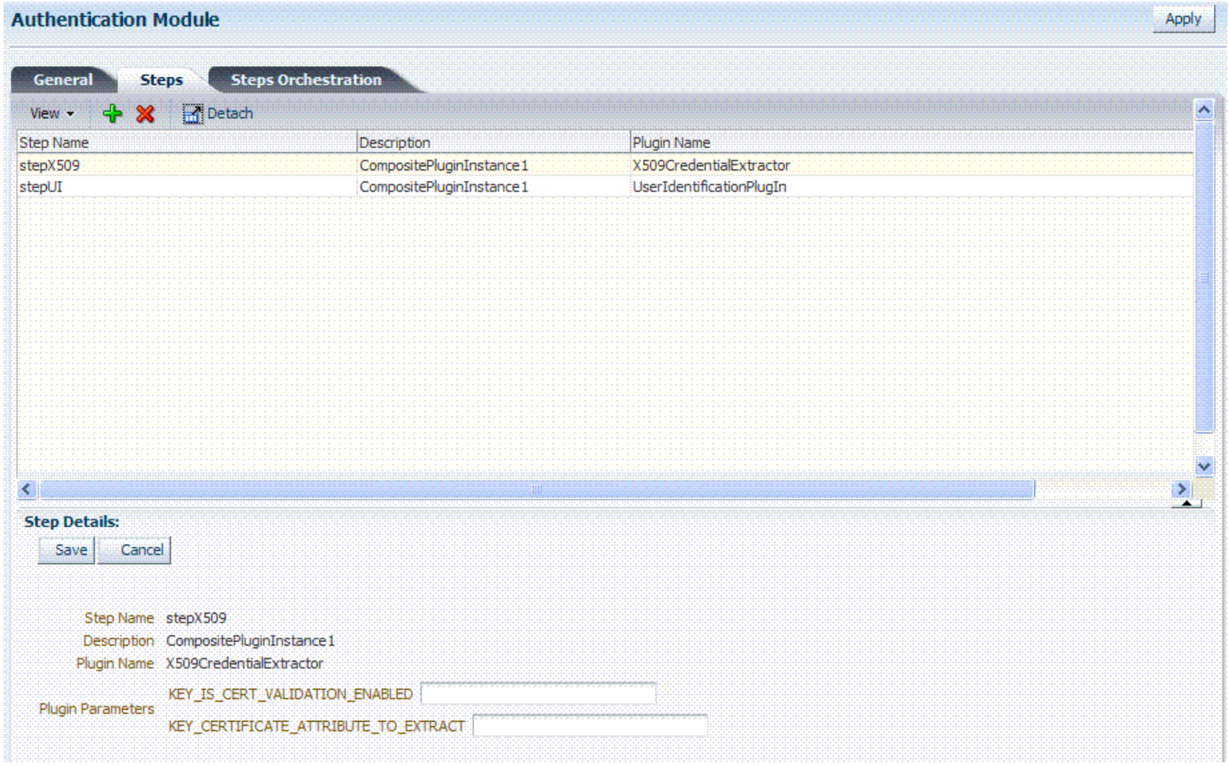


Authentication Module

* Name: X509Plugin

Description: CompositePluginInstance1

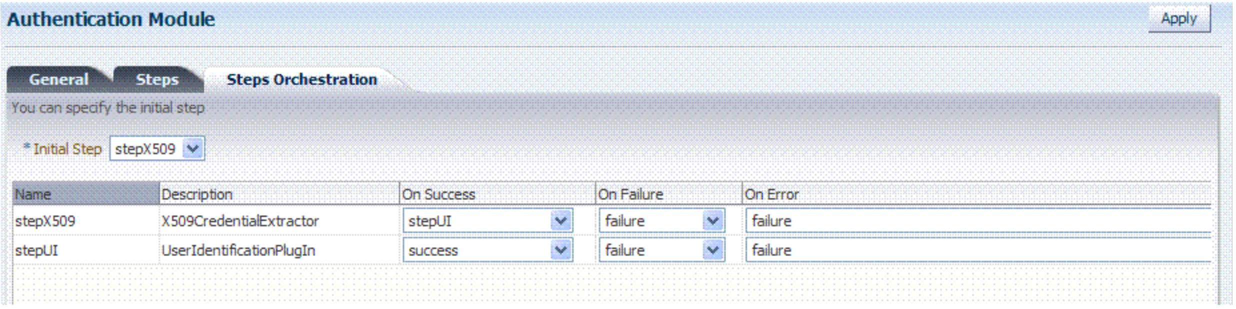
Figure 8–19 X509Plugin Default Steps and Details



Step Name	Description	Plugin Name
stepX509	CompositePluginInstance1	X509CredentialExtractor
stepUI	CompositePluginInstance1	UserIdentificationPlugIn

Step Name stepX509
Description CompositePluginInstance1
Plugin Name X509CredentialExtractor
Plugin Parameters KEY_IS_CERT_VALIDATION_ENABLED
KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT

Figure 8–20 Default Orchestration for X509Plugin Steps



You can specify the initial step

* Initial Step stepX509

Name	Description	On Success	On Failure	On Error
stepX509	X509CredentialExtractor	stepUI	failure	failure
stepUI	UserIdentificationPlugIn	success	failure	failure

8.8.3 Creating a Custom Authentication Module

Users with valid Administrator credentials can use the following procedure to create custom authentication module that uses one or more authentication plug-ins.

Prerequisites

Ensure that any user identity store associated with the module is running and includes the required user population.

To create a custom authentication module using bundled plug-ins

1. From System Configuration tab, Access Manager Settings section, expand the Authentication Modules node.

2. **Create New**:
   a. Click the Custom Authentication Module node.
   b. Click the Create (+) button.
   c. Add General Information: Name and optional Description.

      Click Apply to save general information.
3. **Add Steps**:
   a. Click the Steps subtab.
   b. Click the Add button above the Steps table.
   c. In the Add New Step dialog box, enter a unique Step Name and optional Description.
   d. Browse for and select the desired plug-in name and click OK.
   e. Confirm information in the results table.
   f. Repeat b through e to add other steps until you have listed all required plug-ins for your module.
4. **Configure Each Step**: Use appropriate values for requested parameters (Table 8–11):
   a. Click a *StepName* in the table to reveal required details.
   b. Enter appropriate values for the requested details.
   c. Click the Save button.
   d. Repeat a through c to configure each step appropriately.
   e. Ensure that users are provisioned in any user identity stores assigned in steps.
5. **Orchestrate Steps**: See Table 8–12 as you perform following steps.
   a. Click the Steps Orchestration subtab.
   b. From the InitialStep list, choose the name of the first step to be used.
   c. Select a *StepName* in the table.
   d. From the OnSuccess List, choose a condition (success or failure) or a step name name.
   e. From the OnFailure List, choose the desired condition or a *StepName*.
   f. From the OnError List, choose the desired condition or a *StepName*.
   g. Repeat c through e to orchestrate operations for each plug-in this module.
   h. Review your orchestration.
6. **Initiate Strategy Validation**: Click Apply to initiate validation of your orchestration strategy:
   - **Successful Strategy**: The orchestration strategy is applied and the module is ready to include in an authentication scheme. Continue with Steps 9 and 10.
   - **Invalid Strategy**: Click OK in the Error box, then edit your OnSuccess, OnFailure, OnError strategies (or add or remove plug-ins) to correct the problem. Repeat this step until your strategy is successful.
7. In the navigation tree, confirm the new Custom Authentication Module is listed, and then close the page when you finish.

8. You are ready to use the custom module in an authentication scheme.

9

Registering Partners (Agents and Applications) by Using the Console

Only a registered policy enforcement agent can communicate with OAM 11g authentication and authorization services. Administrator 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 Oracle Access Manager Console to perform agent registration and management. This chapter includes the following topics:

- Prerequisites
- Introduction to Policy Enforcement Agents
- Registering and Managing OAM Agents Using the Console
- Tuning 10g and 11g Webgate Caches
- Registering and Managing OSSO Agents Using the Console

Note: To use the command-line to register a partner, see Chapter 10.

9.1 Prerequisites

Before you can perform tasks in this chapter ensure that the Oracle Access Manager 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 27, "Managing OAM 10g Webgates with OAM 11g" if you are registering OAM 10g Webgates

9.2 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 IAMSuiteAgent

- About Registering Partners (Agents and Applications)
- About File System Changes and Artifacts for Registered Agents

9.2.1 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
  - Programmatic Access Clients
- **OSSO Agent**s: mod_osso is part of the (still) 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 Access Clients:

- 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.
- Custom access clients created for use with non-Web applications must be specifically developed using the Software Developer Kit (SDK), either by you or by Oracle. An Access Client processes requests for Web and non-Web resources (non-HTTP) from users or applications.

Table 9–1 provides information about all agents for OAM 11g.

Table 9–1 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. You can also register 11g Access Clients (Those created with the OAM 11g Access SDK).

Table 9–1 (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 27. ■ 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. ■ Legacy 10g Java AccessGates. See Also IAMSuiteAgent in this table.
Access Clients	Only authentication and authorization is (not policy modification) for Access Clients.
IAMSuiteAgent	The IAM Suite Agent provides single sign-on functionality for the IAM suite of consoles. The IAM Suite Agent and companion application domain (IAMSuite) replaces the earlier IDM Domain Agent and its companion application domain. See Also: "About the Pre-Registered IAMSuiteAgent" on page 9-4.

Table 9–2 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).

See Also:

- Chapter 11 for details about SSO and cookies

Table 9–2 Comparing Agent Types and Differences

	OAM 11g	OAM 10g	OSSO 10g
Available SSO Agents	OAM Agents ■ 11g Webgate ■ 10g Webgate ■ IAMSuiteAgent ■ Programmatic Access Client 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	Use oamreg tool to register OAM 10g and 11g Agents with OAM 11g. *<OAM_HOME>*/oam/server/rreg/client/rreg/bin/oamreg	No equivalent for OAM 10g.	Use oamreg tool to register OSSO Agents with OAM 11g **Note**: 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

Table 9–2 (Cont.) Comparing Agent Types and Differences

	OAM 11g	OAM 10g	OSSO 10g
logout.html	See Chapter 15 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 15.	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.
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 Oracle Access Manager Console or the remote registration tool to:

- Register a freshly installed OAM 11g Webgate
- Provision a legacy (or freshly installed) OAM 10 Webgate for use with OAM 11g, as described in Chapter 27.
- 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 A, "Co-existence Overview: OAM 11g and OSSO 10g".

9.2.2 About the Pre-Registered IAMSuiteAgent

This agent, and the companion application domain, described in Chapter 13, are available with the patch set release. Oracle strongly recommends that you do not alter these definitions.

Note: The original IDMDomainAgent is not available with this patch set. It remains as an artifact after you apply the patch set. However, all content is removed.

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

The IAMSuiteAgent is a domain-wide agent:

- Once deployed, the IAMSuiteAgent is installed on every server in the domain
- Unless disabled, every request coming into the WebLogic Application Server is evaluated and processed by the IAMSuiteAgent
- Configuration details are located under the 10g Webgates node (Policy Configuration tab) in the Oracle Access Manager Console

Certain IAMSuiteAgent configuration elements are available in the WebLogic Administration Console (in the Security Provider section) and others in the Oracle Access Manager Console.

WebLogic Administration Console, Security Provider Settings

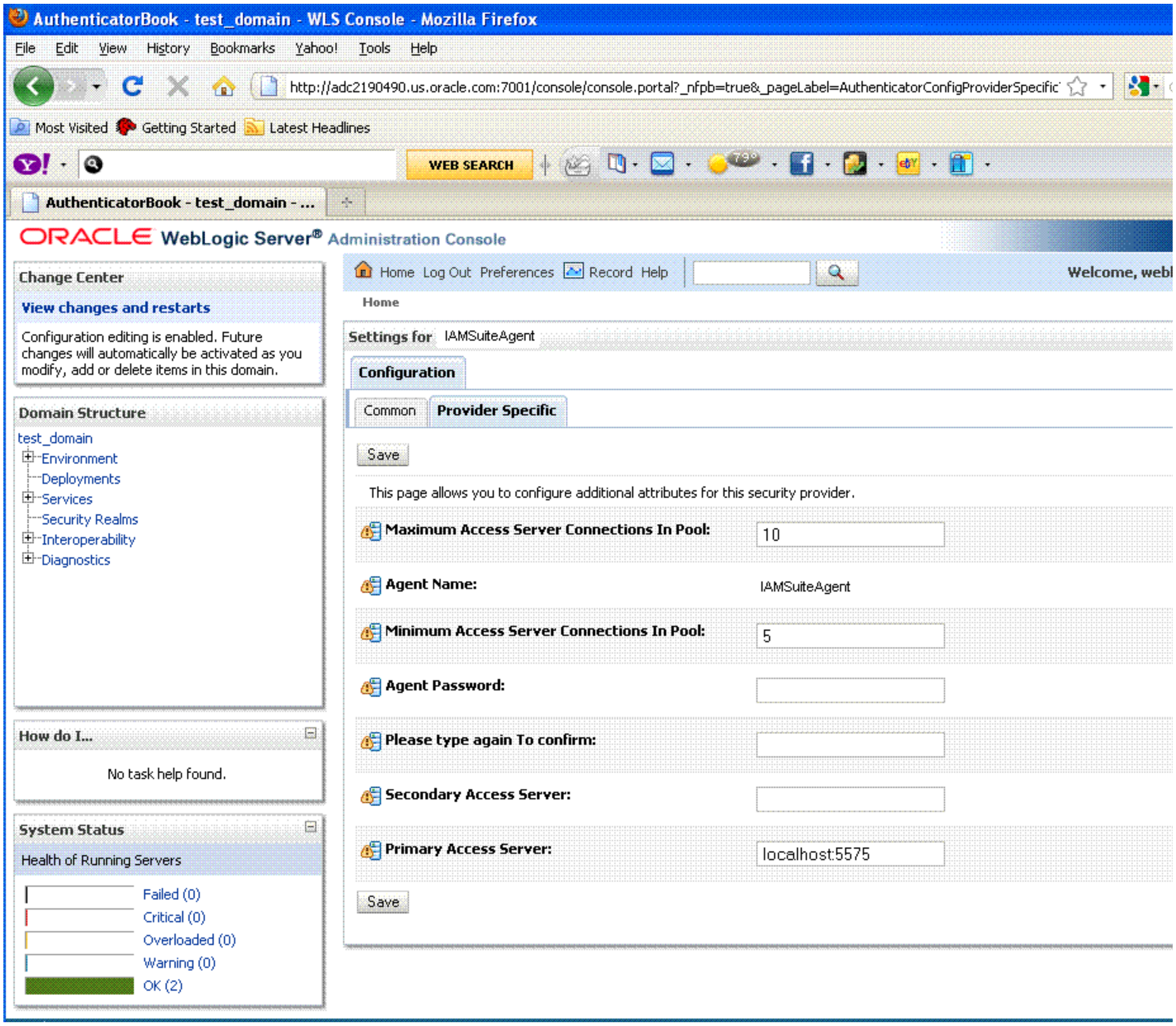
In the Security Provider section of the WebLogic Administration Console are five bootstrap configuration parameters.

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 OAM 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 IAMSuiteAgent connection to the OAM Server through the NetPoint (now Oracle) Access Protocol (NAP or OAP).

Figure 9–1 illustrates the default Security Provider settings for the IAMSuiteAgent.

Figure 9–1 IAMSuiteAgent 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
Home Log Out Preferences Record Help
Welcome, webl
Home
Change Center
View changes and restarts
Domain Structure
test_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
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.

No task help found.

Settings for IAMSuiteAgent

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

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

Oracle Access Manager Console, IAMSuiteAgent Registration

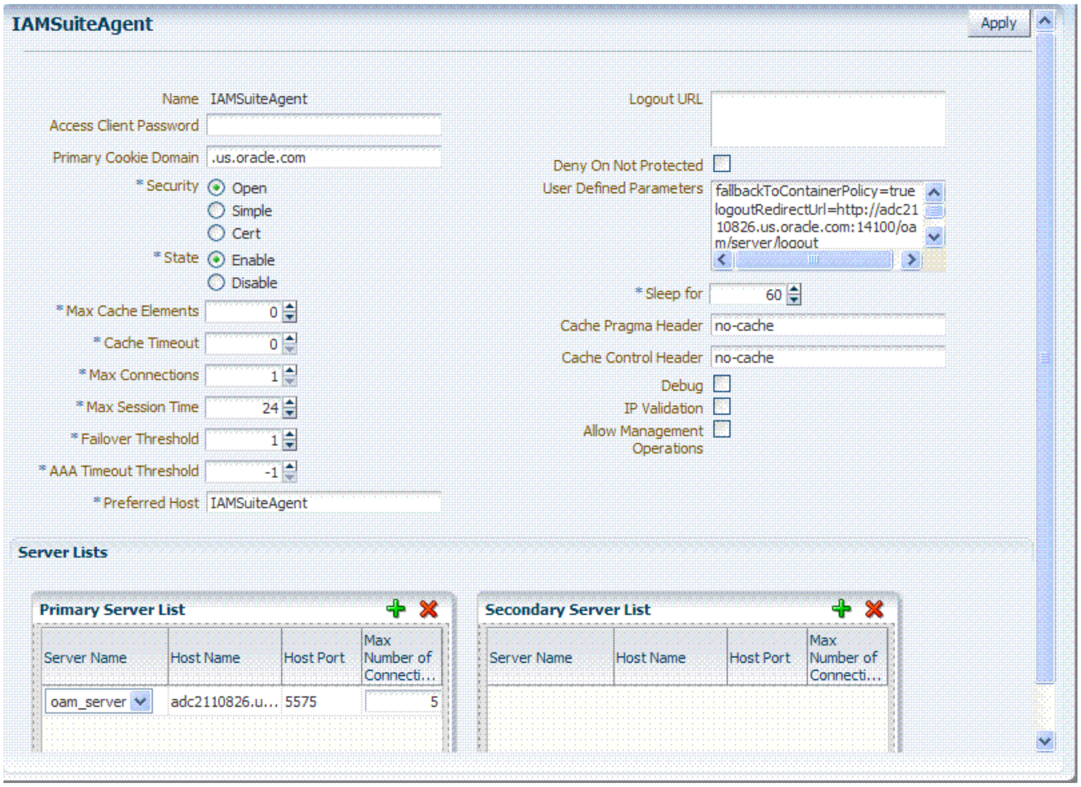
The IAMSuiteAgent registration page provides details about the agent, like all other OAM agent registration pages.

- Security Mode: Open is the only security mode available for the IAMSuiteAgent. This cannot be changed.
- Preferred Host: IAMSuiteAgent 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 9–2 shows the IAMSuiteAgent page. Notice the User Defined Parameter, which informs behavior to fall back to the container policy in the WebLogic Server and provides a redirect URL for logout.

Figure 9–2 IAMSuiteAgent Characteristics



IAMSuiteAgent

Apply

Name	IAMSuiteAgent
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	IAMSuiteAgent
Logout URL	
Deny On Not Protected	□

User Defined Parameters	fallbackToContainerPolicy=true
	logoutRedirectUrl=http://adc2110826.us.oracle.com:14100/oam/server/logout

* Sleep for	60
Cache Pragma Header	no-cache
Cache Control Header	no-cache
Debug	□
IP Validation	□
Allow Management Operations	□

Server Lists

Primary Server List

Server Name	Host Name	Host Port	Max Number of Connecti...
oam_server	adc2110826.u...	5575	5

Secondary Server List

Server Name	Host Name	Host Port	Max Number of Connecti...

Table 9–3 outlines the differences between IAMSuiteAgent and 11g and 10g Webgates. All elements are described in Table 9–5, " Expanded OAM 11g and 10g Webgate Elements and Defaults".

Table 9–3 Comparing IAMSuiteAgent and 11g and 10g Webgates

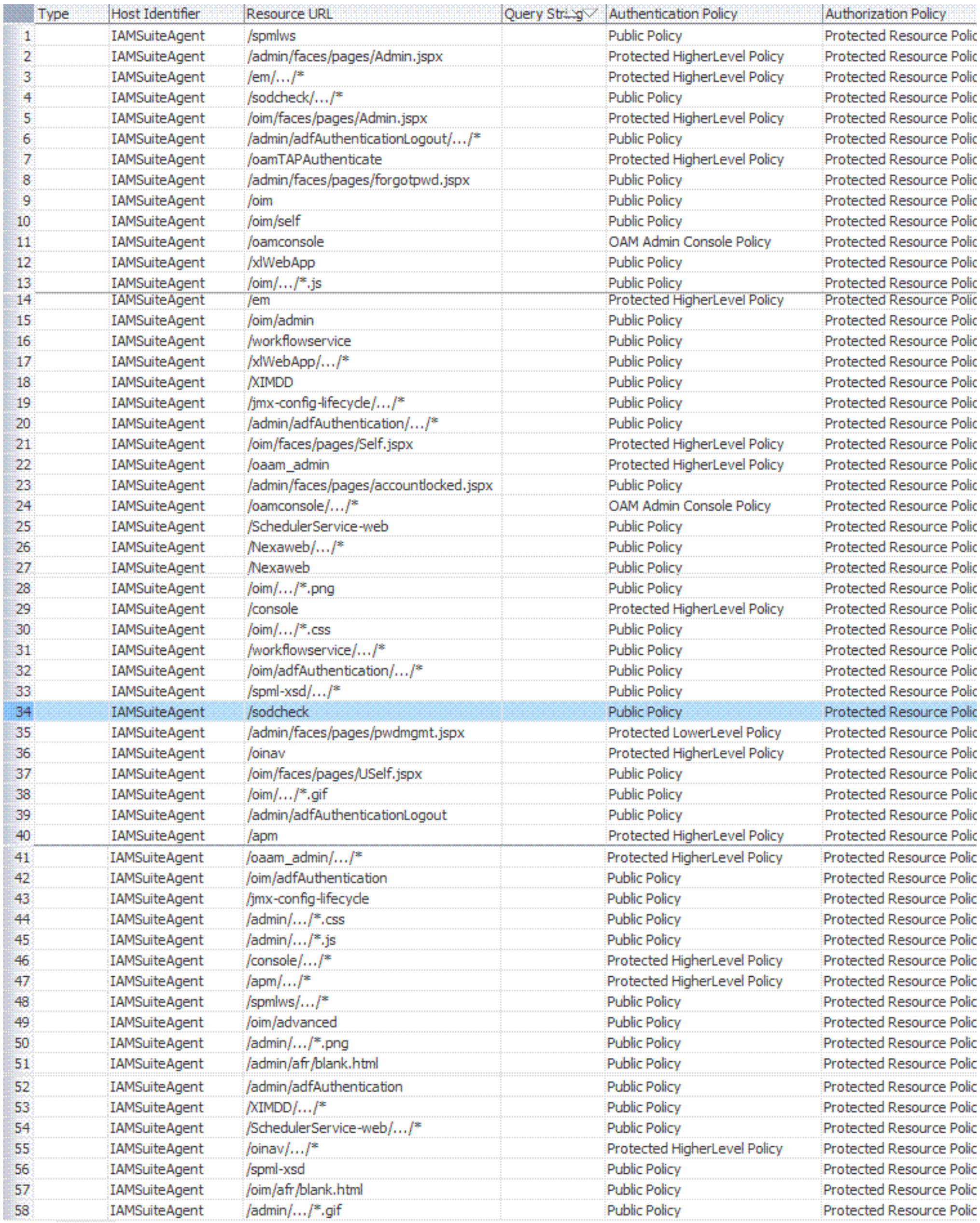
Element	11g Webgate	10g Webgate	IAMSuiteAgent
Primary Cookie Domain	N/A	x	x
Token Validity Period	x	N/A	N/A
Preferred Host	x	x	x
Logout URL	x	x	x
Logout Callback URL	x	N/A	N/A
Logout Redirect URL	x	N/A	N/A
Logout Target URL	x	N/A	N/A
Cache Pragma Header	x	x	x
Cache Control Header	x	x	x

User Defined Parameters	`proxySSLHeaderVar=IS_SSL`
	`URLInUTF8Format=true`
	`client_request_retry_attempts=1`
	`inactiveReconfigPeriod=10`
		`proxySSLHeaderVar=IS_SSL`
		`URLInUTF8Format=true`
		
		`client_request_retry_attempts=1`
		`inactiveReconfigPeriod=10`
			`fallbackToContainerPolicy=true`
			`logoutRedirectUrl=http://hostname.domain.com:14100/oam/server/logout`
			`protectWebXmlSecuredPagesOnly=true`

Deny on Not Protected	x	x	x

Figure 9–3 illustrates the resources protected by the IAMSuiteAgent, including the exact Authentication and Authorization policies. Oracle recommends that you do not make any additions or changes. The WebLogic Administration Console (/console) is protected.

Figure 9–3 Resources Protected by the IAMSuiteAgent



	Type	Host Identifier	Resource URL	Query String	Authentication Policy	Authorization Policy
1		IAMSuiteAgent	/spmlws		Public Policy	Protected Resource Polic
2		IAMSuiteAgent	/admin/faces/pages/Admin.jspx		Protected HigherLevel Policy	Protected Resource Polic
3		IAMSuiteAgent	/em/.../*		Protected HigherLevel Policy	Protected Resource Polic
4		IAMSuiteAgent	/sodcheck/.../*		Public Policy	Protected Resource Polic
5		IAMSuiteAgent	/oim/faces/pages/Admin.jspx		Protected HigherLevel Policy	Protected Resource Polic
6		IAMSuiteAgent	/admin/adfAuthenticationLogout/.../*		Public Policy	Protected Resource Polic
7		IAMSuiteAgent	/oamTAPAuthenticate		Protected HigherLevel Policy	Protected Resource Polic
8		IAMSuiteAgent	/admin/faces/pages/forgotpwd.jspx		Public Policy	Protected Resource Polic
9		IAMSuiteAgent	/oim		Public Policy	Protected Resource Polic
10		IAMSuiteAgent	/oim/self		Public Policy	Protected Resource Polic
11		IAMSuiteAgent	/oamconsole		OAM Admin Console Policy	Protected Resource Polic
12		IAMSuiteAgent	/xlWebApp		Public Policy	Protected Resource Polic
13		IAMSuiteAgent	/oim/.../*.js		Public Policy	Protected Resource Polic
14		IAMSuiteAgent	/em		Protected HigherLevel Policy	Protected Resource Polic
15		IAMSuiteAgent	/oim/admin		Public Policy	Protected Resource Polic
16		IAMSuiteAgent	/workflowservice		Public Policy	Protected Resource Polic
17		IAMSuiteAgent	/xlWebApp/.../*		Public Policy	Protected Resource Polic
18		IAMSuiteAgent	/XIMDD		Public Policy	Protected Resource Polic
19		IAMSuiteAgent	/jmx-config-lifecycle/.../*		Public Policy	Protected Resource Polic
20		IAMSuiteAgent	/admin/adfAuthentication/.../*		Public Policy	Protected Resource Polic
21		IAMSuiteAgent	/oim/faces/pages/Self.jspx		Protected HigherLevel Policy	Protected Resource Polic
22		IAMSuiteAgent	/oaam_admin		Protected HigherLevel Policy	Protected Resource Polic
23		IAMSuiteAgent	/admin/faces/pages/accountlocked.jspx		Public Policy	Protected Resource Polic
24		IAMSuiteAgent	/oamconsole/.../*		OAM Admin Console Policy	Protected Resource Polic
25		IAMSuiteAgent	/SchedulerService-web		Public Policy	Protected Resource Polic
26		IAMSuiteAgent	/Nexaweb/.../*		Public Policy	Protected Resource Polic
27		IAMSuiteAgent	/Nexaweb		Public Policy	Protected Resource Polic
28		IAMSuiteAgent	/oim/.../*.png		Public Policy	Protected Resource Polic
29		IAMSuiteAgent	/console		Protected HigherLevel Policy	Protected Resource Polic
30		IAMSuiteAgent	/oim/.../*.css		Public Policy	Protected Resource Polic
31		IAMSuiteAgent	/workflowservice/.../*		Public Policy	Protected Resource Polic
32		IAMSuiteAgent	/oim/adfAuthentication/.../*		Public Policy	Protected Resource Polic
33		IAMSuiteAgent	/spml-xsd/.../*		Public Policy	Protected Resource Polic
34		IAMSuiteAgent	/sodcheck		Public Policy	Protected Resource Polic
35		IAMSuiteAgent	/admin/faces/pages/pwdmgmt.jspx		Protected LowerLevel Policy	Protected Resource Polic
36		IAMSuiteAgent	/oinav		Protected HigherLevel Policy	Protected Resource Polic
37		IAMSuiteAgent	/oim/faces/pages/USelf.jspx		Public Policy	Protected Resource Polic
38		IAMSuiteAgent	/oim/.../*.gif		Public Policy	Protected Resource Polic
39		IAMSuiteAgent	/admin/adfAuthenticationLogout		Public Policy	Protected Resource Polic
40		IAMSuiteAgent	/apm		Protected HigherLevel Policy	Protected Resource Polic
41		IAMSuiteAgent	/oaam_admin/.../*		Protected HigherLevel Policy	Protected Resource Polic
42		IAMSuiteAgent	/oim/adfAuthentication		Public Policy	Protected Resource Polic
43		IAMSuiteAgent	/jmx-config-lifecycle		Public Policy	Protected Resource Polic
44		IAMSuiteAgent	/admin/.../*.css		Public Policy	Protected Resource Polic
45		IAMSuiteAgent	/admin/.../*.js		Public Policy	Protected Resource Polic
46		IAMSuiteAgent	/console/.../*		Protected HigherLevel Policy	Protected Resource Polic
47		IAMSuiteAgent	/apm/.../*		Protected HigherLevel Policy	Protected Resource Polic
48		IAMSuiteAgent	/spmlws/.../*		Public Policy	Protected Resource Polic
49		IAMSuiteAgent	/oim/advanced		Public Policy	Protected Resource Polic
50		IAMSuiteAgent	/admin/.../*.png		Public Policy	Protected Resource Polic
51		IAMSuiteAgent	/admin/afr/blank.html		Public Policy	Protected Resource Polic
52		IAMSuiteAgent	/admin/adfAuthentication		Public Policy	Protected Resource Polic
53		IAMSuiteAgent	/XIMDD/.../*		Public Policy	Protected Resource Polic
54		IAMSuiteAgent	/SchedulerService-web/.../*		Public Policy	Protected Resource Polic
55		IAMSuiteAgent	/oinav/.../*		Protected HigherLevel Policy	Protected Resource Polic
56		IAMSuiteAgent	/spml-xsd		Public Policy	Protected Resource Polic
57		IAMSuiteAgent	/oim/afr/blank.html		Public Policy	Protected Resource Polic
58		IAMSuiteAgent	/admin/.../*.gif		Public Policy	Protected Resource Polic

You can replace this agent with a 10g Webgate, as described in Chapter 27, "Managing OAM 10g Webgates with OAM 11g".

9.2.3 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.

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 Oracle Access Manager 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 Keystore on the server side.

  The Agent specific key must be accessible to Webgates through a secure local storage on the client machine. See Table 9–2.

- A key is generated for the partner (application) during registration. (except for 10g Webgates).
- 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 13.

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 Oracle Access Manager Console or custom WLST commands for OAM 11g.

For more information, see:

- Registering and Managing OAM Agents Using the Console
- Registering and Managing OSSO Agents Using the Console

> **See Also:**
>
> - Chapter 10, "Registering Partners (Agents and Applications) Remotely"
> - Chapter 27, "Managing OAM 10g Webgates with OAM 11g"
> - Appendix F, "Introduction to Custom WLST Commands for Administrators"

9.2.4 About File System Changes and Artifacts for Registered Agents

When you register an agent using the Oracle Access Manager Console, a new file system directory is created for the Agent on the Oracle Access Manager 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/Access Client: 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/Access Client: copy generated files to *Webgate_install_dir*/access/oblix/lib

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 IAMSuiteAgent 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. You can use the Oracle Access Manager 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

9.3 Registering and Managing OAM Agents Using the Console

This section describes how to manage OAM Agents using the Oracle Access Manager Console. Topics include:

- About Creating and Editing Webgate Registration
- About User-Defined Webgate Parameters
- About IP Address Validation for Webgates
- Searching for an OAM Agent Registration

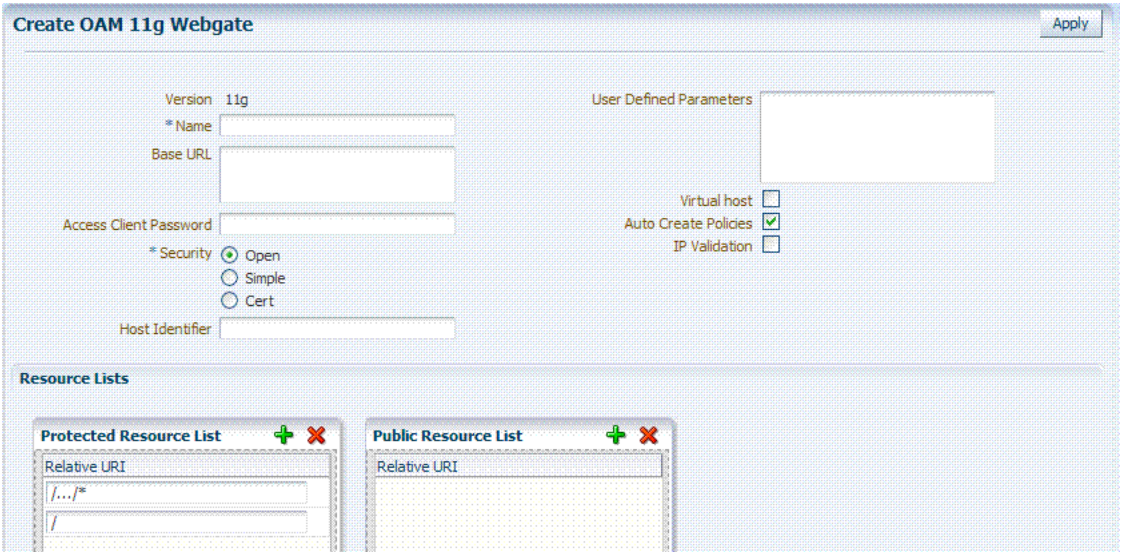
- Registering a Webgate or Programmatic Access Client
- Viewing or Editing an OAM Agent Registration
- Deleting Webgate Registration

> **See Also:** The following chapters as needed
>
> - Chapter 27, "Managing OAM 10g Webgates with OAM 11g"
> - Chapter 10, "Registering Partners (Agents and Applications) Remotely"
> - Appendix E, "Securing Communication for Oracle Access Manager 11g"

9.3.1 About Creating and Editing Webgate Registration

The Create OAM *nn*g Webgate page requests minimal information to streamline registration. Required details are identified by the asterisk (*). Compare the 11g Webgate page in Figure 9–4 with the Create OAM 10g Webgate page in Figure 9–5.

Figure 9–4 Create OAM 11g Webgate Page



Create OAM 11g Webgate

Apply

Version 11g

* Name

Base URL

Access Client Password

* Security ● Open ○ Simple ○ Cert

Host Identifier

User Defined Parameters

Virtual host □

Auto Create Policies ☑

IP Validation □

Resource Lists

Protected Resource List

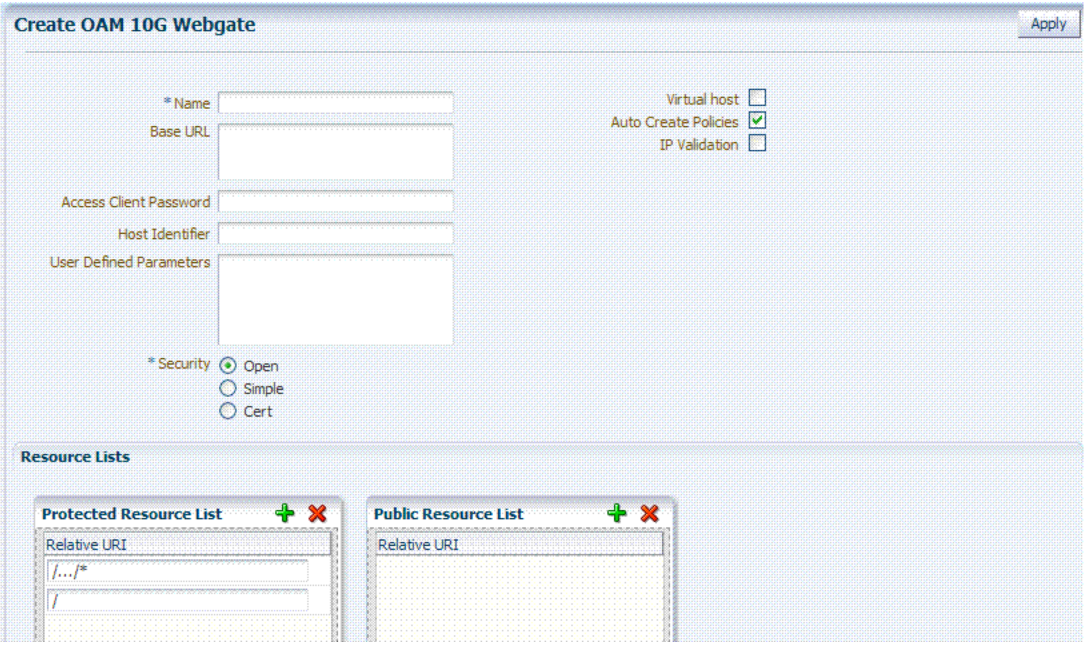
Relative URI
/…/*
/

Public Resource List

Relative URI

Whether you register an OAM 11g Webgate or 10g Webgate, the initial information requested is nearly the same, as shown in Figure 9–5.

Figure 9–5 Create OAM 10g Webgate Page



* Name

Base URL

Access Client Password

Host Identifier

User Defined Parameters

* Security ● Open ○ Simple ○ Cert

Virtual host □

Auto Create Policies ☑

IP Validation □

Protected Resource List — Relative URI: /.../*, /

Public Resource List — Relative URI

Table 9–4 describes the Create page for 10g and 11g Webgates.

Table 9–4 Create Pages for OAM 10g and 11g Webgates

OAM Agent Element	Description
Name	The unique identifying name for this Agent registration. This is often the name of the computer that is hosting the Web server used by Webgate. A unique identifying name for each Agent registration is preferred. However: ▪ If the Agent Name exists, no error occurs and the registration does not fail. Instead, Oracle Access Manager creates the policies if they are not already in place. ▪ If the host identifier exists, the unique Agent Base URL is added to the existing host identifier and registration proceeds.
Base URL Optional	The host and port of the computer on which the Web server for the Webgate is installed. For example, http://*my_host:port* or https://*my_host:port*. The port number is optional. **Note**: A particular Base URL can be registered once only. There is a one-to-one mapping from this 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 Base URLs.
Access Client Password Optional	An optional, unique password for this Webgate, which can be assigned during this registration process. 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 9–4 (Cont.) Create Pages for OAM 10g and 11g Webgates

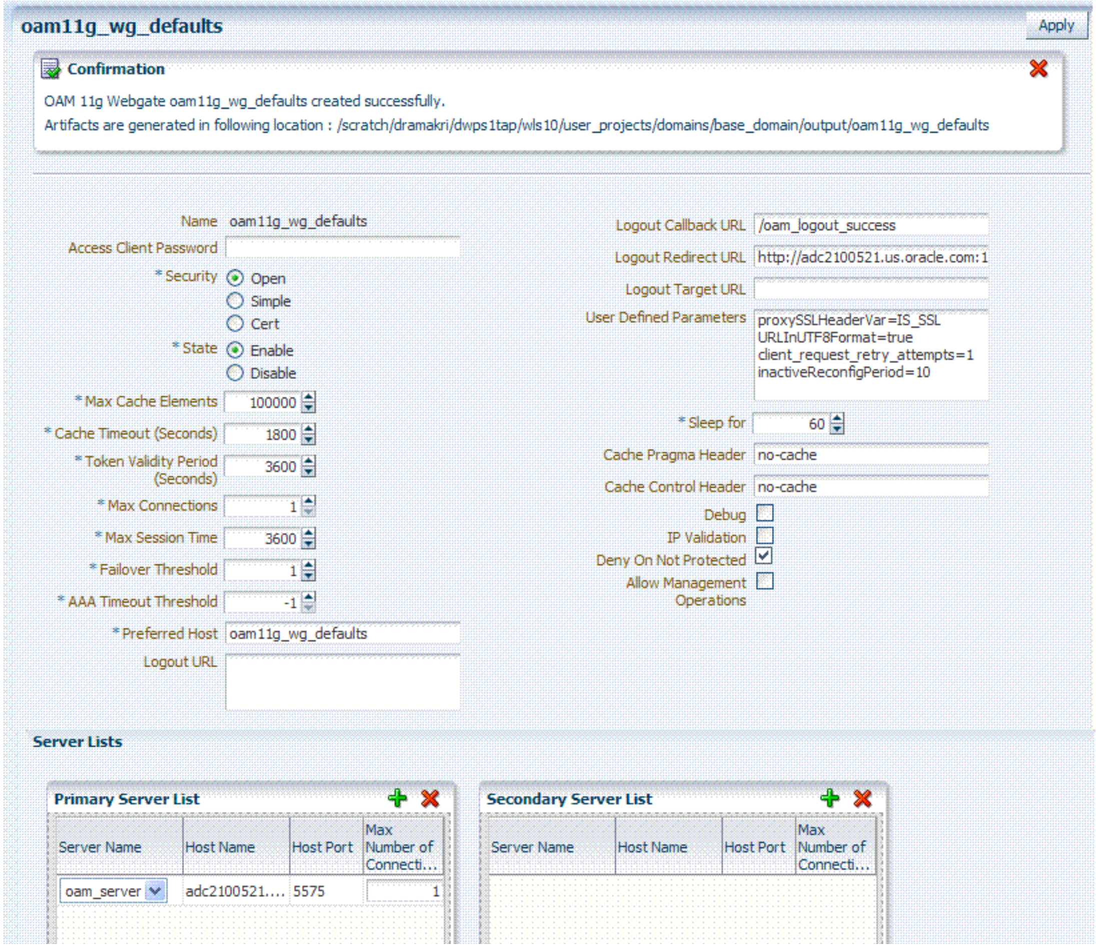
OAM Agent Element	Description
Security	Level of communication transport security between the Agent and the OAM Server (this must match the level specified for the OAM Server): ▪ Open--No transport security ▪ Simple--SSL v3/TLS v1.0 secure transport using dynamically generated session keys ▪ Cert--SSL 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. Agent Key Password: The private key file (aaa_key.pem) is encrypted using DES algorithm. The Agent Key Password is saved in obfuscated format in password.xml and is required by the server to generate password.xml. However, this password is not retained by the server. 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. **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. See Also: "About Virtual Web Hosting" on page 12-7.
User-defined Parameters	Parameters you can enter to enable specific Webgate behaviors: See Also: "About User-Defined Webgate Parameters" on page 9-21.
Virtual Host	Check the box beside Virtual Host if you installed a Webgate on a Web server that contains multiple Web site and domain names. The Webgate must reside in a location that enables it to protect all of the Web sites on that server. See Also: "About Virtual Web Hosting" on page 12-7.
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.
IP Validation	Check the box beside IP Validation to ensure a client's IP address is the same as the IP address stored in the ObSSOCookie generated for single sign-on. In the IP Validation Exceptions box, enter any IP addresses to exclude from validation using standard notation for the addresses: for example, 10.20.30.123. When enabled, the IP address stored in the ObSSOCookie must match the client's IP address. Otherwise, the cookie is rejected and the user must reauthenticate. Default: Disabled See Also: "About IP Address Validation for Webgates" on page 9-25.

Table 9–4 (Cont.) Create Pages for OAM 10g and 11g Webgates

OAM Agent Element	Description
Agent Key Password	Requested for only Cert mode communication, this passphrase is used to encrypt the private key used for SSL communication between Webgate and the OAM Server in Simple and Cert modes. **Note**: The Agent Key Password has no relationship to the Access Client Password described earlier within this table. **Simple Mode**: In this mode, the agent key password is a global passphrase that must be the same on both the client and server. Once the OAM Server has this configured, the password can be retrieved during agent registration. However, the administrator must copy to the client side, the password.xml file generated during agent registration. **Cert Mode**: In this mode, the agent key can be different on the client and server; it is no longer global. Administrators must enter the Agent Key Password to enable generation of a password.xml file during agent registration, which must be copied to the agent side. For certificate generation, you must encrypt the private key (used for SSL) using this password through openssl or other third-party tools to be placed inside aaa_key.pem. At runtime, Webgate retrieves the key from password.xml, and uses it to decrypt the key in aaa_key.pem. ■ If the key is encrypted, Webgate internally invokes the call back function to obtain the password. ■ If the key is encrypted and password.xml does not exist, Webgate cannot establish connections with the OAM Server. ■ If the key is not encrypted, there is no attempt to read password.xml. For more information, see Appendix E.
Resource Lists	
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 13-16.
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 Webgate registration, additional elements are concealed and default values are applied. When you view or edit a Webgate registration page in the console, all elements and values are revealed, as shown in Figure 9–6. Most elements are the same as those you define when using the remote registration tool with the expanded OAM template, as described in Chapter 10.

Figure 9–6 Confirmation Window and Expanded 11g Webgate Page with Defaults



oam11g_wg_defaults

Confirmation

OAM 11g Webgate oam11g_wg_defaults created successfully.

Artifacts are generated in following location : /scratch/dramakri/dwps1tap/wls10/user_projects/domains/base_domain/output/oam11g_wg_defaults

Name: oam11g_wg_defaults

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: oam11g_wg_defaults

Logout URL:

Logout Callback URL: /oam_logout_success

Logout Redirect URL: http://adc2100521.us.oracle.com:1

Logout Target URL:

User Defined Parameters:
proxySSLHeaderVar=IS_SSL
URLInUTF8Format=true
client_request_retry_attempts=1
inactiveReconfigPeriod=10

* Sleep for: 60

Cache Pragma Header: no-cache

Cache Control Header: no-cache

Debug: □

IP Validation: □

Deny On Not Protected: ☑

Allow Management Operations: □

Server Lists

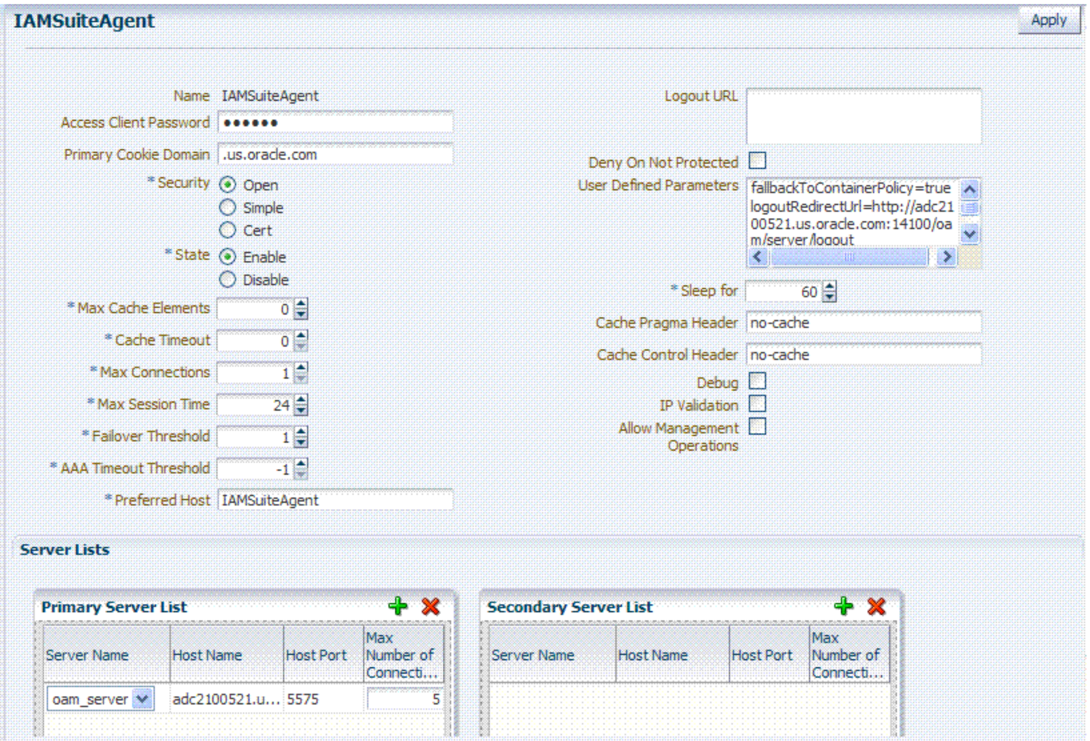
Primary Server List

Server Name	Host Name	Host Port	Max Number of Connecti...
oam_server	adc2100521....	5575	1

Secondary Server List

Server Name	Host Name	Host Port	Max Number of Connecti...

Figure 9–7 Expanded OAM 10g Webgate Registration Page



IAMSuiteAgent

Name IAMSuiteAgent

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 IAMSuiteAgent

Logout URL

Deny On Not Protected □

User Defined Parameters fallbackToContainerPolicy=true logoutRedirectUrl=http://adc2100521.us.oracle.com:14100/oam/server/logout

* Sleep for 60

Cache Pragma Header no-cache

Cache Control Header no-cache

Debug □

IP Validation □

Allow Management Operations □

Server Lists

Primary Server List

Server Name	Host Name	Host Port	Max Number of Connecti...
oam_server	adc2100521.u...	5575	5

Secondary Server List

Server Name	Host Name	Host Port	Max Number of Connecti...

Table 9–5 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 Oracle Access Manager Console as described here or the remote registration tool as described in Chapter 10.

Table 9–5 Expanded OAM 11g and 10g Webgate Elements and Defaults

OAM Agent Element	Description
Name	Name of this Webgate registration.
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.

Table 9–5 (Cont.) Expanded OAM 11g and 10g Webgate Elements and Defaults

OAM Agent Element	Description
State Set only in the Oracle Access Manager Console.	Specifies whether the OAM Agent is enabled or disabled. Default = Enabled
Max Cache Elements	Number of elements maintained in the cache. Caches are the following: ■ Resource to Authentication Scheme—This cache maintains information about Resources (URLs), including whether it is protected and, if so, the authentication scheme used for protection. ■ (11g Webgate only) Resource to Authorization Policy—This cache maintains information about Resources and associated authorization policy—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 these caches. Default = 100000
Cache Timeout (seconds)	Amount of time cached information remains in the Webgate caches (Resource to Authentication Scheme, Authentication Schemes, and 11g Webgate only Resource to Authorization Policy) when the information is neither used nor referenced. Default = 1800 (seconds)

User Defined Parameters
11g Webgate only
	■ Max Authorization Results Cache Elements—Number of elements maintained in the Authorization Result Cache. This cache maintains information about authorization results for associated user sessions. For example:
	`maxAuthorizationResultCacheElems=10000`
	Default = 100000
	■ Authorization Results Cache Timeout—Number of elements maintained in the Authorization Result Cache. This cache maintains information about authorization results for associated user sessions. For example:
	`authorizationResultCacheTimeout=60`
	Default, when not specified = 15 (seconds)
	See Also: "About User-Defined Webgate Parameters" on page 9-21 and "Tuning 10g and 11g Webgate Caches" on page 9-31.

Token Validity Period ***11g Webgate only***	Maximum valid time period for an agent token (the content of OAMAuthnCookie for 11g Webgate). 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 9–5 (Cont.) Expanded OAM 11g and 10g Webgate 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 OAM 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 OAM 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. Defaults to Name (of Webgate registration)

Table 9–5 (Cont.) Expanded OAM 11g and 10g Webgate 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 15 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 15, "Configuring Centralized Logout for 11g Webgate with OAM 11g Server"
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 15, "Configuring Centralized Logout for 11g Webgate with OAM 11g Server"

User-defined Parameters	Parameters you can enter to enable specific Webgate behaviors:
	Defaults:
	`proxySSLHeaderVar=IS_SSL`
	`URLInUTF8Format=true`
	`client_request_retry_attempts=1`
	`inactiveReconfigPeriod=10`
	See Also: "About User-Defined Webgate Parameters" on page 9-21.

Sleep for	The frequency (in seconds) with which the OAM Server checks its connections to the directory server. For example, if you set a value of 60 seconds, the OAM Server checks its connections every 60 seconds from the time it comes up. Default: 60 (seconds)

Table 9–5 (Cont.) Expanded OAM 11g and 10g Webgate Elements and Defaults

OAM Agent Element	Description
Cache Pragma Header Cache Control Header Webgate only (not Access Clients)	These settings apply only to Webgates and control the browser’s cache. By default, both parameters 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. Defaults: no-cache See Also: "Tuning 10g and 11g Webgate Caches" on page 9-31.
IP Validation	Check the box beside IP Validation to ensure a client's IP address is the same as the IP address stored in the ObSSOCookie generated for single sign-on. In the IP Validation Exceptions box, enter any IP addresses to exclude from validation using standard notation for the addresses: for example, 10.20.30.123. When enabled, the IP address stored in the ObSSOCookie must match the client's IP address. Otherwise, the cookie is rejected and the user must reauthenticate. Default: Disabled See Also: "About IP Address Validation for Webgates" on page 9-25.
Deny on Not Protected Webgates only (not Access Clients)	Oracle recommends enabling Deny On Not Protected. When enabled, this element denies access to all resources to which access is not explicitly allowed by a rule or policy. Enabling this can limit the number of times the Webgate queries the OAM Server, and can improve performance for large or busy application domains. ■ 11g Webgate: Always enabled, and cannot be changed ■ 10g Webgate: Can be disabled. **Important**: DenyOnNotProtected overrides Host Identifiers and Preferred Host. Oracle recommends enabling DenyOnNotProtected. Otherwise security holes can occur in large installations with multiple Host Identifiers, virtual hosts, and other complex configurations.

Table 9–5 (Cont.) Expanded OAM 11g and 10g Webgate Elements and Defaults

OAM Agent Element	Description
Allow Management Operations	This Agent Privilege function enables the provisioning of session operations per agent, as follows: ■ Terminate session ■ Enumerate sessions ■ Add or Update attributes for an existing session ■ List all attributes for a given session ID or read session Default: Disabled Note: Only privileged agents can invoke session management operations. When this parameter is enabled, session management requests (listed above) are processed by the OAM Server. If disabled, such requests are rejected for the agent.
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).)

9.3.2 About User-Defined Webgate Parameters

To implement user-defined parameters, you must enter them in the Webgate registration page as shown in Table 9–6.

> **See Also:** Table 10–6, " Elements Common to Full Remote Registration Requests" for details about User-defined Webgate parameters in remote registration requests.

Table 9–6 User-Defined Webgate Parameters

Parameter	Description
UrlInUTF8Format=true	In an environment that uses Oracle HTTP Server 2, this parameter must be set to true to display latin-1 and other character sets.

Table 9–6 (Cont.) User-Defined Webgate Parameters

Parameter	Description
ProxySSLHeaderVar=IS_SSL	Uses 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**
client_request_retry_attempts=1	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. If the OAM Server takes longer to service a request than the value of the timeout threshold, the Webgate abandons the request and retries the request on a new connection. Default: 1 **Note**: 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 Servers may also require more time to process the request than the time specified on the timeout threshold. In some cases, the Webgate can retry the request until the OAM Servers are shut down. You can configure a limit on the number of retries that the Webgate performs for a non-responsive server using the client_request_retry_attempts parameter.
InactiveReconfigPeriod=10	The Webgate update thread reads the shared secret from the OAM Server every 1 minute when Webgate is active. The OAM Server server returns the shared secret in its own cache (the OAM Server cache). Default: 10 (minutes) See Also: "Changing the Webgate Polling Frequency" on page 9-34.
fallbackToContainerPolicy=true	Used for the IAMSuiteAgent. When set to `false`, user access to the resource is denied and an HTTP response code, 403 is returned. When set to 'true' the request goes through to the container and uses whatever policy (related to J2EE authentication/authorization) is configured on the container to grant or deny the user access. Default: `true`
logoutRedirectUrl=	http://***host.domain.com:14100***/oam/server/logout

Table 9–6 (Cont.) User-Defined Webgate Parameters

Parameter	Description
protectWebXmlSecuredPagesOnly =true	Used for the IAMSuiteAgent. After the user is authenticated, this parameter is used for all subsequent requests to determine if the Agent should validate the incoming request. When set to: `false`: The Agent always validates the incoming request `true`: The default. The Agent determines whether to validate the incoming request based on the following: ■ If the application specifies 'CLIENT-CERT' as part of the construct: "<auth-method>" in its web.xml, the Agent validates the incoming request. ■ If the application does not specify 'CLIENT-CERT' as part of the construct: "<auth-method>" in its web.xml, the Agent does not validate the incoming request. Instead, the Agent lets the request go through to the application.
SSODomains	Specifies legitimate Web servers within the Oracle Access Manager installation to control where the obrar.cgi redirect is sent. Each value describes one or more Web servers. You can provide a relatively short list using domain names and IP addresses with wild cards that cover all your installation's Web servers. See Also: "About SSODomains Parameter" on page 9-23.

maxAuthorizationResultCacheElems	Max Authorization Results Cache Elements—Number of elements maintained in the Authorization Result Cache. This cache maintains information about authorization results for associated user sessions. For example:
	`maxAuthorizationResultCacheElems=10000`
	Default = 100000
	See Also: "Tuning 10g and 11g Webgate Caches" on page 9-31.

authorizationResultCacheTimeout	Authorization Results Cache Timeout—Number of elements maintained in the Authorization Result Cache. This cache maintains information about authorization results for associated user sessions. For example: `authorizationResultCacheTimeout=60` Default, when not specified = 15 (seconds) See Also: "Tuning 10g and 11g Webgate Caches" on page 9-31.

About SSODomains Parameter

Oracle recommends applying the SSODomains user-defined parameter to all Webgate registration to completely mitigate any security risk.

Without this parameter, the host always matches (same as previous Webgate behavior.

If SSODomains is specified but empty, then the host never matches. This allows you to specify that a Webgate will not service any obrareq.cgi requests. Oracle recommends this for all Webgates that are not intended to be authentication servers (as indicated by authentication scheme challenge redirect URLs).

Syntax

```
ssoDomains = ssoDomainSpec | ssoDomains ssoDomainSpec
   ssoDomainsSpec = domainSpec | ipSpec
```

```
domainSpec = domain[:port]
domain = domainName | domainName.domain
domainName = usual_DNS_name
ipSpec = ipPart.ipPart,ipPart,ipPart[:port]
ipPart = ipComponent | ipWildCard
ipComponent = usual_IPaddress_component
ipWildCard = *
port = 12345
```

Guidelines:

domainName = the usual DNS name, alphanumeric characters plus '-', "_', and so on.

ipComponent = the usual IPaddress component, 1-3 digits

ipWildCard = "*"

port = the usual port number, 1-5 digits

For each spec in the SSODomains parameter:

- If the spec has a port, it must match the host port. If the host does not have an explicit host, the default (80 for HTTP, 443 for HTTPS) is used.
- If the host is an IP address and the spec is an ipSpec, match each ipPort of the host and spec from left to right. A wild card "*" in the ipSpec matches all values for the corresponding host. For example, a spec of 130.35.*.* matches a host of 130.35.12.45 but not 130.36.12.45.
- If the host is an DNS name and the spec is a domainSpec, match each domainName of the host and spec from right to left, until the spec has been completely matched. For example, a spec of company.com matches target.company.com and target.us.company.com but not www.badsite.com.
- Continue until a spec matches the host or all specs have been tested.

The right hand parameter in the obrareq.cgi URL is (usually) taken from the original target URL from the user. The user may specify the host as a fully-qualified domain name (the preferred form), an IP address, or a partially or unqualified domain name. The SSODomains spec must take into account the host formats that might be used, hence the need for IP addresses as well as domain names. Note also that partially and unqualified domain names can be specified as domainSpecs.

The burden of covering all hostname variations in the SSODomains parameter can be lessened by configuring Preferred HTTP Hosts for the target Webgates. If SSODomains is also configured for the target Webgate (preferably with no domains to prevent the Webgate from being used for authentication), the (patched) target Webgate will use the preferred host for the right-hand parameter in the obrareq.cgi URL. Consequently the SSODomain for the authenticating Webgate only needs to cover the domains for the preferred hosts.

One good strategy is to include in the SSODomains specs the Primary HTTP Cookie domains defined for each configured Webgate, on the theory the ObSSOCookie will be available to every web server in those domains.

If the right-hand parameter in an obrareq.cgi request does not match any spec in the SSODomains, Webgate returns the following error:

```
Bad Oracle NetPoint Request
The URL /obrareq.cgi is reserved for use by Oracle NetPoint and has been
used with incorrect parameters.
```

Webgate also logs a WARNING "`The rh parameter of a received /obrareq.cgi URL is not allowed by the Webgate's SSODomains parameter`" with the right-hand parameter and the SSODomains values. This means that either someone, potentially an attacker, is misusing the obrareq.cgi URL, or a legitimate obrareq.cgi redirection is not adequately covered by the SSODomains parameter.

9.3.3 About IP Address Validation for Webgates

IP address validation is specific to Webgates. It determines if a client's IP address is the same as the IP address stored in the ObSSOCookie generated for single sign-on. The IPValidation parameter turns IP address validation on and off. If `IPValidation` is `true`, the IP address stored in the ObSSOCookie must match the client's IP address, otherwise, the cookie is rejected and the user must reauthenticate. By default, `IPValidation` is `true`.

The IP Validation parameter can cause problems with certain Web applications. For example, Web applications managed by a proxy server typically change the user's IP address, substituting the IP address of the proxy. This prevents single sign-on using the ObSSOCookie.

Note: The `IP Validation Exceptions` parameter lists IP addresses that are exceptions to this process.

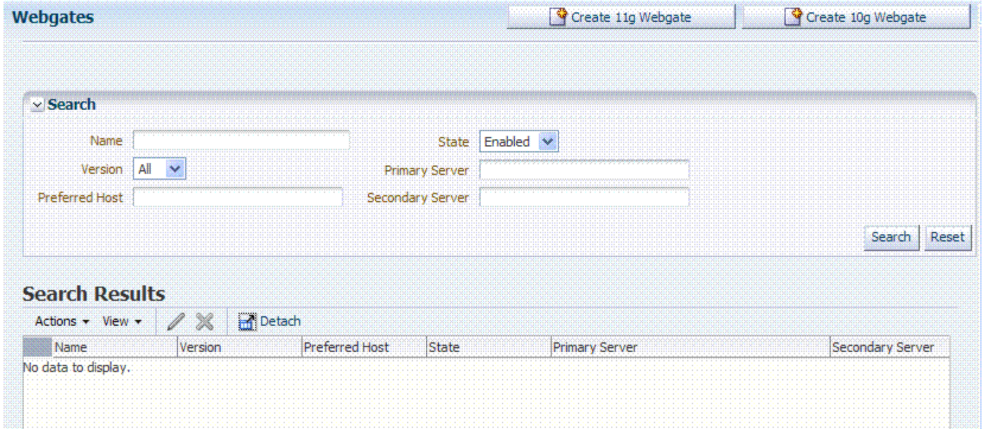
If `IPValidation` is `true`, the IP address can be compared to the `IP Validation Exceptions` list. If the address is found on the exceptions list, it does not need to match the IP address stored in the cookie. You can add as many IP addresses as needed. These addresses are the actual IP addresses of the client, not the IP addresses that are stored in the obSSOCookie. If a cookie arrives from one of the exception IP addresses, the Access System ignores the address stored in the ObSSOCookie cookie for validation. For example, the IP addresses in the IP Validation Exceptions parameter can be used when the IP address in the cookie is for a reverse proxy.

To configure single sign-on between Webgate and an access client that does not have the client IP address at authentication, the IP validation option can be explicitly turned off (set IP Validation to false). When the IP Validation parameter is set to false, the browser or client IP address is not used as a part of the ObSSOCookie. However, Oracle recommends that you keep IP validation on whenever possible.

9.3.4 Searching for an OAM Agent Registration

Figure 9–8 shows the OAM Agent (Webgates) Search controls, defaults, and the empty Search Results table. From this page you can create a new OAM 11g Webgate or 10g Webgate registration, or search for a specific Webgate or group of Webgates (all 11g Webgates, for instance).

Figure 9–8 Webgate Search Controls and Create Agent Buttons



Version: All

State: Enabled

No data to display.

If you do not know the exact name, you can use a wild card (*) in the search string. From the search results table, you can choose an name to open and view or edit the registration page.

The controls available on this page are described in Table 9–7.

Table 9–7 OAM Agent Search Controls

Control	Description
Create 11g Webgate	Click to open a fresh 11g Webgate registration page.
Create 10g Webgate	Click to open a fresh 10g Webgate registration page.
Name	Enter the name (or partial name and wild card (*)) as defined on the registration page. For example: entering a* could return Agent_WebGate_AccessDebugNew in the result table.
Version	Choose a Webgate version to narrow the search and results: ■ 11g ■ 10g
Preferred Host	Enter all (or part of with a wild card (*)) hostname as it appears in HTTP requests. For example: iam* could return IAMSuiteAgent in the result stable.
State	Choose a Webgate version to narrow the search and results: ■ Enabled ■ Disabled
Primary Server	Enter the entire (or partial with a wild card (*)) Primary Server name.
Secondary Server	Enter the entire (or partial with a wild card (*)) Secondary Server name.

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, Access Manager Settings section.
2. Expand the SSO Agents node, and double-click the OAM Agents node.
3. **Find All Enabled**: Select Version All, State All, and click the Search button.

4. **Find a Version**: From the Agent version list, choose 10g or 11g to define your search.
5. **Find a Name**: In the text field, enter the exact name of the instance you want to find. For example:

   ```
   my_OAM_Agent
   ```

6. Click the Search button to initiate the search.
7. Click the Search Results tab to display the results table, and then:
   - **Edit or View:** 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.
   - **Reconfigure Table**: Select a View menu item to alter the appearance of the results table.
8. Apply any changes (or dismiss the page) when you finish.

9.3.5 Registering a Webgate or Programmatic Access Client

You can register a Webgate before you install it. You can register a programmatic Access Client the same as a Webgate. Users with valid Administrator credentials can perform the following task to register an OAM Agent using the Oracle Access Manager Console.

See Also:

- About Creating and Editing Webgate Registration
- 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 or programmatic Access Client

1. From the Oracle Access Manager Console Welcome page, SSO Agent panel, click one of the following to open a fresh page:
   - New OAM 11g Agent
   - New OAM 10g Agent (see also Chapter 27)

Alternatively: From the System Configuration tab, Access Manager Settings section, expand the SSO Agents node, double-click OAM Agents node, and then click the desired Create ... Webgate button in the upper-right corner.

2. On the Create: OAM Agent page, enter required details (those with an *) to register this OAM Agent (Table 9–4).
3. **Protected Resource List**: In this table, enter individual resource URLs to be protected by this OAM Agent, as shown in Table 9–4.
4. **Public Resource List**: In this table, enter individual resource URLs to be public (not protected), as shown in Table 9–4.
5. Confirm that the Auto Create Policies box is checked (or clear the box to disable this function if you do not want a new application domain).
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 27.

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 Oracle Access Manager 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/Access Client: *11gWebgate_instance_dir*/webgate/config/ObAccessClient.xml
(for instance *WebTier_Middleware_Home*/Oracle_WT1/instances1/config/OHS/ohs1/webgate/config/ObAccessClient.xml)

10g Webgate/Access Client: $*Webgate_install_dir*/oblix/lib/ObAccessClient.xml

   - c. Proceed to Part IV, "Managing Oracle Access Manager SSO, Policies, and Testing".

9.3.6 Viewing or Editing an OAM Agent Registration

This procedure is the same whether you are editing a Webgate or Access Client registration. Users with valid Administrator credentials can change any setting for a registered OAM Agent using the Oracle Access Manager 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 Console.

> **See Also:**
>
> - Searching for an OAM Agent Registration
> - About Creating and Editing Webgate Registration

To view or modify details for a registered Webgate

1. From the System Configuration tab, Access Manager Settings section, expand the SSO Agents node.
   a. Double-click OAM Agents node name to display the Search page.
   b. **Find the Registration**: See "Searching for an OAM Agent Registration".
   c. Click the Agent name in the results table to open the page.
2. Modify Agent details, and Primary or Secondary Server details, as needed (Table 9–5).
3. **User Defined Parameters**: Modify these as desired (Table 9–6).
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 27 if you are provisioning an OAM 10g Webgate.

   a. On the Oracle Access Manager 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/Access Client: *11gWebgate_instance_dir*/webgate/config/ObAccessClient.xml
(for instance, WebTier_Middleware_Home/Oracle_WT1/instances1/config/OHS/ohs1/webgate/config/ObAccessClient.xml)

10g Webgate/Access Client: $*Webgate_install_dir*/oblix/lib/ObAccessClient.xml

c. Proceed to Part IV, "Managing Oracle Access Manager SSO, Policies, and Testing".

d. See Appendix E, "Securing Communication for Oracle Access Manager 11g", if needed.

9.3.7 Deleting Webgate Registration

Users with valid Administrator credentials can perform the following procedure to delete a registered OAM Agent from the Oracle Access Manager 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 an OAM Agent Registration
- About Creating and Editing Webgate Registration
- Removing a 10g Webgate from the OAM 11g Deployment in Chapter 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 delete an OAM agent registration

1. From the System Configuration tab, Access Manager Settings section, expand the SSO Agents node.

   a. Double-click OAM Agents node to display the Search page.

   b. **Find the Registration**: See "Searching for an OAM Agent Registration".

2. Optional: Click the Agent name in the results table to open the page, confirm it is the right agent to remove, and 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 27-25, 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/Access Client:

11gWebgate_instance_dir/webgate/config/ObAccessClient.xml

WebTier_Middleware_Home/Oracle_WT1/instances1/config/OHS/ohs1/webgate/

10g Webgate/Access Client:
$*Webgate_install_dir*/oblix/lib/ObAccessClient.xml

9.4 Tuning 10g and 11g Webgate Caches

This section provides the following topics:

- Introducing Webgate Caches
- Reducing Network Traffic Between Components
- Changing the Webgate Polling Frequency

See Also: "Reviewing OAM Agent Metrics" on page 25-5

9.4.1 Introducing Webgate Caches

Webgate caches various information related to resources, authentications and authorizations to improve performance. It uses the cached information to avoid trips to 11g Server for requesting same information. Table 9–8 are the caches used by Webgate to maintain this information.

Table 9–8 Webgate Caches

Cache Type	Description
Resource to Authentication Scheme	This cache maintains information related to authentication schemes being used. Default: 100000 elements See Also: "Tuning Maximum Cache Elements" on page 9-32 and "Tuning Cache Timeout Values" on page 9-33.
Authentication Scheme	This cache maintains information related to authentication schemes being used. Default: 25 elements Typically Authentication Scheme cache elements require less than 2 Kb of memory per element. See Also: "Tuning Cache Timeout Values" on page 9-33.
Resource to Authorization Policy *11g Webgate only*	This cache maintains information related to resources accessed and associated authorization policy. Default: 100000 elements See Also: "Tuning Maximum Cache Elements" on page 9-32 and "Tuning Cache Timeout Values" on page 9-33.

Table 9–8 (Cont.) Webgate Caches

Cache Type	Description
Authorization Result *11g Webgate only*	This cache maintains information related to authorizations associated with user sessions. Default: 1000 elements See Also: "Tuning Authorization Result Cache" on page 9-32 and "Tuning Authorization Result Cache Timeout" on page 9-33.

See Also: "About the 11g Webgate Diagnostics Page"

About the 11g Webgate Diagnostics Page

This page displays useful information related to currently effective Cache configuration parameters. It also displays runtime information about the caches that include information on the number of cached elements, number of hits and misses so far, and current memory usage of individual caches. The page is found at the following URL:

http://*webserver:port*/ohs/modules/webgate.cgi?progid=1

Note: Changes to Webgate parameters are not reflected on Webgate until the next configuration refresh. For 11g Agents, the default configuration refresh interval is 10 minutes.

Tuning Maximum Cache Elements

By default, the Resource to Authentication Scheme and Resource to Authorization Policy caches are created to store 100000 elements. Typically, elements of these caches require less than 1 Kb of memory per element. Therefore, with 100000 elements in each of these caches, typical memory requirement for the caches will be 100000 Kb or 100 Mb each.

Considering memory requirements and your deployment, the Web Server being used and number of unique URLs in your application, you might want to increase or decrease the maximum number of elements to be cached.

Note: Increase or decrease the Maximum Cache Elements parameter value as needed. If this is set to a value of -1, all Webgate caches are disabled.

For both 10g and 11g Webgates, you can tune the maximum number of elements to be cached property, by changing the Maximum Cache Elements parameter. Updates to this parameter require a Webgate restart.

1. Locate and open the desired 10g or 11g Webgate registration page in the Oracle Access Manager Console.
2. Set the Maximum Cache Elements parameter as desired.
3. Restart Webgate Web server.

Tuning Authorization Result Cache

By default, the Authorization Result cache is created to store 1000 elements. Authorization Result cache elements store the user session identifier, authorization policy identifier, and associated authorization result including any processed policy responses. Therefore, Authorization Result cache elements are bulky and generally require more than 2Kb of memory per element.

Considering memory requirements and the number of concurrent user sessions in your deployment, you might want to increase the number of elements to be cached.

1. Locate and open the desired 11g Webgate registration page in the Oracle Access Manager Console.
2. In User Defined Parameters, add or update maxAuthorizationResultCacheElems as desired.
3. Restart Webgate Web server.

Tuning Cache Timeout Values

By default, the following caches are created with a timeout value of 1800 seconds or 30 minutes:

- Resource to Authentication Scheme
- Authentication Scheme
- Resource to Authorization Policy

Elements in these caches are stored with an expiry time that forces these caches to be flushed on expiry.

Considering the frequency of updates to Authentication Schemes, and Authentication and Authorization Policies in your deployment, you might want to increase or decrease the default timeout value.

1. Locate and open the desired 10g or 11g Webgate registration page in the Oracle Access Manager Console.
2. Set the Cache Timeout parameter as desired.
3. Restart Webgate Web server.

Tuning Authorization Result Cache Timeout

By default, the Authorization Result Cache timeout value is set at 15 seconds. Elements in the Authorization Result Cache is stored with an expiry time that forces it to be flushed on expiry. A low timeout value ensures that authorization results are cached for a small amount of time only.

Considering average length of user sessions and frequency with which user sessions are created and destroyed, you might want to change the default timeout value. Unlike other caches and parameters, updates to this parameter do not require Webgate restart. Instead, the updated value is dynamically picked up by 11g Webgate and enforced immediately.

> **Note:** If authorizationResultCacheTimeout is set to 0, Authorization Cache is disabled.

1. Locate and open the desired 11g Webgate registration page in the Oracle Access Manager Console.
2. In User Defined Parameters, add or update authorizationResultCacheTimeout as desired.
3. Restart Webgate Web server.

9.4.2 Reducing Network Traffic Between Components

The Webgate-to-OAM Server configuration polling reduces the traffic between both the Webgate and OAM Server and the OAM Server and the registered data stores for Oracle Access Manager.

Process overview: Webgate-to-OAM Server configuration polling

1. When the Webgate is inactive for 60 seconds, it reduces the frequency of polling for its configuration information.

   The polling frequency is determined by the parameter InactiveReconfigPeriod, which is a user-defined parameter that is set in the Webgate configuration page. The value for InactiveReconfigPeriod is specified in minutes. Within ten seconds of resuming activity, the Webgate performs reconfiguration polling once a minute.

2. At startup, the Webgate checks the bootstrap configuration to see if any important parameters have changed.

   This makes the re-initialization process unnecessary in most cases and reduces the transient OAM Server load.

3. Webgate and Access client configurations are cached in the OAM Server.

   The default cache timeout is 59 seconds. This should cause no modifications to the system behavior on non-Apache Access clients. The Apache Web server with Webgate avoids unnecessary hits to the directory server. The caching parameters can be set in the Webgate registration page.

   - Max Cache Elements sets the maximum size of the cache (default 9999)
   - Cache Timeout determines the maximum lifetime of any element in the cache (default 59 seconds)

There are two ways to reduce off-time network traffic between both the Webgate and OAM Server and the OAM Server and the database:

- Changing the default configuration cache timeout for Webgate and Access client configurations that are cached in the OAM Server, as described in Step 3.
- Changing Webgate polling frequency for configuration information, as described next.

9.4.3 Changing the Webgate Polling Frequency

One way to reduce off-time network traffic between both the Webgate and OAM Server and between the OAM Server and the database is to change the Webgate polling frequency using the InactiveReconfigPeriod parameter.

The default is 1 minute. When the Webgate is inactive for more than 60 seconds (for example, when no authentication requests are being processed), it reduces the frequency of polling for its configuration information. Within ten seconds of resuming activity, the Webgate resumes reconfiguration polling once every minute:

- If set to -2, Webgate never polls.
- If set to a value greater than 0 it polls at the specified interval.
- If set to -1 and Webgate is inactive and has been for 1 minute, then Webgate does not poll. Webgate resumes reconfiguration polling when it returns to an active state.

For example, the OAM Server reads the shared secret from the directory at an interval of 10 minutes and this cached value is returned to Webgate. In the idle state the

Webgate reads the shared secret from the OAM Server using the InactiveReconfigPeriod value. If this value is not set, the Webgate polls the OAM Server for the shared secret value at an interval of 1 minute even though the updated shared secret value will be returned only after 10 minutes.

To change the configuration polling frequency

1. Locate the desired Webgate registration page using instructions in "Searching for an OAM Agent Registration" on page 9-25.
2. Add the InactiveReconfigPeriod parameter as a user-defined parameter on the Webgate registration page.
3. Specify the value for InactiveReconfigPeriod in minutes.
4. Apply your changes to the Webgate registration page.

9.5 Registering and Managing OSSO Agents Using the Console

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

- About OSSO Agents and the OSSO Proxy
- About the Create OSSO Agent Page
- Refining the Search 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

9.5.1 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 inter-operability 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.

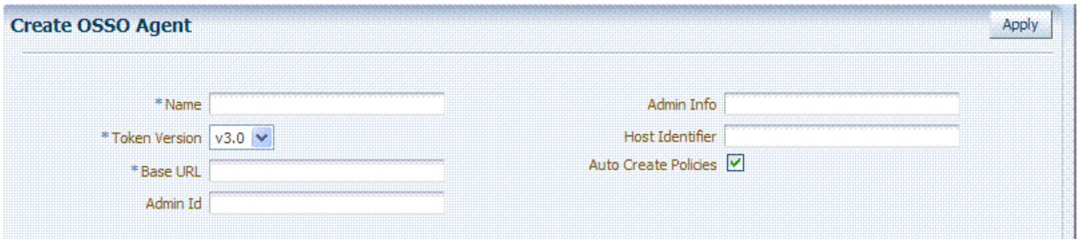
9.5.2 About the Create OSSO Agent Page

This topic describes OSSO Agent registration using the Oracle Access Manager 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 9–9 shows a Create OSSO Agent page, under the System Configuration tab in the Oracle Access Manager Console.

Figure 9–9 Create OSSO Agent Page



Create OSSO Agent

*Name		Admin Info	
*Token Version	v3.0	Host Identifier	
*Base URL		Auto Create Policies	☑
Admin Id			

On the Create OSSO Agent page, required information is identified by the asterisk (*). Table 9–9 describes the required and optional details that you can specify when you register a new agent.

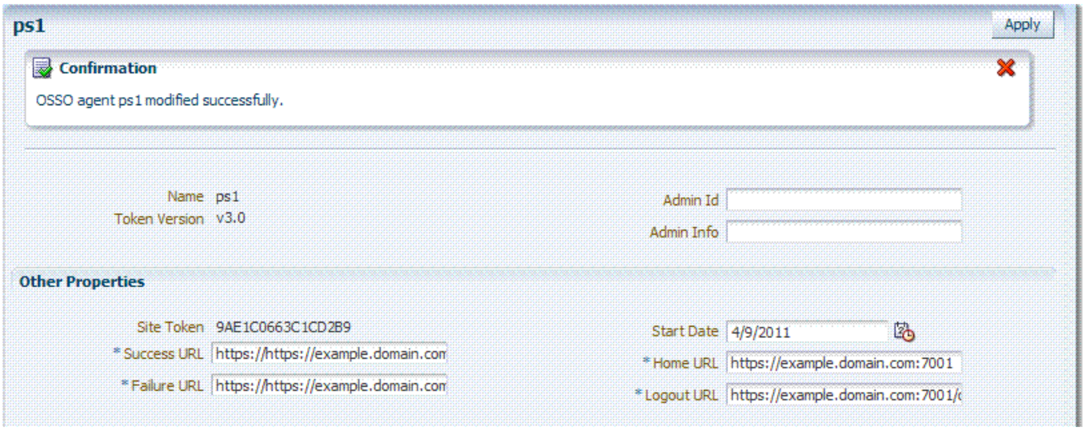
Table 9–9 Create OSSO Agent Page Elements

Element	Description
Name	The identifying name for this mod_osso Agent.
Token Version	The default version of the token is 3.0; the following options are available: ■ 1.2 ■ 1.4 ■ 3.0
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.example.domain.com:port* or https://*example.domain.com:port*. Note: The host and port are used as defaults for the expanded registration. See Table 9–10.
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
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 Oracle Access Manager Console, all elements and values are revealed.

Figure 9–10 shows the full Agent page as viewed in the console. The Confirmation window is still visible. All details specified, and defaults, are shown.

Figure 9–10 OSSO Agent Page and Confirmation Window



OSSO agent ps1 modified successfully.

Name	ps1	Admin Id	
Token Version	v3.0	Admin Info	

Site Token	9AE1C0663C1CD2B9	Start Date	4/9/2011
* Success URL	https://https://example.domain.con	* Home URL	https://example.domain.com:7001
* Failure URL	https://https://example.domain.con	* Logout URL	https://example.domain.com:7001/(

Table 9–10 summarizes the expanded elements and defaults that are used by the OSSO Agent.

Table 9–10 Expanded OSSO Agent Elements

Element	Description
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 Base URL are used. For example: Default: http://*example.domain.com:7001*/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://*example.domain.com:7001*/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.
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://*example.domain.com:7001*
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://*example.domain.com:7001*/osso_logout_success See Also: "Introduction to OAM 11g Centralized Logout" on page 15-2.

9.5.3 Refining the Search for an OSSO Agent (mod_osso) Registration

When you first open the OSSO Agents node, the Search form appears. The Results table lists all OSSO Agents. If there are too many to quickly locate the one you want, you can use the controls to refine your search.

> **Note:** At the top of the Search page is the Create OSSO Agent button.

There are only two element on which you can refine an OSSO Agent search: The Agent Name that assigned during registration or the Agent ID assigned by the system.

Prerequisites

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

To search for an OSSO Agent registration

1. Activate the System Configuration tab, Access Manager section.
2. Expand the SSO Agents node, and open the OSSO Agents node.
3. In the Name field, enter some criteria for your search (with or without including the wild card (*)). For example:

   ```
   my*
   ```

4. Click the Search button to initiate the search.
5. In the Search Results table:
   - **Edt or View:** 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.
   - **Reconfigure Table**: Select a View menu item to alter the appearance of the results table.
6. Apply any changes (or dismiss the page) when finished.

9.5.4 Registering an OSSO Agent (mod_osso)

Users with valid Administrator credentials can perform the following procedure to register an OSSO Agent using the Oracle Access Manager 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 10 for details about using the remote command-line tool

To register an OSSO Agent

1. From the Oracle Access Manager Console 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.
2. Click the Create button in the tool bar.
3. On the Create: OSSO Agent page, enter required details, as shown in Table 9–9:
   - Name
   - Base URL

4. Select the desired Token Version, and enter optional details as desired (Table 9–9).
5. Click Apply to submit the registration (or close the page without applying changes).
6. 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
   ```

7. Copy the updated osso.conf file to the OSSO Agent host:
   a. On the Oracle Access Manager 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.
8. Proceed to Part IV, "Managing Oracle Access Manager SSO, Policies, and Testing".

9.5.5 Viewing or Editing OSSO Agent (mod_osso) Registration

Users with valid Administrator credentials can change any setting for a registered OSSO Agent using the Oracle Access Manager 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:**
>
> - Refining the Search 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, Access Manager section, expand the SSO Agents node.
2. Double-click the OSSO Agents node.
3. Find the Agent: "Refining the Search for an OSSO Agent (mod_osso) Registration"
4. In the Search Results table, click the desired Agent name to open the registration page.
5. On the registration page, view or modify details as needed (Table 9–9 and Table 9–10).
6. Click Apply to submit the changes (or close the page without applying changes), and close the Confirmation window.
7. Copy the updated osso.conf file to the OSSO Agent host:

a. On the Oracle Access Manager 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 IV, "Managing Oracle Access Manager SSO, Policies, and Testing":

9.5.6 Deleting an OSSO Agent (mod_osso) Registration

Users with valid Administrator credentials can perform the following procedure to delete a registered OSSO Agent from the Oracle Access Manager 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:** Refining the Search for an OSSO Agent (mod_osso) Registration

To delete an OSSO Agent registration

1. From the System Configuration tab, Access Manager Settings section, expand the SSO Agents node, nd open the OSSO Agents node.
2. Click Search and choose the desired name (or refine your search as needed).
3. Optional: Double-click the desired 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.

10

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
- Introducing Remote Management Modes
- Managing Agents Remotely
- Creating or Updating an Application Domain Without an Agent

10.1 Prerequisites

Before you can perform tasks in this chapter, ensure that an Oracle Access Manager Console and a managed OAM Server are running.

> **See Also:** Chapter 27 for information about registering and using 10g Webgates with Oracle Access Manager 11g

10.2 Introduction to Remote Partner Registration

Supported policy enforcement agents must be registered with Oracle Access Manager 11g to communicate with Oracle Access Manager authentication and authorization services. A partner application (one that delegates the authentication function to the Oracle Access Manager 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 Oracle Access Manager 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 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 Request Files
- About Out-of-Band Registration Responses

10.2.1 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 Administrators in the Default System User Identity Store registered with Oracle Access Manager.

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 10-22.
2. Update the input file with unique values for the agent and application domain as described in "Creating the Registration Request" on page 10-23.
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 10-24.
4. Validate the configuration as described in "Validating Remote Registration and Resource Protection" in on page 10-26.
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 10-27.

10.2.2 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 10-22.

  - Copy and edit a template to input unique values for the agent and application domain as described in "Creating the Registration Request" on page 10-23.
  - 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 10-22.
   - 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 10-24 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 10-26.
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 10-27.

10.2.3 About Key Use, Generation, Provisioning, and Storage

Each registered agent has a symmetric key, regardless of the registration method (Oracle Access Manager 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

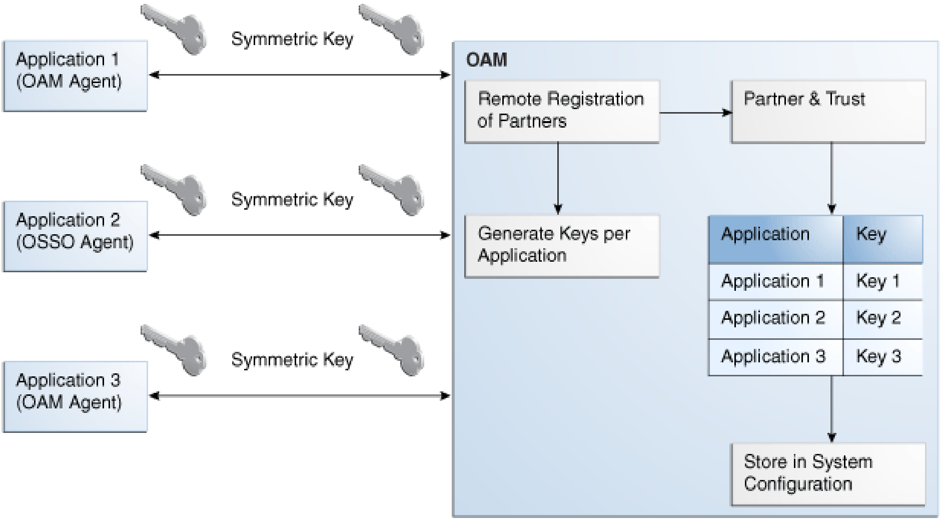
Each Webgate agent has its own secret key that is shared between the agent and the OAM 11g server. If one Webgate is compromised, other Webgates are unaffected. The following presents an overview:

- 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 10–1 illustrates the process of key generation, which occurs automatically when the agent is registered, regardless of the method used (Oracle Access Manager Console versus remote registration). There is one symmetric key per agent.

Figure 10–1 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
Store in System Configuration

Application	Key
Application 1	Key 1
Application 2	Key 2
Application 3	Key 3

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.

10.2.4 About the Remote Registration Tool

This topic provides an overview of the registration tool, requirements, usage, and results.

Location

The registration tool, oamreg, is located in:

<*OAM_HOME*>/oam/server/rreg/client/RREG.tar.gz

After untarring RREG.tar.gz on the computer hosting the Agent, the registration tool, oamreg, resides in the following path:

<*OAM_HOME*>/oam/server/rreg/client/rreg/bin/oamreg

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 the script as described here:

Environment Variable	Description
OAM_REG_HOME	The directory under which RREG.tar was exploded, followed by /rreg: <*OAM_HOME*>/oam/server/rreg/client/rreg
JAVA_HOME	The location where Java is located on the client computer. For example: *WLS_HOME*/Middleware/jdk160_11. Note: JAVA_HOME should point to JDK 1.6.

In addition, you must modify several tags in the Registration request. For details, see "About Remote Registration Request Files" on page 10-9.

Registration Administrators

The user can be a part of any group that is mapped against the Administrator's Role in the primary user-identity store. For more information, see Table 5–2, " 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 described in the previous "Requirements" table.

Agent Types and Associated Request Files

Both `inband` and `outofband` modes use a request file with the input argument, as follows (as described in the previous "Requirements" table):

Table 10–1 Remote Registration Request Files

Agent Type	Request File
10g Webgates	$OAM_REG_HOME/input/OAMRequest_short.xml $OAM_REG_HOME/input/OAMRequest.xml $OAM_REG_HOME/input/OAMUpdateAgentRequest.xml
11g Webgates	$OAM_REG_HOME/input/OAM11GRequest.xml $OAM_REG_HOME/input/OAM11GRequest_short.xml $OAM_REG_HOME/input/OAM11GUpdateAgentRequest.xml
OSSO Agents (mod_osso)	$OAM_REG_HOME/input/OSSORequest.xml $OAM_REG_HOME/input/OSSOUpdateAgentRequest.xml
Create New Host Identifiers and an Application Domain without Registering an Agent	$OAM_REG_HOME/input/CreatePolicyRequest.xml See Also: "Managing Agents Remotely" on page 10-39
Update existing Host Identifiers and Application Domain not associated with an Agent Registration	$OAM_REG_HOME/input/UpdatePolicyRequest.xml See Also: "Managing Agents Remotely" on page 10-39

See Also:

- "About Remote Registration Request Files" on page 10-9
- "Introducing Remote Management Modes" on page 10-29
- "About Remote Application Domain Management Modes" on page 10-32

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 10-22

Sample Remote Registration Commands and Results

Sample commands are shown in Table 10–2 and presume the location of the tool to be $*OAM_REG_HOME* (previous "Requirements" table) on a Linux system.

Table 10–2 Remote Registration Sample Commands

Command Type	Sample (on Linux)
In-band Administrator In-band Request	`./bin/oamreg.sh inband input/*Request.xml`

Table 10–2 (Cont.) Remote Registration Sample Commands

Command Type	Sample (on Linux)
In-band Administrator Submitted Request	`./bin/oamreg.sh outofband input/`*`starting_request.xml`*
Out-of-band Administrator Returned Response	`./bin/oamreg.sh outofband input/agentName_Response.xml`

[prompt_flag] value:
[-noprompt]
	Optional. When -noprompt is used, oamreg does not wait for prompts (password, and so on). Instead these values can be piped in, either from an input file or from the command line itself using an echo command.
	Examples from OAM_REG_HOME location:
	`(echo username; echo password; echo webgate_password;) | ./bin/oamreg.sh inband input/Request.xml -noprompt config.file`
	
	`(echo username; echo password; echo webgate_password; echo httpscert_trust_prompt;) | ./bin/oamreg.sh inband input/Request.xml -noprompt`
	
	`(echo username; echo password; echo webgate_password; echo cert_password;) | ./bin/oamreg.sh inband input/Request.xml -noprompt`
	
	`(echo username; echo password; echo webgate_password; echo httpscert_trust_prompt; echo cert_password;) | ./bin/oamreg.sh inband input/Request.xml -noprompt`
	"Managing Agents Remotely" on page 10-39

After launching the script, administrators are prompted for a username and password (unless -noprompt is used as described in Table 10–2. 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 10–3.

Table 10–3 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/Access Client: 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/Access Client: 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/Access Client: 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/Access Client: 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 10-24
- "Performing Out-of-Band Remote Registration" on page 10-25
- "Managing Agents Remotely" on page 10-39

10.2.5 About Remote Registration Request Files

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

> **See Also:** "Introducing Remote Management Modes" on page 10-29

10.2.5.1 OSSO Remote Registration Request

Example 10–1 provides 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 10-11

Example 10–1 OSSORequest.xml

```
. . .
<OSSORegRequest>
    <serverAddress>http://{oam_admin_server_host}:{oam_admin_server_port}
        <http://%7boam_admin_server_host%7d:%7boam_admin_server_port%7d>
    </serverAddress>
    <hostIdentifier>RREG_HostId</hostIdentifier>
    <agentName>RREG_OSSO</agentName>
    <agentBaseUrl>http://{web_server_host}:{web_server_port}
        <http://%7bweb_server_host%7d:%7bweb_server_port%7d>
    </agentBaseUrl>
    <applicationDomain>RREG_OSSO</applicationDomain>
    <autoCreatePolicy>true</autoCreatePolicy>
    <ssoServerVersion>v3.0</ssoServerVersion>
    <oracleHomePath>$ORACLE_HOME</oracleHomePath>
    <virtualHost></virtualHost>
    <updateMode></updateMode>
    <adminInfo></adminInfo>
    <adminId></adminId>
    <logoutUrl></logoutUrl>
    <failureUrl></failureUrl>
</OSSORegRequest>
```

10.2.5.2 Short, Simplified OAM Remote Registration Requests

Example 10–2 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 10–2, 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 10–2 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>
         <excludedResourcesList>
              <resource>/excluded/index.html</resource>
         </excludedResourcesList>
</OAMRegRequest>
```

10.2.5.3 Common Elements of Remote Registration Requests

Unless otherwise stated, Table 10–4, explains the global elements within all request files.

Table 10–4 Elements Common to Remote Registration Requests

Element	Description	Example
<serverAddress>	Points to a running instance of the Oracle Access Manager 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. A unique identifying name for each Agent registration is preferred. However: ■ If the Agent Name exists, no error occurs and the registration does not fail. Instead, Oracle Access Manager creates the policies if they are not already in place. ■ If the host identifier exists, the unique Agent Base URL is added to the existing host identifier and registration proceeds.	<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>

10.2.5.4 OSSO-Specific Elements in a Remote Registration Request

Table 10–5 describes the remote registration elements that are OSSO-specific.

Table 10–5 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>`
<adminId>	Optional administrator log in ID for this mod_osso instance. For example, *SiteAdmin*. Default: None specified >	`<adminId></adminId>`
<logoutUrl>	Include the Logout URLs for consumption during remote registration. Default: None specified >	`<logoutUrl>logout1.html</logoutUrl>`
<failureUrl>	Include the Failure URLs for consumption during remote registration. Default: None specified >	`<failureUrl>failure1.html</failureUrl>`

10.2.5.5 Full OAM Remote Registration Requests

Table 10–6 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 10–4):

- 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 10–6 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>`

<virtualhost>	Specifies whether this is a virtual host. Values: true or false Default: false See Also: "About Virtual Web Hosting" on page 12-7.	`<virtualhost>false<virtualhost>`

<hostportVariations>	Specifies all variations of a particular host.
	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.
	See Also: "About Host Identifiers" on page 12-5.
		`<hostPortVariationsList>`
		`<host>host1</host>`
		`<port>7777</port>`
		`</hostPortVariations>`
		`<host>host2</host>`
		`<port>7778</port>`
		`</hostPortVariations>`
		`</hostPortVariationsList>`

<applicationDomain>	Defines the name of the application domain, which is based on the specified Agent Name (see Table 10–4).	`<applicationDomain>RREG_OAM11G`
		`</applicationDomain>`

<autoCreatePolicy>
Absent in OAM 11g Short Version
	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>`

<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>`

Table 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

<excludedresourcesList>	Specifies the HTTP type 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.
	Only HTTP resource types can be excluded. Typically security insensitive files like Images (*.jpg, *.png) that do not require Authentication, Authorization, Response processing, Session management, and Auditing. Excluded resources cannot be added to any user-defined policy in the console.
	See Also: Table 13–1, " Resource Definition Elements" for more information on excluded resource lists.
		`<excludedresourcesList>`
		`   <resource>/excluded/index.html`
		`   </resource>`
		`</excludedresourcesList>`

<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 Agentuses 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>`

<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>`

Table 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example
<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</maxSessionTime>`
<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 patch set. ■ v1.4: This is supported by OSSO 10g prior to OSSO 10.1.4.3 patch set. 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`
<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>`

Table 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

<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 9–5 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 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example
<allowManagementOperations>	This Agent Privilege function enables the provisioning of session operations per agent, as follows: ■ Terminate session ■ Enumerate sessions ■ Add or Update attributes for an existing session ■ List all attributes for a given session ID or read session Default: false Note: Only privileged agents can invoke session management operations. When this parameter is enabled, session management requests (listed above) are processed by the OAM Server. If disabled, such requests are rejected for the agent.	<allowManagementOperations> false/<allowManagementOperations>
<cachePragmaHeader> <cacheControlHeader>	These settings apply only to Webgates and control the browser's cache. By default, both 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>

<ipValidationExceptions>	Exceptions to IP address validation.	<ipValidationExceptions>
		  <ipAddress>10,11,11,11</ipAddress>
		  <ipAddress>10,11,11,12</ipAddress>
		  <ipAddress>10,11,11,13</ipAddress>
		</ipValidationExceptions>

Table 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

<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 15 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 15 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 15 for steps to configure logout for OAM 11g Webgates.
		`<logoutTargetUrlParamName>end_url`
		`</logoutTargetUrlParamName>`

User-Defined Parameter Names	**Descriptions**	**Examples**

	■ Available for configuration in the remote registration request files only
	■ Each parameter can have only one value.
	■ User-defined parameters cannot be set using the Oracle Access Manager 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 29 for more information about configuring IIS Web servers for Webgate.
		`<name>MaxPostDataLength</name>`
		`<value>750000</value>`

Table 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

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>`

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>`

Table 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

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, 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>`

Table 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

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>`

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>`

Table 10–6 (Cont.) Elements Common to Full Remote Registration Requests

Element	Description	Example

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 29.
	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 29-9.
		`<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>`

10.2.6 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.

10.3 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.

Note: Oracle Recommends using the latest tool and files release by applying the latest bundle patch and untarring RREG.tar.gz again.

For remote registration, two variables are required: JAVA_HOME and OAM_REG_HOME, as described in Table 10–7.

Table 10–7 Variables Required for Remote Registration

Location	Variable	Description
Client Side	JAVA_HOME	The JDK 1.6 location on the computer that relies on $JAVA_HOME already set in the environment.
	OAM_REG_HOME	The absolute file location for RREG HOME (directory under which RREG.tar was exploded, followed by /rreg and one directory above where the scripts reside). For example: *<OAM_HOME>*/oam/server/rreg/client/rreg If IM_ORACLE_HOME is MW_HOME/Oracle_IDM: export OAM_REG_HOME=MW_HOME/Oracle_IDM/oam/server/rreg
rreg folder location (not RREG.tar.gz location)	JAVA_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 10-6

To acquire the tool and update the script with your environment variables

1. Locate RREG.tar.gz file in the following path:

```
ORACLE_HOME/oam/server/rreg/client/RREG.tar.gz
```

2. Untar RREG.tar.gz file, which creates directories beneath `/client` containing the required tool and templates.
3. In the oamreg script (.../`rreg/client/rreg/bin`) set environment variables as follows:
   a. Set JAVA_HOME to JDK 1.6 (Table 10–7).
   b. Set OAM_REG_HOME to the *exploded_dir_for_RREG*.tar/rreg based on your environment (client side or server side Table 10–7).
4. Proceed with "Creating the Registration Request".

10.4 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 10-6
- "About Remote Registration Request Files" on page 10-9

To create the registration request

1. Locate the required *Request*.xml input file for the agent you want to register (Table 10–1).

   ```
   ORACLE_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 10–4 and Table 10–6):
4. Proceed with:
   - Performing In-Band Remote Registration
   - Performing Out-of-Band Remote Registration

10.5 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 Oracle Access Manager Console to register the Agent and add an Application domain, as described in Chapter 9 and Chapter 13, 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 27, "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/Access Client: $*WG_install_dir*/oblix/lib/ObAccessClient.xml

      11g Webgate/Access Client: $*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 10-26.

10.6 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/Access Client: $*WG_install_dir*/oblix/lib/ObAccessClient.xml

   11g Webgate/Access Client: $*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 10-26.

10.7 Validating Remote Registration and Resource Protection

This section provides the following topics:

- Validating Remote Registration
- Validating Authentication, Resource Protection, and Access After Remote Registration

10.7.1 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 Oracle Access Manager Console. Out-of-band Administrators must test authentication and access remotely.

See Also:

- Chapter 9, "Registering Partners (Agents and Applications) by Using the Console"
- Chapter 13, "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 Oracle Access Manager Console, as described in Chapter 9 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 Oracle Access Manager 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".

10.7.2 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 Oracle Access Manager 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 13, "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.

10.8 Introducing Remote Management Modes

This section provides the following topics:

- About Remote Agent Management Modes
- About Remote Application Domain Management Modes
- About <rregApplicationDomain> Elements

10.8.1 About Remote Agent Management Modes

Several remote management modes enable Administrators to update, or validate, or delete an existing agent registration. Table 10–8 presents remote agent management modes. Command parameters include the mode, input *Request.xml file (a relative path with respect to OAM_REG_HOME, the preferred location for the input *Request.xml files):

```
./oamreg.sh <mode> <input_file> [prompt_flag] [component.oam.config_file] <mode>
value
```

Table 10–8 Remote Agent and Policy Updates

Mode and Input Files	Description and Syntax
agentUpdate mode *OSSOUpdateAgentRequest.xml* *OAM11GUpdateAgentRequest.xml* *OAMUpdateAgentRequest.xml*	Allows Administrators to update agent attributes: `./bin/oamreg.sh agentUpdate input/OAM11GUpdateAgentRequest.xml`
agentValidate mode *No input file needed.*	Validates whether the agent is already provisioned in Oracle Access Manager 11g: `./bin/oamreg.sh agentValidate` *`agentname`*

Table 10–8 (Cont.) Remote Agent and Policy Updates

Mode and Input Files	Description and Syntax
agentDelete mode *No input file needed.*	Allows Administrators to delete the agent registration: `./bin/oamreg.sh agentDelete` *`agentname`*

For a look at the templates for the agentUpdatemode, see:

- OSSOUpdateAgentRequest.xml
- OAM11GUpdateAgentRequest.xml
- OAMUpdateAgentRequest.xml

10.8.1.1 OSSOUpdateAgentRequest.xml

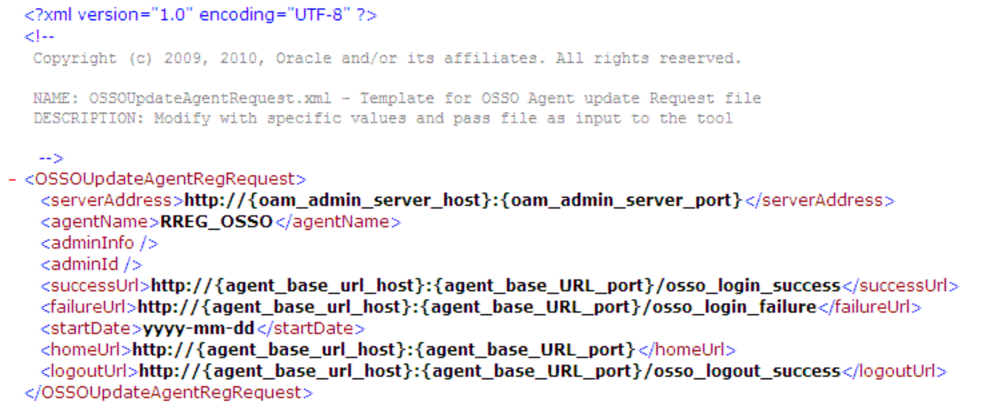
You use OSSOUpdateAgentRequest.xml to pass specific values to the remote registration tool, oamreg. The primary differences between the update template and the original registration template is that the update template:

- Provides a <startDate>*yyyy_mm_dd*</startDate> element to track changes
- Provides a <homeUrl> element that specifies the agent_base_url_port
- Omits the <hostidentifier> element
- Omits the <agentbaseURL> element

> **See Also:** Table 10–5, " OSSO-Specific Elements in a Remote Registration Request" for details about elements and values

Figure 10–2 illustrates the update template for the OSSO Agent.

Figure 10–2 OSSOUpdateAgentRequest.xml



```
  <?xml version="1.0" encoding="UTF-8" ?>
  <!--
   Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.

   NAME: OSSOUpdateAgentRequest.xml - Template for OSSO Agent update Request file
   DESCRIPTION: Modify with specific values and pass file as input to the tool

    -->
- <OSSOUpdateAgentRegRequest>
    <serverAddress>http://{oam_admin_server_host}:{oam_admin_server_port}</serverAddress>
    <agentName>RREG_OSSO</agentName>
    <adminInfo />
    <adminId />
    <successUrl>http://{agent_base_url_host}:{agent_base_URL_port}/osso_login_success</successUrl>
    <failureUrl>http://{agent_base_url_host}:{agent_base_URL_port}/osso_login_failure</failureUrl>
    <startDate>yyyy-mm-dd</startDate>
    <homeUrl>http://{agent_base_url_host}:{agent_base_URL_port}</homeUrl>
    <logoutUrl>http://{agent_base_url_host}:{agent_base_URL_port}/osso_logout_success</logoutUrl>
  </OSSOUpdateAgentRegRequest>
```

10.8.1.2 OAM11GUpdateAgentRequest.xml

You use OAM11GUpdateAgentRequest.xml to pass specific Agent-update values to the remote registration tool, oamreg. The primary differences between the update template and the original registration template is that the update template:

- Provides the <ipValidation> element but omits <ipValidationExceptions>
- Omits the <authCreatePolicy> and application domain-related elements

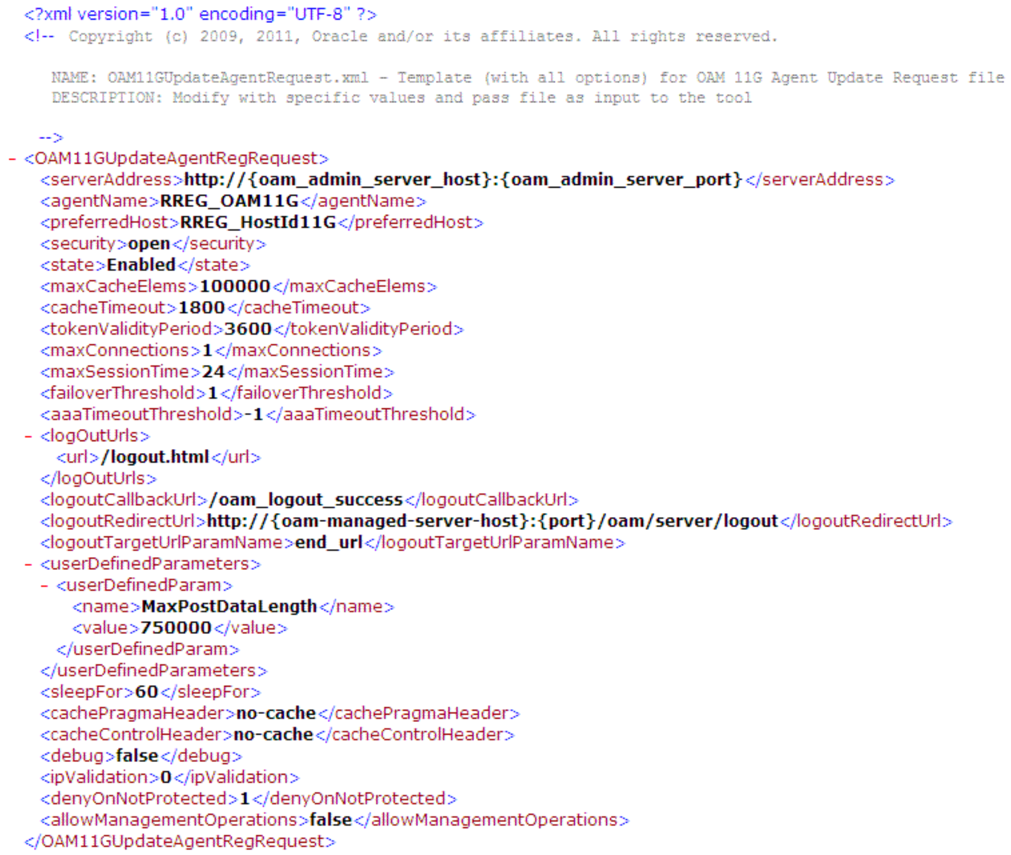
- Omits the <hostidentifier>, <virtualhost>, and <hostportVariations> elements
- Omits the <agentbaseURL> element
- Omits the <ssoServerVersion> element
- Omits the <idleSessionTimeout> element

See Also:

- Table 10–10, " <rregApplicationDomain> Remote Management Template Elements"
- Table 10–6, " Elements Common to Full Remote Registration Requests" for details about elements and values

Figure 10–2 illustrates the update template for the OAM 11g Agent.

Figure 10–3 OAM11GUpdateAgentRequest.xml



```
  <?xml version="1.0" encoding="UTF-8" ?>
  <!--  Copyright (c) 2009, 2011, Oracle and/or its affiliates. All rights reserved.

    NAME: OAM11GUpdateAgentRequest.xml - Template (with all options) for OAM 11G Agent Update Request file
    DESCRIPTION: Modify with specific values and pass file as input to the tool

   -->
- <OAM11GUpdateAgentRegRequest>
    <serverAddress>http://{oam_admin_server_host}:{oam_admin_server_port}</serverAddress>
    <agentName>RREG_OAM11G</agentName>
    <preferredHost>RREG_HostId11G</preferredHost>
    <security>open</security>
    <state>Enabled</state>
    <maxCacheElems>100000</maxCacheElems>
    <cacheTimeout>1800</cacheTimeout>
    <tokenValidityPeriod>3600</tokenValidityPeriod>
    <maxConnections>1</maxConnections>
    <maxSessionTime>24</maxSessionTime>
    <failoverThreshold>1</failoverThreshold>
    <aaaTimeoutThreshold>-1</aaaTimeoutThreshold>
  - <logOutUrls>
      <url>/logout.html</url>
    </logOutUrls>
    <logoutCallbackUrl>/oam_logout_success</logoutCallbackUrl>
    <logoutRedirectUrl>http://{oam-managed-server-host}:{port}/oam/server/logout</logoutRedirectUrl>
    <logoutTargetUrlParamName>end_url</logoutTargetUrlParamName>
  - <userDefinedParameters>
    - <userDefinedParam>
        <name>MaxPostDataLength</name>
        <value>750000</value>
      </userDefinedParam>
    </userDefinedParameters>
    <sleepFor>60</sleepFor>
    <cachePragmaHeader>no-cache</cachePragmaHeader>
    <cacheControlHeader>no-cache</cacheControlHeader>
    <debug>false</debug>
    <ipValidation>0</ipValidation>
    <denyOnNotProtected>1</denyOnNotProtected>
    <allowManagementOperations>false</allowManagementOperations>
  </OAM11GUpdateAgentRegRequest>
```

10.8.1.3 OAMUpdateAgentRequest.xml

You use OAMUpdateAgentRequest.xml to pass specific OAM 10g Agent-update values to the remote registration tool, oamreg. The primary differences between this update template and the original OAM 10g Agent registration template is that the update template:

- Provides the <ipValidation> element but omits <ipValidationExceptions>

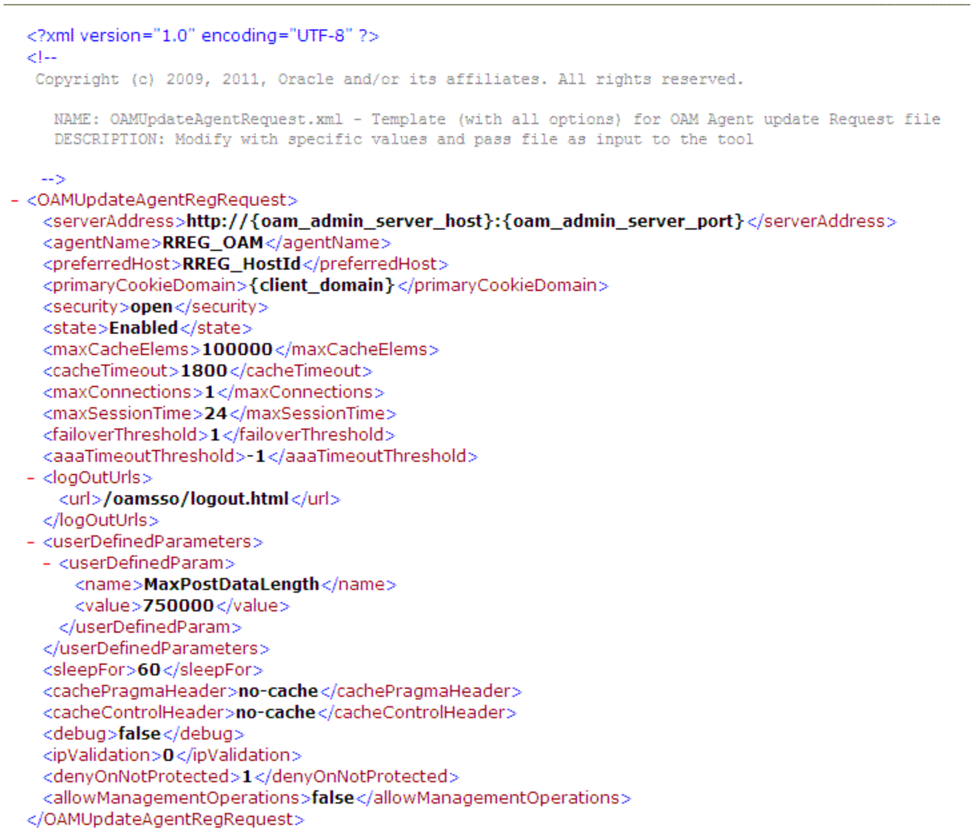
- Omits the <authCreatePolicy> and application domain-related elements
- Omits the <hostidentifier>, <virtualhost>, and <hostportVariations> elements
- Omits the <agentbaseURL> element
- Omits the <ssoServerVersion> element
- Omits the <idleSessionTimeout> element

See Also:

- Table 10–10, " <rregApplicationDomain> Remote Management Template Elements"
- Table 10–6, " Elements Common to Full Remote Registration Requests" for details about elements and values

Figure 10–4 illustrates OAMUpdateAgentRequest.xml for OAM 10g Agents.

Figure 10–4 OAMUpdateAgentRequest.xml



```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
 Copyright (c) 2009, 2011, Oracle and/or its affiliates. All rights reserved.

   NAME: OAMUpdateAgentRequest.xml - Template (with all options) for OAM Agent update Request file
   DESCRIPTION: Modify with specific values and pass file as input to the tool

  -->
- <OAMUpdateAgentRegRequest>
    <serverAddress>http://{oam_admin_server_host}:{oam_admin_server_port}</serverAddress>
    <agentName>RREG_OAM</agentName>
    <preferredHost>RREG_HostId</preferredHost>
    <primaryCookieDomain>{client_domain}</primaryCookieDomain>
    <security>open</security>
    <state>Enabled</state>
    <maxCacheElems>100000</maxCacheElems>
    <cacheTimeout>1800</cacheTimeout>
    <maxConnections>1</maxConnections>
    <maxSessionTime>24</maxSessionTime>
    <failoverThreshold>1</failoverThreshold>
    <aaaTimeoutThreshold>-1</aaaTimeoutThreshold>
  - <logOutUrls>
      <url>/oamsso/logout.html</url>
    </logOutUrls>
  - <userDefinedParameters>
    - <userDefinedParam>
        <name>MaxPostDataLength</name>
        <value>750000</value>
      </userDefinedParam>
    </userDefinedParameters>
    <sleepFor>60</sleepFor>
    <cachePragmaHeader>no-cache</cachePragmaHeader>
    <cacheControlHeader>no-cache</cacheControlHeader>
    <debug>false</debug>
    <ipValidation>0</ipValidation>
    <denyOnNotProtected>1</denyOnNotProtected>
    <allowManagementOperations>false</allowManagementOperations>
  </OAMUpdateAgentRegRequest>
```

10.8.2 About Remote Application Domain Management Modes

Oracle Access Manager 11g provides two modes to manage Application Domains without registering or modifying an Agent.

> **Note:** Remote Application Domain management supports only create and update functions. Remove application domain management does not support remote removal of any application domains. Application Domains removal is a manual task that must be performed using the Oracle Access Manager Console.

Table 10–9 describes these remote Application Domain management modes. Again, command parameters include the mode, and an input *Request.xml file using a relative path with respect to OAM_REG_HOME, the preferred location for input files):

```
./oamreg.sh <mode> <input_file> [prompt_flag] [component.oam.config_file] <mode>
value
```

Table 10–9 Remote Application Domain Management Modes

Mode	Description
policyCreate *CreatePolicyRequest.xml*	Allows Administrators to create Host Identifiers and an Application Domain without registering an Agent. `./bin/oamreg.sh policyCreate input/CreatePolicyRequest.xml` Functions include: ■ Create an application domain ■ Create default protected, public, and excluded resource ■ Create host Port variations list ■ Create policies ■ Create resources with query string
policyUpdate *UpdatePolicyRequest.xml*	Allows Administrators to update existing Host Identifiers and Application Domain without updating an Agent. `./bin/oamreg.sh policyUpdate input/UpdatePolicyRequest.xml` Functions include: ■ Update an application domain ■ Update default protected, public, and excluded resource ■ Update host Port variations list ■ Update policies ■ Update resources with query string

Table 10–9 (Cont.) Remote Application Domain Management Modes

Mode	Description

[prompt_flag] value:
[-noprompt]
	Optional. When the -noprompt flag is used, oamreg can read input from system.in by using echo and pipe to pass data.
	Examples from OAM_REG_HOME location:
	`(echo username; echo password; echo webgate_password;) | ./bin/oamreg.sh inband input/Request.xml -noprompt component.oam.conf`
	`(echo username; echo password; echo webgate_password; echo httpscert_trust_prompt;) | ./bin/oamreg.sh inband input/Request.xml -noprompt`
	`(echo username; echo password; echo webgate_password; echo cert_password;) | ./bin/oamreg.sh inband input/Request.xml -noprompt`
	`(echo username; echo password; echo webgate_password; echo httpscert_trust_prompt; echo cert_password;) | ./bin/oamreg.sh inband input/Request.xml -noprompt`

For more information, see:

- About the Create Policy Request File
- About the Update Policy Request File
- About <rregApplicationDomain> Elements

10.8.2.1 About the Create Policy Request File

The following information can be created using `policyCreate` mode and the CreatePolicyRequest.xml without creating or updating an agent registration:

- Create a Host Identifier add multiple `hostPortVariations` (host port pairs).
- Create an Application Domain.
- Add multiple protected, public, and excluded resources. Resources can be with or without query strings, both are supported.
- Create default authentication and authorization policies for the resources that do not require customized policies.

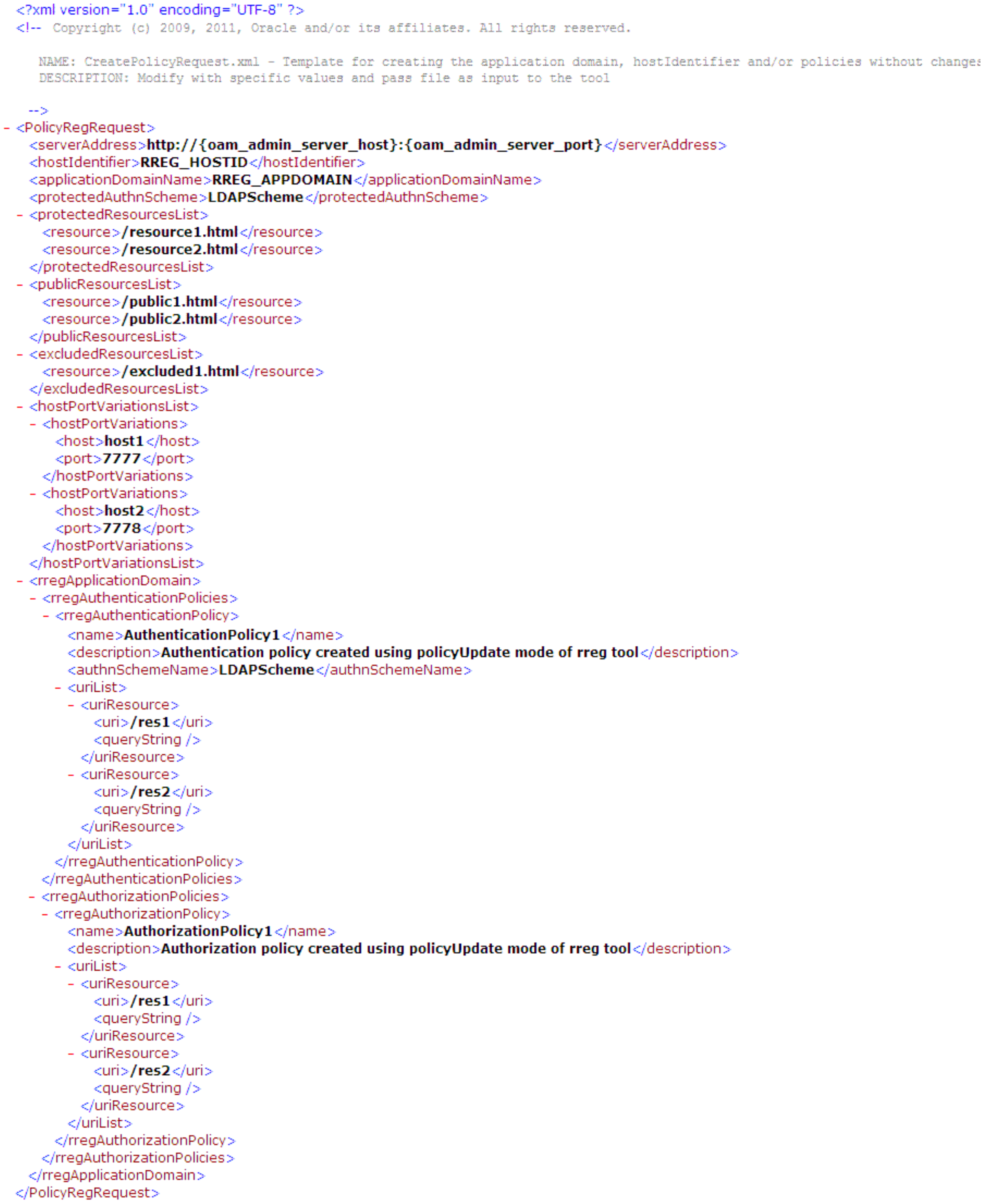
Some parameters in the CreatePolicyRequest.xml file are new and not included in the full agent registration XML files, while certain elements in the original agent registration file are used to create or update. However, some elements are The primary differences of CreatePolicyRequest.xml are specific to:

- Elements for Authentication and Authorization Policies and resources are provided
- No <agentName> element or related elements are provided

Figure 10–5 illustrates CreatePolicyRequest.xml.

> **See Also:** "About <rregApplicationDomain> Elements" on page 10-37

Figure 10–5 CreatePolicyRequest.xml



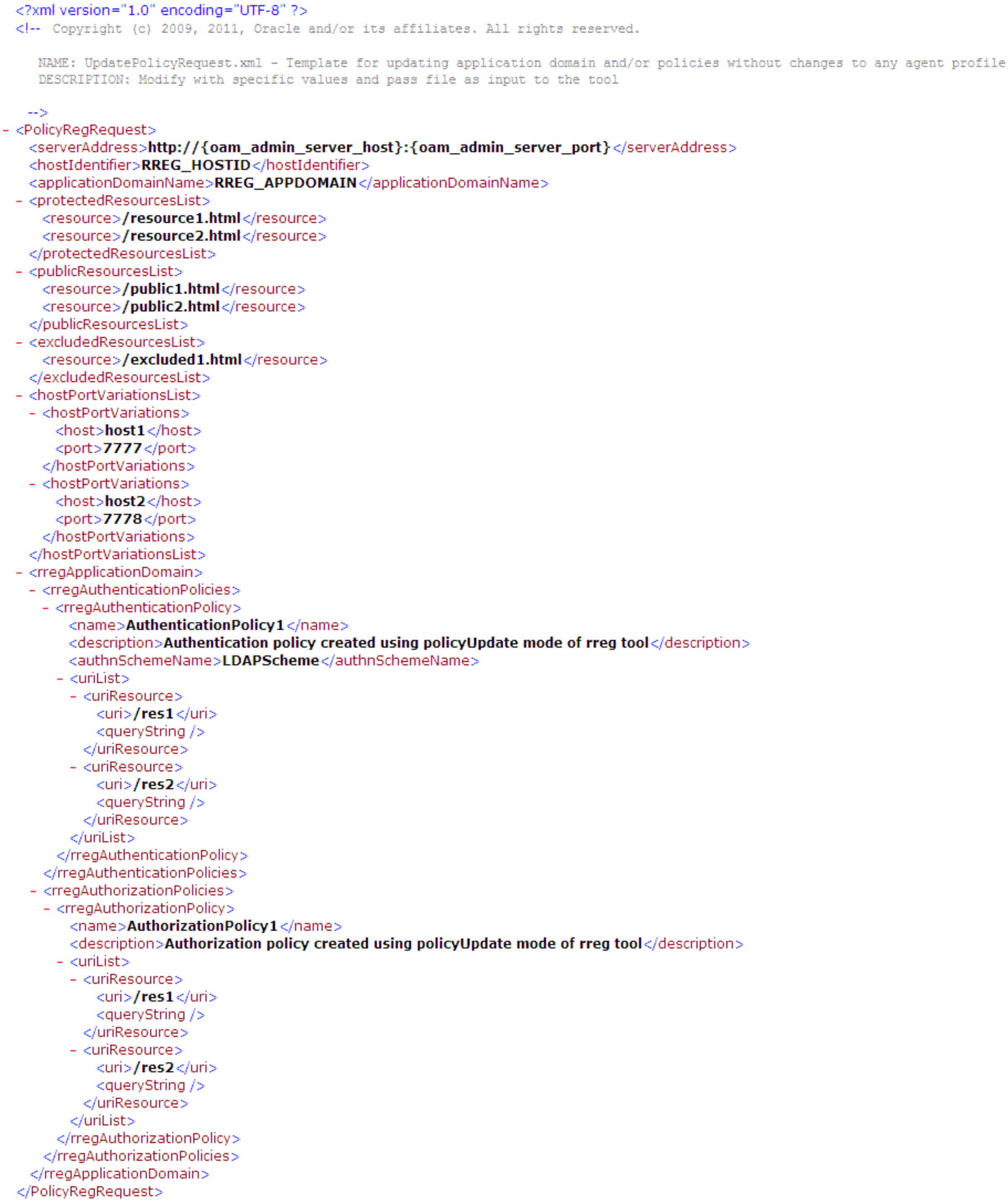
```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Copyright (c) 2009, 2011, Oracle and/or its affiliates. All rights reserved.

   NAME: CreatePolicyRequest.xml - Template for creating the application domain, hostIdentifier and/or policies without change
   DESCRIPTION: Modify with specific values and pass file as input to the tool

  -->
- <PolicyRegRequest>
    <serverAddress>http://{oam_admin_server_host}:{oam_admin_server_port}</serverAddress>
    <hostIdentifier>RREG_HOSTID</hostIdentifier>
    <applicationDomainName>RREG_APPDOMAIN</applicationDomainName>
    <protectedAuthnScheme>LDAPScheme</protectedAuthnScheme>
  - <protectedResourcesList>
      <resource>/resource1.html</resource>
      <resource>/resource2.html</resource>
    </protectedResourcesList>
  - <publicResourcesList>
      <resource>/public1.html</resource>
      <resource>/public2.html</resource>
    </publicResourcesList>
  - <excludedResourcesList>
      <resource>/excluded1.html</resource>
    </excludedResourcesList>
  - <hostPortVariationsList>
    - <hostPortVariations>
        <host>host1</host>
        <port>7777</port>
      </hostPortVariations>
    - <hostPortVariations>
        <host>host2</host>
        <port>7778</port>
      </hostPortVariations>
    </hostPortVariationsList>
  - <rregApplicationDomain>
    - <rregAuthenticationPolicies>
      - <rregAuthenticationPolicy>
          <name>AuthenticationPolicy1</name>
          <description>Authentication policy created using policyUpdate mode of rreg tool</description>
          <authnSchemeName>LDAPScheme</authnSchemeName>
        - <uriList>
          - <uriResource>
              <uri>/res1</uri>
              <queryString />
            </uriResource>
          - <uriResource>
              <uri>/res2</uri>
              <queryString />
            </uriResource>
          </uriList>
        </rregAuthenticationPolicy>
      </rregAuthenticationPolicies>
    - <rregAuthorizationPolicies>
      - <rregAuthorizationPolicy>
          <name>AuthorizationPolicy1</name>
          <description>Authorization policy created using policyUpdate mode of rreg tool</description>
        - <uriList>
          - <uriResource>
              <uri>/res1</uri>
              <queryString />
            </uriResource>
          - <uriResource>
              <uri>/res2</uri>
              <queryString />
            </uriResource>
          </uriList>
        </rregAuthorizationPolicy>
      </rregAuthorizationPolicies>
    </rregApplicationDomain>
  </PolicyRegRequest>
```

Many of the same parameters are found in the CreatePolicyRequest.xml file and the full agent registration XML files discussed earlier. The CreatePolicyRequest.xml file provides elements for Authentication and Authorization Policies and resources (and no <agentName> element).

10.8.2.2 About the Update Policy Request File

Figure 10–6 illustrates the UpdatePolicyRequest.xml, which is nearly identical to CreatePolicyRequest.xml.

Figure 10–6 UpdatePolicyRequest.xml



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Copyright (c) 2009, 2011, Oracle and/or its affiliates. All rights reserved.

    NAME: UpdatePolicyRequest.xml - Template for updating application domain and/or policies without changes to any agent profile
    DESCRIPTION: Modify with specific values and pass file as input to the tool

  -->
- <PolicyRegRequest>
    <serverAddress>http://{oam_admin_server_host}:{oam_admin_server_port}</serverAddress>
    <hostIdentifier>RREG_HOSTID</hostIdentifier>
    <applicationDomainName>RREG_APPDOMAIN</applicationDomainName>
  - <protectedResourcesList>
      <resource>/resource1.html</resource>
      <resource>/resource2.html</resource>
    </protectedResourcesList>
  - <publicResourcesList>
      <resource>/public1.html</resource>
      <resource>/public2.html</resource>
    </publicResourcesList>
  - <excludedResourcesList>
      <resource>/excluded1.html</resource>
    </excludedResourcesList>
  - <hostPortVariationsList>
    - <hostPortVariations>
        <host>host1</host>
        <port>7777</port>
      </hostPortVariations>
    - <hostPortVariations>
        <host>host2</host>
        <port>7778</port>
      </hostPortVariations>
    </hostPortVariationsList>
  - <rregApplicationDomain>
    - <rregAuthenticationPolicies>
      - <rregAuthenticationPolicy>
          <name>AuthenticationPolicy1</name>
          <description>Authentication policy created using policyUpdate mode of rreg tool</description>
          <authnSchemeName>LDAPScheme</authnSchemeName>
        - <uriList>
          - <uriResource>
              <uri>/res1</uri>
              <queryString />
            </uriResource>
          - <uriResource>
              <uri>/res2</uri>
              <queryString />
            </uriResource>
          </uriList>
        </rregAuthenticationPolicy>
      </rregAuthenticationPolicies>
    - <rregAuthorizationPolicies>
      - <rregAuthorizationPolicy>
          <name>AuthorizationPolicy1</name>
          <description>Authorization policy created using policyUpdate mode of rreg tool</description>
        - <uriList>
          - <uriResource>
              <uri>/res1</uri>
              <queryString />
            </uriResource>
          - <uriResource>
              <uri>/res2</uri>
              <queryString />
            </uriResource>
          </uriList>
        </rregAuthorizationPolicy>
      </rregAuthorizationPolicies>
    </rregApplicationDomain>
  </PolicyRegRequest>
```

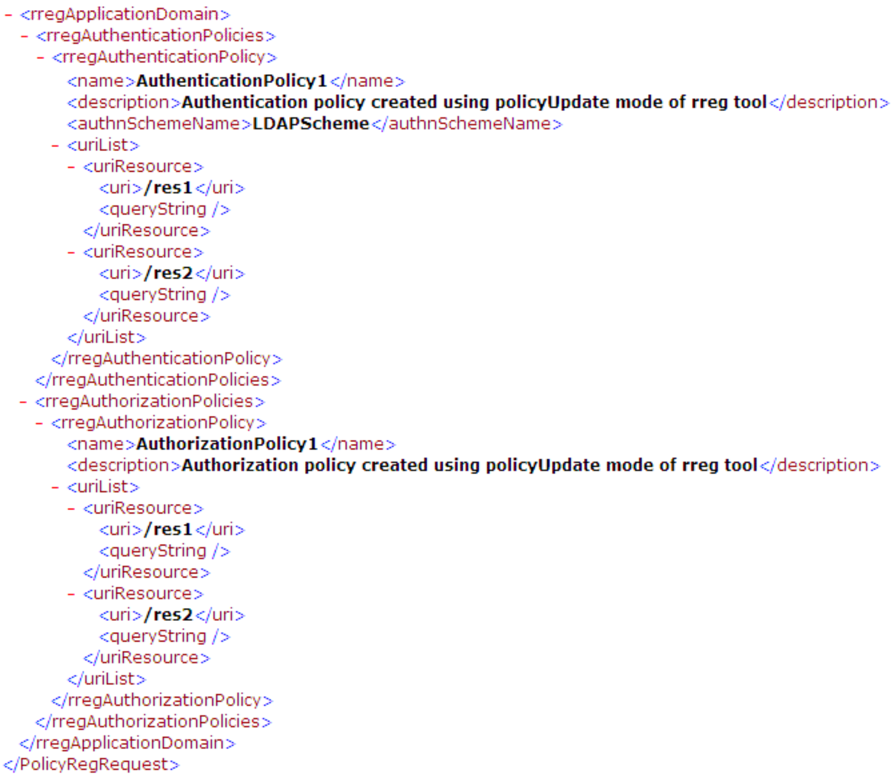
The UpdatePolicyRequest.xml provides the same elements as the CreatePolicyRequest.xml, with the exception of the `<protectedAuthnScheme>` element. Using UpdatePolicyRequest.xml, Administrators can:

- Update a Host Identifier add multiple `hostPortVariations` (host port pairs)
- Update an Application Domain
- Add multiple protected, public, and excluded resources.Resources can be with or without query strings, both are supported.
- Update default authentication and authorization policies for the resources that do not require customized policies
- Create customized policies that include:
  - Policy display name
  - Policy description
  - Authentication scheme (Authentication policies only)
  - A subset of resources to be associated with the policy

10.8.2.3 About <rregApplicationDomain> Elements

This section describes the unique remote management elements for Application Domains shown in Figure 10–7. These are found in the CreatePolicyRequest.xml and UpdatePolicyRequest.xml files and described in Table 10–10.

Figure 10–7 Remote Management Elements for Application Domains



```
- <rregApplicationDomain>
  - <rregAuthenticationPolicies>
    - <rregAuthenticationPolicy>
        <name>AuthenticationPolicy1</name>
        <description>Authentication policy created using policyUpdate mode of rreg tool</description>
        <authnSchemeName>LDAPScheme</authnSchemeName>
      - <uriList>
        - <uriResource>
            <uri>/res1</uri>
            <queryString />
          </uriResource>
        - <uriResource>
            <uri>/res2</uri>
            <queryString />
          </uriResource>
        </uriList>
      </rregAuthenticationPolicy>
    </rregAuthenticationPolicies>
  - <rregAuthorizationPolicies>
    - <rregAuthorizationPolicy>
        <name>AuthorizationPolicy1</name>
        <description>Authorization policy created using policyUpdate mode of rreg tool</description>
      - <uriList>
        - <uriResource>
            <uri>/res1</uri>
            <queryString />
          </uriResource>
        - <uriResource>
            <uri>/res2</uri>
            <queryString />
          </uriResource>
        </uriList>
      </rregAuthorizationPolicy>
    </rregAuthorizationPolicies>
  </rregApplicationDomain>
</PolicyRegRequest>
```

See Also: Table 10–6, " Elements Common to Full Remote Registration Requests" for a description of elements common to remote registration and remote management.

Table 10–10 <rregApplicationDomain> Remote Management Template Elements

Element	Description	Example
<rregAuthenticationPolicies> <rregAuthenticationPolicy>	Specifies the name and description for the Authentication Policy (to use when creating it anew or updating an existing policy).	<rregAuthenticationPolicies> <rregAuthenticationPolicy> <name>AuthenticationPolicy1</name> <description>Authentication policy created using policyUpdate mode of rreg tool</description> . . </rregAuthenticationPolicy> </rregAuthenticationPolicies>
<authnSchemeName>	Specifies the Authentication Scheme to use in the Authentication Policy.	<rregAuthenticationPolicies> . . authnSchemeName>LDAPScheme </authnSchemeName> . . </rregAuthenticationPolicy> </rregAuthenticationPolicies>
<uriList>	Identifies a resource that requires authentication using the Authentication Policy.	<rregAuthenticationPolicies> . . <uriList> - <uriResource> <uri>/res1</uri> <queryString /> </uriResource> </uriList> . . </rregAuthenticationPolicy> </rregAuthenticationPolicies>
<rregAuthorizationPolicies> <rregAuthorizationPolicy>	Specifies the name and description for the Authorization Policy (to use when creating it anew or updating an existing policy).	<rregAuthorizationPolicies> <rregAuthorizationPolicy> <name>AuthorizationPolicy1</name> <description>Authorization policy created using policyUpdate mode of rreg tool</description> . . </rregAuthorizationPolicy> </rregAuthorizationPolicies>
<uriList>	Identifies a resource that requires Authorization using the Authorization Policy.	<rregAuthorizationPolicies> . . <uriList> - <uriResource> <uri>/res1</uri> <queryString /> </uriResource> </uriList> . . </rregAuthorizationPolicy> </rregAuthorizationPolicies>

10.9 Managing Agents Remotely

This section provides the following topics:

- Performing Remote Agent Updates
- Performing Remote Agent Validation
- Performing Remote Agent Removal

10.9.1 Performing Remote Agent Updates

Prerequisites

Review About Remote Agent Management Modes

To remotely update an Agent registration

1. Set up the registration tool as described in, "Acquiring and Setting Up the Registration Tool" on page 10-22.
2. Create your update request using one of these templates:
   - OSSOUpdateAgentRequest.xmlOSSOUpdateAgentRequest.xml
   - OAM11GUpdateAgentRequest.xml
   - OAMUpdateAgentRequest.xml
3. On the computer hosting the Agent, run the following command with `agentUpdate` mode specify your own *Request*.xml as the input file. For example:

   ```
   ./bin/oamreg.sh agentUpdate input/*UpdateAgentRequest.xml
   ```

4. Provide the registration Administrator user name and password when asked.
5. Read the messages on-screen to confirm:
   - Success: On-screen message confirms

     ```
     agentUpdate 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*/
6. 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.
7. Finalize Agent Registration: Perform the following steps to finalize an OAM Agent registration:

   See Also:

   - Chapter 27, "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/Access Client: $*WG_install_dir*/oblix/lib/ObAccessClient.xml

11g Webgate/Access Client: $*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.

10.9.2 Performing Remote Agent Validation

Prerequisites

Review About Remote Agent Management Modes

To remotely validate an Agent registration

1. Set up the registration tool as described in, "Acquiring and Setting Up the Registration Tool" on page 10-22.
2. On the computer hosting the Agent, run the following command in `agentValidate` mode. For example:

   ```
   ./bin/oamreg.sh agentValidate agentname
   ```

3. Provide the registration Administrator user name and password when asked.
4. Read the messages on-screen to confirm:

   - **Success**: On-screen message confirms

     ```
     AgentValidation 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*/

10.9.3 Performing Remote Agent Removal

Prerequisites

Review About Remote Agent Management Modes

To remotely remove an Agent registration

1. Set up the registration tool as described in, "Acquiring and Setting Up the Registration Tool" on page 10-22.
2. On the computer hosting the Agent, run the following command in `agentDelete` mode. For example:

   ```
   ./bin/oamreg.sh agentDelete agentname
   ```

3. Provide the registration Administrator user name and password when asked.
4. Read the messages on-screen to confirm:

   - **Success**: On-screen message confirms

     ```
     AgentDelete process completed successfully!
     ```

10.10 Creating or Updating an Application Domain Without an Agent

Prerequisites

Review About Remote Application Domain Management Modes

To create or update an application domain without an Agent

1. Set up the registration tool as described in, "Acquiring and Setting Up the Registration Tool" on page 10-22.
2. Create your policy update request using one of these templates:
   - Create Policy Request File
   - Update Policy Request File
3. On the computer hosting the Agent, run the following command with `agentUpdate` mode specify your own *Request*.xml as the input file. For example:

   Create:

   ```
   ./bin/oamreg.sh policyCreate input/CreatePolicyRequest.xml
   ```

   Update:

   ```
   ./bin/oamreg.sh policyUpdate input/UpdatePolicyRequest.xml
   ```

4. Provide the registration Administrator user name and password when asked.
5. Read the messages on-screen to confirm:
   - **Success**: On-screen message confirms

     `agentUpdate 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*/

Part IV

Managing Oracle Access Manager SSO, Policies, and Testing

This part, Part III, provides information to help you understand single-sign on (SSO) with Oracle Access Manager, and help you to configure Oracle Access Manager policies and logout. Testing your single sign-on connection and policies is also described.

Part III contains the following chapters:

- Chapter 11, "Introduction to the OAM Policy Model, Single Sign-On"
- Chapter 12, "Managing Policy Components"
- Chapter 13, "Managing Policies to Protect Resources and Enable SSO"
- Chapter 14, "Validating Connectivity and Policies Using the Access Tester"
- Chapter 15, "Configuring Centralized Logout for OAM 11g"

11

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:

- Prerequisites
- Comparing the OAM 11g Policy Model and OAM 10g Model
- Introduction to the OAM 11g Policy Model
- Introduction to Configuring OAM Single Sign-On
- Introduction to SSO Components
- Introduction to OAM 11g Single Sign-On Implementation Types
- Introduction to OAM 11g SSO Processing

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 15, "Configuring Centralized Logout for OAM 11g".

11.1 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 9, "Registering Partners (Agents and Applications) by Using the Console"

11.2 Comparing the OAM 11g Policy Model and OAM 10g Model

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 11–1 compares the OAM 11g policy model with other models.

Table 11–1 Comparing OAM 11g Policy Model with OAM 10g

Policy Elements	OAM 11g Policy Model	OAM 10g Policy Model
Policy Authoring	Oracle Access Manager Console	OAM Policy Manager
Policy Store	Database	LDAP directory server
Domain	Application Domain	Policy Domain
Resources	1. No URL 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. Query-string 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. 7. Resources can designated as either Protected, Unprotected, or Excluded.	1. URL 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. 5. Non-HTTP resource types and operations can be defined.
Host identifiers	1. Host Identifiers are defined outside of policies and are used while defining HTTP resources. 2. Host Identifiers are mandatory for defining HTTP resources.	1. Host Identifiers are defined outside of policies and are used while defining HTTP resources. 2. Host Identifiers are not mandatory, for defining HTTP resources, till there are no Host Identifiers defined in the system.
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). 7. Token Issuance Policies can be defined using resources and user- or partner-based constraints. See "About Token Issuance Policies" on page 13-7.	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. 6. There is no support for Token Issuance Policies.

Table 11–1 (Cont.) Comparing OAM 11g Policy Model with OAM 10g

Policy Elements	OAM 11g Policy Model	OAM 10g Policy Model
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. Response values can be literal strings or can contain additional embedded expressions that derive values from request, use,r and session attributes.	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. 4. Response values can contain literal strings and list of user attribute values.
Authentication Schemes	Authentication Schemes are defined globally and can be referenced within authentication policies. The trust level is expressed as an integer value between 0 (no trust) and 99 (highest level of trust). Note: Level 0 is unprotected. Only unprotected resources can be added to an Authentication Policy that uses an authentication scheme at protection level 0. For more information, see Table 12–3, " Authentication Scheme Definition".	Authentication Schemes can be defined outside of policies and can be referenced within authentication policies.
Cookies	See: "About Single Sign-On Cookies During User Login" on page 11-13	See: "About Single Sign-On Cookies During User Login" on page 11-13
Query String-based HTTP Resource Definitions	Supported within Access Policies.	The Policy Model supports query string-based HTTP resource definitions within Access Policies. At run time, the OAM Proxy passes the Query String to the policy layer after URL encoding, just like for base resource URL. Only Query String that are part of HTTP GET requests are passed. Query String pattern does not apply to HTTP POST data. See: Table 13–1, " Resource Definition Elements"

11.3 Introduction to the OAM 11g Policy Model

This section introduces the Oracle Access Manager 11g policy model and the global shared components within it.

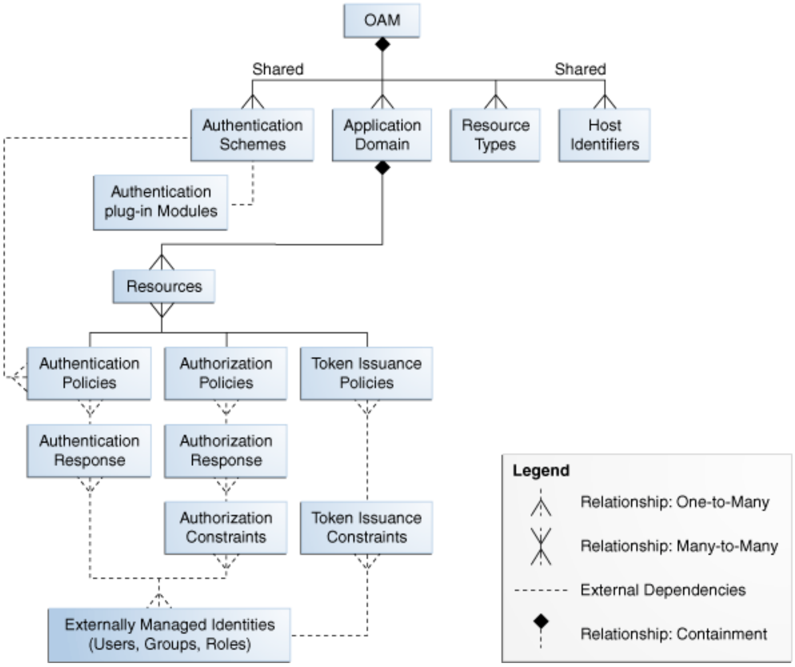
The Oracle Access Manager 11g policy model supports both authentication and authorization services within the context of an OAM application domain. The policy model relies on external user identity stores and on authentication modules, which are a part of the overall system configuration.

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 11–1 illustrates the different elements within the Oracle Access Manager 11g policy model.

See Also: "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31

Figure 11–1 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

See Also:

- "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31

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

11.3.1 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 12-2.

See Also:

- "Managing TokenServiceRP Type Resources" on page 19-34

11.3.2 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 11–2 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 11–2 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 12-5.

See Also: The following chapters for more information about registering agents and applications:

- Chapter 9, "Registering Partners (Agents and Applications) by Using the Console"
- Chapter 10, "Registering Partners (Agents and Applications) Remotely"

11.3.3 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

11.3.3.1 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 8-10.

For more information, see "Managing Authentication Schemes" on page 12-15.

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 12-15.

11.3.3.2 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 22, "Logging Component Event Messages"
> - Chapter 24, "Auditing Administrative and Run-time Events."

Several monitoring and diagnostic metrics are collected during authentication. For more information, see Chapter 25, "Monitoring Performance by Using Oracle Access Manager Console".

11.3.4 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.

11.3.5 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 13-11.

11.3.6 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 13-32.

> **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 13-3.

11.3.7 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 13-32.

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 13-39.

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 13-27.

11.4 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 and OAM 10g Model
   - 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 9, "Registering Partners (Agents and Applications) by Using the Console"
   - Chapter 10, "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 12-2.
5. Confirm that you have the proper authentication modules and schemes, as described in:

  - "Managing Authentication Modules" on page 8-10
  - "Managing Authentication Schemes" on page 12-15

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 12-5.
7. Add resource definitions and configure OAM 11g policies in the application domain as described in Chapter 13, "Managing Policies to Protect Resources and Enable SSO".

   See Also: The following topics, if needed:

   - "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31
   - "Managing TokenServiceRP Type Resources" on page 19-34

8. Configure the SSO Engine, as described in "Managing SSO Tokens and IP Validation" on page 8-4.
9. Validate the configuration, as described in "Validating Global Sign-On and Centralized Logout" on page 15-16.

11.5 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 During User Login
- About Single Sign-On Cookies

11.5.1 About Single Sign-On Components

This topic introduces key components to implementing and enforcing Oracle Access Manager 11g policies for single sign-on.

Table 11–3 summarizes key single sign-on components.

Table 11–3 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 11-19.	- OAM Server - Oracle Access Manager Console (installed on the WebLogic Administration Server) **Note**: Administrative users access the console home page by typing the URL: https://*host:port*/oamconsole. See Also: "Logging In to and Signing Out of Oracle Access Manager Console" on page 3-3.	- 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/Access Clients) - 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 11–3 (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 11-19

11.5.2 About Single Sign-On Cookies During User Login

Table 11–4 describes the cookies that can be set or cleared during user login.

Table 11–4 SSO Cookies

SSO Cookie Set at User Login	Set By	Description
OAM_ID cookie	OAM 11g 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	11g Webgate	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 11-14.

Table 11–4 (Cont.) SSO Cookies

SSO Cookie Set at User Login	Set By	Description
ObSSOCookie	10g Webgate	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 inter-operability between OAM 11g and older agents.
OAM_REQ	OAM 11g Server	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	11g Webgate	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*	Oracle HTTP Server	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 inter-operability between OAM 11g and older agents.
GITO cookie	OAM 11g Server	Provides backward compatibility and inter-operability 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 13, "Managing Policies to Protect Resources and Enable SSO".

11.5.3 About Single Sign-On Cookies

- OAMAuthnCookie for 11g OAM Webgates
- ObSSOCookie for 10g OAM Webgates
- OAM_REQ Cookie
- mod_osso Cookies

11.5.3.1 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 6-4.

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, described next.

11.5.3.2 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 6-4.

Cookie Expiration: Administrators can specify the desired Cookie Session Time in the OAM Agent registration. For more information, see "Registering and Managing OAM Agents Using the Console" on page 9-10.

11.5.3.3 OAM_REQ Cookie

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.

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.

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" on page F-7
- Table 11–4, " SSO Cookies"

11.5.3.4 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*.

11.6 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

11.6.1 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 13-32.

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
```

11.6.2 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 11-19

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 Oracle Access Manager 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.

11.6.3 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.

11.7 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

11.7.1 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

11.7.1.1 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.

11.7.1.2 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.

11.7.1.3 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".

11.7.2 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.

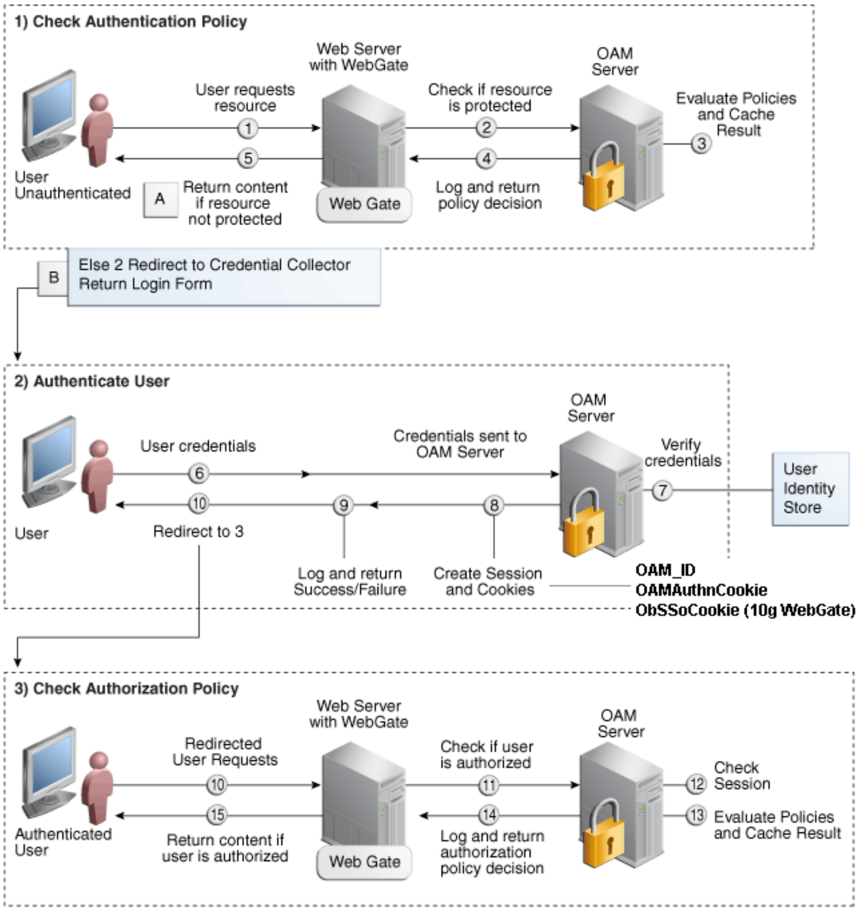
After authenticating the user and setting up the user context and token, OAM sets the SSO cookie and encrypts the cookie with the SSO Server key (which can be decrypted only by the SSO Engine).

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 11–2 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/Access Client 10g or Webgate 11g). There are slight variations with 11g Webgates.

Figure 11–2 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
2
3
5
4
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
User Identity Store
6
10
9
8
7
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
11
12
15
14
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.

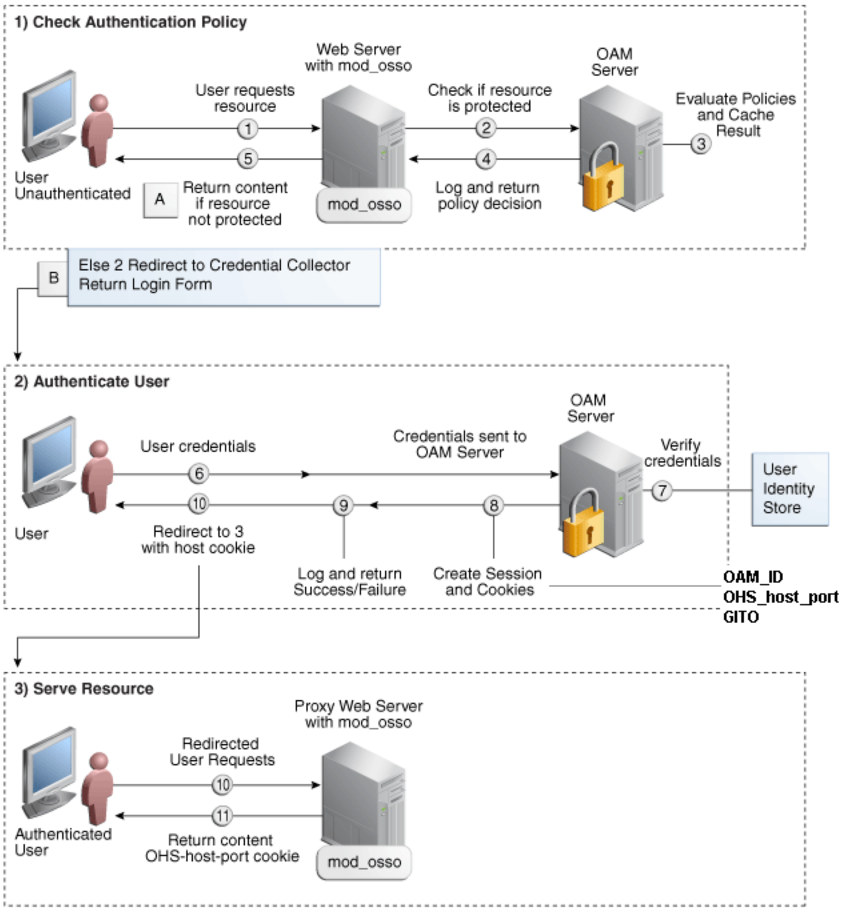
11.7.3 About SSO Login Log In Processing with OSSO Agents (mod_osso)

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 11–3 illustrates the login processing with mod_osso and OAM 11g.

Figure 11–3 SSO Login Processing with OSSO Agents


1) Check Authentication Policy
Web Server with mod_osso
OAM Server
User requests resource
1
5
Check if resource is protected
2
4
Evaluate Policies and Cache Result
3
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
6
10
Credentials sent to OAM Server
Verify credentials
7
User Identity Store
9
8
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.

11.7.4 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.

12 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 Schemes
- Configuring Challenge Parameters for Encrypted Cookies

12.1 Prerequisites

The Oracle Access Manager 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 11-3
- Review a comparison of the current policy model versus other models in "Comparing the OAM 11g Policy Model and OAM 10g Model" on page 11-1
- Learn more about the Oracle Access Manager Console and controls from Chapter 3, "Getting Started with Common Administration and Navigation"

12.2 Introduction to Managing Policy Components

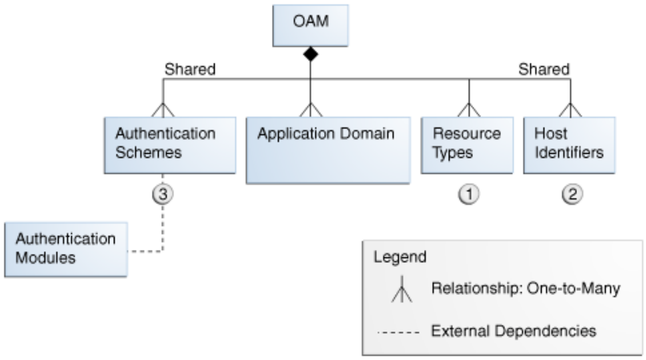
This section introduces the Oracle Access Manager 11g policy model and the global components within it.

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 12–1 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 12–1 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 13, "Managing Policies to Protect Resources and Enable SSO".

12.3 Managing Resource Types

This section includes the following topics:

- About Resource Types and Their Use
- About the Resource Type Page
- Searching for a Specific Resource Type

12.3.1 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.

With Oracle Access Manager 11g (11.1.1.5) command buttons to create/edit/delete a resource type are disabled. Oracle provided resource types include the following. For HTTP type resources, include a host identifier. For non-HTTP resource types, use the type name:

- HTTP
- TokenServiceRP
- wl_authen

HTTP: The default resource type 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.

wl_authen: Resources for representing WebLogic Authentication schemes. A non-HTTP resource type, wl_authen, is available to use with resources deployed in a WebLogic container. Resources of type wl_authen, require a custom Access Client. The protected resource is accessed using its URL on the Oracle WebLogic Server.

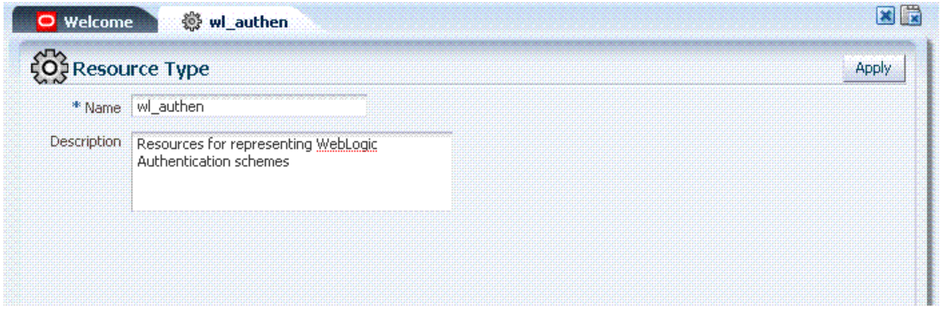
TokenServiceRP: Resources for representing Token Service Relying Party.

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.

12.3.2 About the Resource Type Page

In the Oracle Access Manager Console, resource types are organized with other Components under the Policy Configuration tab, as shown in Figure 12–2. 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 12–2 Default wl_authen Resource Type Definition



* Name: wl_authen

Description: Resources for representing WebLogic Authentication schemes

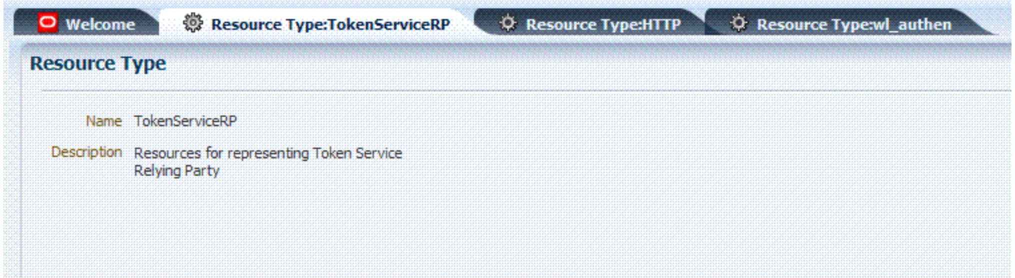
The `HTTP` resource type is used for Web applications protected by Oracle Access Manager 11g.

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

The `TokenServiceRP` resource type is used to represent the Token Service Relying Party, as shown in Figure 12–3. For more information, see "Managing TokenServiceRP Type Resources" on page 19-34.

Figure 12–3 TokenServiceRP



Name TokenServiceRP

Description Resources for representing Token Service Relying Party

Table 12–1 describes the elements in each resource type definition.

Table 12–1 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.

12.3.3 Searching for a Specific Resource Type

Users with valid Administrator credentials can use the following procedure to locate a non-HTTP resource type.

See Also: "Conducting Policy Element Searches Using the Console"

To search for a resource type

1. Activate the Policy Configuration tab, then click the Search tab.
2. From the search type list, choose Resource Type, enter the name of the Resource Type you want to find (with or without a wild card (*)), and click Search. For example:

   ```
   h*
   ```

   Alternatively: Go to the desired Application Domain, open the Resources node to display controls for that domain, choose a Resource Type from the list, and click Search.
3. Click the Search Results tab to display the results table, and then:
   - **Edit or View:** 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.
   - **Reconfigure Table**: Select a View menu item to alter the appearance of the results table.

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

12.4 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

12.4.1 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 Oracle Access Manager 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

12.4.1.1 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 12-7), 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.

For instance, a user accessing aseng-wiki, would enter:

```
http://corp-wiki.uk.mycompany.com/mywikipage
```

Here, "mywikipage" is the resource URL and "corp-wiki.uk.mycompany.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 12-7

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.

12.4.1.2 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".

12.4.1.3 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

12.4.2 About Virtual Web Hosting

You can install a Webgate on a Web server that contains multiple Web site and domain names. The Webgate must reside in a location that enables it to protect all of the Web sites on that server.

> **Note:** The information here is the same for both 11g and 10g Webgates.

The virtual Web hosting feature of many Web servers enables you to support multiple domain names and IP addresses that each resolve to their unique subdirectories on a single virtual server. For example, you can host abc.com and def.com on the same virtual server, each with its own domain name and unique site content. You can have name-based or IP-based virtual hosting.

A virtual host referees 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 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.

12.4.2.1 Placing a Webgate Behind a Reverse Proxy

You can use OAM 10g Webgates with reverse proxies for OAM 11g. This topic discusses benefits and pitfalls of this strategy.

Benefits:

- All Web content can be protected from a single logical component as long as all requests go through the proxy.

  This is true even for platforms that are not supported by Oracle Access Manager. If you have different types of Web servers (for example, iPlanet, Apache, and so on) on different platforms (for example, Windows XP, Linux, and so on), all content on these servers can be protected. A reverse proxy can be a workaround for unsupported Web servers, eliminating the need to write custom Access Clients for unsupported Web servers and on platforms that do not have Webgate support, for example, MacOS.

- A reverse proxy offers architecture flexibility.

  Reverse proxies can allow deployments to expose an application that is available on the intranet to the extranet. Or applications that are available on the extranet can be exposed to the intranet. This can be done without any changes to the application that is already deployed.

- You only need to install a separate Webgate on the reverse proxy, rather than on every Web server.

  This allows for a single management point and can help with manageability of the system. You can manage the security of all of the Web servers through the reverse proxy without establishing a footprint on the other Web Servers.

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

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

  This will also require that Webgates deployed on this Web server be configured to not enforce IP validation for requests from the reverse proxy server that front-ends the Webgate. This is done by configuring the known IP addresses of the reverse proxy server or servers in the IP Validation list. Note that while you can achieve the same effect by turning IP validation off for the Webgate, this is not a recommended approach due to security risks.

Ensuring that the Web server only accepts requests from reverse proxies is typically done by adding an ACL statement in the server. This prevents users from bypassing the reverse proxy and directly accessing restricted content.

- Update the virtual hosts that are configured in the Policy Manager so that the Access System intercepts requests that are sent to the reverse proxy.
- Prevent people from circumventing the proxy by entering URLs that point directly to the back-end system.

  You can prevent this problem through the use of Web Server Access Control Lists or firewall filters.
- Since all user requests are processed by the proxy, you must deploy enough proxy servers to enable the system to handle the load.
- Redirect all existing URLs to the host name and port number of the reverse proxy server.

  This often requires configuring the reverse proxy to perform content inspection and rewriting to prevent any absolute HTML links, for instance, to prevent broken link. This is achievable with most reverse proxies, and this is something you can configure independently of the Access System,.
- It is a best practice that URL links exposed to the front-ended applications rely on only relative URLs (../../*sub-path*/*resource*) rather than absolute URLs (http://*hostname.domain*:[*port*]/*path*/*resource*).

  Absolute URLs can break links on the end user's browser when deployed behind a reverse proxy.

12.4.2.2 Configuring Virtual Hosting for Non-Apache Web Servers

Ensure that the Virtual Host box is checked on the 10g Webgate registration page.

On most Web servers, other than Apache-based servers, you must set the Preferred Host value to HOST_HTTP_HEADER. This ensures that, when user's browser sends a request, the Webgate sets the value of the Preferred 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.

In the Preferred Host field of the expanded 10g Webgate registration page, enter the following:

HOST_HTTP_HEADER.

IIS Virtual Hosting: From the IIS console, you must configure each virtual Web site to contain the following fields:

- Host Header Name
- IP address
- Port

See Also:

- `http://www.simpledns.com/kb.aspx?kbid=1149`
- `http://support.microsoft.com/kb/q190008/`

12.4.2.3 Associating a Webgate for Apache with Virtual Hosts, Directories, or Files

Ensure that the Virtual Host box is checked on the 10g Webgate registration page.

On Apache-based Web servers (Apache, Apache 2, IBM HTTP Server, Oracle HTTP Server, and so on), the Preferred 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 must 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.

When using an Apache-based reverse proxy for single sign-on, in the Web server configuration file (httpd.config, for example) file you specify the Web sites to run on the Apache server. The settings can be global across all Web sites or local to a Web site. You can restrict the Oracle Access Manager loading references in the httpd.config file to be associated with a specified site, with virtual hosts, specific directories or even files.

To associate the Webgate with specific targets, you move the following directives the the http.conf file:

```
AuthType Oblix
require valid-user
```

You can put these directives in a block that tells Apache to use Webgate for every request. You can also move the directives to a block that limits when the Webgate is called. The following is an example of putting the LocationMatch directive after a VirtualHost directive:

```
  DocumentRoot /usr/local/apache/htdocs/myserver
  ServerName myserver.example.net
  AuthType Oblix
  require valid-user
```

After you move the LocationMatch block to the VirtualHost directive, the Webgate will only work for that virtual host. You can add the LocationMatch block to as many virtual hosts as you want. The following examples shows how you could protect one virtual server:

```
ServerAdmin webmaster@example.net
DocumentRoot "Z:/Apps/Apache/htdocs/MYsrv"
ServerName apps.example.com
    ProxyRequests On
    SSLEngine on
    SSLCACertificateFile Z:/Apps/sslcert_myapps_ptcweb32/intermediateca.cer
    SSLCertificateFile Z:/Apps/sslcert_myapps_ptcweb32/sslcert_myapps_ptcweb32.cer
    SSLCertificateKeyFile Z:/Apps/sslcert_myapps_ptcweb32/sslcert_myapps_
ptcweb32.key
    ErrorLog logs/proxysite1_log
    CustomLog logs/proxysite1_log common
    ProxyPass /https://apps.example.com/
    ProxyPassReverse /https://apps.example.com/
    ProxyPass /bkcentral https://apps.example.com/bkcentral
    ProxyPassReverse /bkcentral https://apps.example.com/bkcentral
    ProxyPass /NR https://apps.example.com/NR
    ProxyPassReverse /NR https://apps.example.com/NR

      AuthType Oblix
      require valid-user

#*** BEGIN Oracle Access Manager Webgate Specific ****

LoadModule obWebgateModule
Z:/apps/Oracle/WebComponent/access/oblix/apps/webgate/bin/webgate.dll
WebgateInstalldir Z:/apps/Oracle/WebComponent/access
WebgateMode PEER

 SetHandler obwebgateerr

SSLMutex sem
SSLRandomSeed startup builtin
SSLSessionCache none

SSLLog logs/SSL.log
SSLLogLevel info
# You can later change "info" to "warn" if everything is OK
```

12.4.3 About the Host Identifier Page

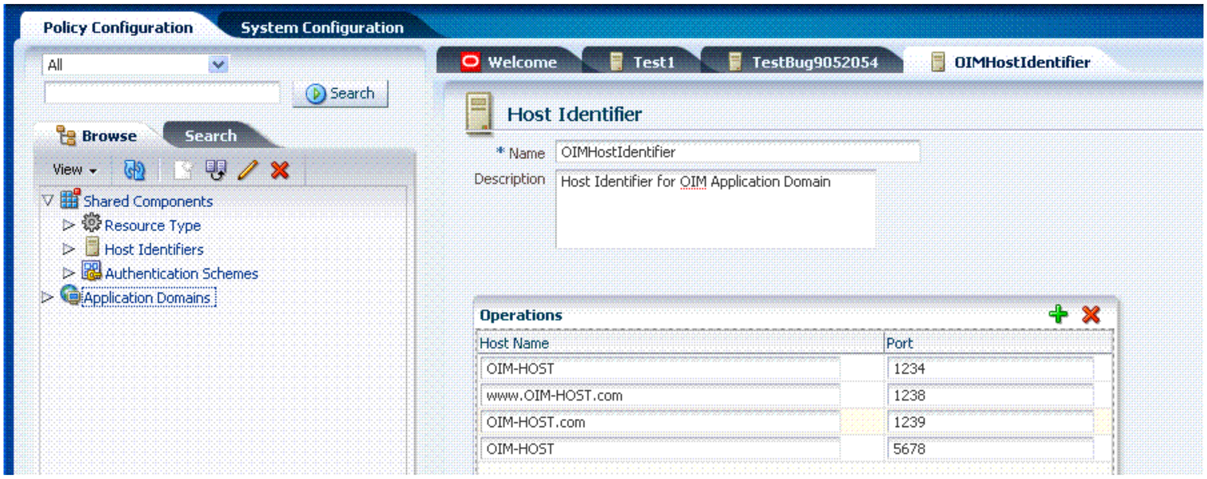
A host identifier is automatically created when an Agent (and application) are registered using either the Oracle Access Manager Console or the remote registration tool. In the application domain that is registered with the Agent, the host identifier is used automatically.

Administrators can use the console to create and manage host identifiers. Within the Oracle Access Manager 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 12–4 illustrates a typical Host Identifier configuration page in the 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 12–4 Host Identifier Page



Host Identifier

* 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 12–2 describes the host identifier definition.

Table 12–2 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 12-7 and "Host Identifier Guidelines" on page 12-6. ■ Port: The Web server port used by each host or permutation

12.4.4 Creating a Host Identifier

Users with valid Administrator credentials can use the following procedure to create a host identifier definition manually. This is needed if an application and resources were manually added to a host that has no mapped host identifier. When you choose Auto Create Policies when registering an Agent, this is done automatically.

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 12-5
- "About Virtual Web Hosting" on page 12-7

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.

12.4.5 Searching for a Host Identifier Definition

Users with valid Administrator credentials can perform the following task to search for a specific host identifier.

See Also: "About Policy Configuration Search Controls"

To search for a host identifier

1. Activate the Policy Configuration tab, and then click the Search tab.
2. From the search type list, choose Host Identifiers to define your search.
3. In the text field, enter your search criteria (with or without a wild card (*)). For example:

   ```
   my_h*
   ```

4. Click the Search button to initiate the search.
5. Click the Search Results tab to display the results table, and then:
   - **Edit or View:** Click the Edit button in the tool bar (or double-click the name in the results table) to display the configuration page.
   - **Delete:** Click the Delete button in the tool bar to remove the selected item in the results table; confirm removal in the Confirmation window.
   - **Detach**: Click Detach in the tool bar to expand the table to a full page.
   - **Reconfigure Table**: 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.

12.4.6 Viewing or Editing a Host Identifier Definition

Users with valid 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 12-11

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 12–2):
   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.

12.4.7 Deleting a Host Identifier Definition

Users with valid 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 12-13 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.

12.5 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

12.5.1 About the Authentication Schemes Page

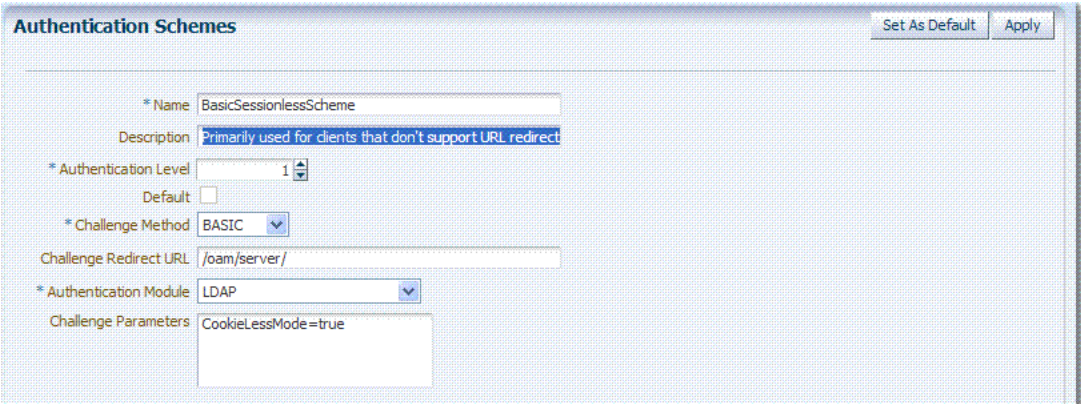
Access to a resource or group of resources can be governed by a single authentication process known as an authentication scheme. An authentication scheme is a named component that defines the challenge mechanism required to authenticate a user. Each authentication scheme must also include a defined authentication module (standard or custom, as described in "Managing Authentication Modules" on page 8-10).

When you register a partner (either using the 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 *KerberosScheme*, the copy is named *copyofKerberosScheme*.

All authentication schemes include the same elements with differing values. Figure 12–5 shows the default LDAPScheme page as an example. The Authentication Schemes navigation tree lists other default schemes that are delivered.

Figure 12–5 Default LDAPScheme Page



* Name: BasicSessionlessScheme

Description: Primarily used for clients that don't support URL redirect

* Authentication Level: 1

Default: □

* Challenge Method: BASIC

Challenge Redirect URL: /oam/server/

* Authentication Module: LDAP

Challenge Parameters: CookieLessMode=true

Table 12–3 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 12–3 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 12-18
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: Level 0 is unprotected. Only unprotected resources can be added to an Authentication Policy that uses an authentication scheme at protection level 0. For more information, see Table 13–1, " Resource Definition Elements". **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 12-24.
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 12-20
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 12-5.

Table 12–3 (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) ■ LDAPNoPasswordAuthModule ■ Kerberos Plugin (under Authentication Modules, Custom Authetication Module) ■ LDAP Plugin (under Authentication Modules, Custom Authetication Module ■ X509 Plugin (under the X509 Authentication Modules node) See also "Managing Authentication Modules" on page 8-10.
Challenge Parameters	Supported challenge parameters are discussed in "About Challenge Parameters for Authentication Schemes" on page 12-23
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 12–3. 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

12.5.1.1 Pre-configured Authentication Schemes

Table 12–4 identifies the pre-configured authentication schemes available with Oracle Access Manager 11g and some specific details of each. For more information about challenge parameters, see Table 12–5.

Table 12–4 Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
AnonymousScheme	Authentication Level: 0 Challenge Method: None Authentication Module: AnonymousModule	Leaves unprotected specific Oracle Access Manager URLs and allows users to access such 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. **Also**: When a resource is protected by AnonymousScheme, it is not displayed in a session search.
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.
BasicSessionlessScheme	Authentication Level: 1 Challenge Method: Basic Authentication Module: LDAP	Primarily used for clients that don't support URL redirect or cookies. Challenge Parameters: CookieLessMode=true **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.

Table 12–4 (Cont.) Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
FAAuthScheme	Authentication Level: 2 Challenge Method: FORM Authentication Module: LDAP Context: customWar Context Value: /fusion_apps	Protects Fusion Applications.
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 Context Type: default Context Value: /oam **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 Context Type: external	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
	Context Type: default
	Context Value: /oam
		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.
		Challenge Parameters:
		oaamPostAuth=true
		oaamPreAuth=true

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.
OAMAdminConsoleScheme	Authentication Level: 2 Challenge Method: FORM Authentication Module: LDAP Context Type: default Context Value: /oam	Authentication scheme for OAM Administration Console.
OIFScheme	Authentication Level: 2 Challenge Method: DAP Authentication Module: DAP Context Type: External	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). Challenge Parameters: TAPPartnerId=OIFDAPPartner

Table 12–4 (Cont.) Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
OIMScheme	Authentication Level: 1 Challenge Method: Form Authentication Module: LDAP Context Type: default Context Value: /oam	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.
TAPScheme	Authentication Level: 2 Challenge Method: DAP Authentication Module: DAP Context Type: External	To use TAPScheme for IDM product resources in the IAM Suite application domain, Protected HigherLevel Policy, the following configuration must be done in addition to changing the Authentication Scheme. 1. From the IAM Suite application domain, Protected HigherLevel Policy, remove IAMSuiteAgent:/oamTAPAuthenticate. 2. Create a new Authentication Policy in the IAM Suite application domain, that uses LDAPScheme. 3. Protect IAMSuiteAgent:/oamTAPAuthenticate using the newly created policy. Challenge Parameters: TAPPartnerId=TAPPartnerName
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.

12.5.1.2 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 keystore 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 Managerfor 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 12–3). 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 Oracle

Access Manager Console. This integration replaces OSSO 10g with OAM11g, with no changes from the Oracle Identity Federation side

The DAP challenge mechanism delegates authentication to a third party (Oracle Identity Federation in this case). The challenge_url points to the Oracle Identity Federation Server URL. When a resource is protected by this scheme, the OAM Server redirects to the Oracle Identity Federation Server URL for credential collection. OAM Server does not perform the credential collection or validation in this case. Oracle Identity Federation collects the credentials, authenticates the user against its identity store and returns an assertion token to the OAM Server consisting of the username. Oracle Access Manager receives and decrypts this token, checks whether the user is a valid user in the default 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 Oracle Identity Federation 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.

12.5.1.3 About Challenge Parameters for Authentication Schemes

Challenge parameters are short text strings consumed and interpreted by Webgates and Credential Collector modules to operate in the manner indicated by those values.

Here is an example of the default challenge parameter, which is interpreted by Webgate to use HttpOnly property for the cookies that it uses:

```
ssoCookie=httponly
```

Table 12–5 identifies the pre-configured schemes with challenge parameters. For more information on the schemes themselves, see Table 12–4.

Table 12–5 Challenge Parameters in Pre-configured Schemes

Pre-configured Schemes	Challenge Parameter
BasicSessionlessScheme	CookieLessMode=true Primarily used for clients that don't support URL redirect or cookies.

OAAMBasic	oaamPostAuth=true
	oaamPreAuth=true
	Protects OAAM-related resources. These parameters should be used when basic integration with OAAM is required.

OIFScheme	TAPPartnerId=OIFDAPPartner This scheme delegates authentication to Oracle Identity Federation, after which, Federation sends back a token that is asserted by the OAM Server.
TapScheme	TAPPartnerId=TAPPartnerName

12.5.1.4 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: 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.
- LDAPNoPasswordAuthModule
- Custom Authentication Modules: This type of module relies on custom plug-ins developed using the Oracle Access Manager Authentication Extensibility Java API. This type of module can use one or more custom plug-ins. With multiple plug-ins you can orchestrate each one to perform an authentication function and, depending on success or failure, call another authentication plug-in.

> **See Also:**
>
> - "Creating Custom Authentication Modules" on page 8-15
> - Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service for details about developing and deploying plug-ins, custom authentication modules, and schemes that use custom modules.

12.5.1.5 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)

12.5.1.5.1 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 12-26
- 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 12-26

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.

12.5.1.5.2 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.

12.5.1.5.3 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.

12.5.2 Creating an Authentication Scheme

Users with valid 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 12-15

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 (Table 12–3):
   a. Name
   b. Description
   c. Authentication Level
   d. Challenge Method
   e. Challenge Redirect URL
   f. Authentication Module
   g. Challenge Parameters
   h. Challenge URL
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

12.5.3 Searching for a Authentication Scheme

Users with valid Administrator credentials can perform the following task to search for a specific authentication scheme.

See Also: "About Policy Configuration Search Controls" on page 3-18

To search for an authentication scheme

1. Activate the Policy Configuration tab, and then click the Search tab.
2. From the search type list, choose Authentication Schemes to define your search.
3. In the text field, enter the desired scheme name (with or without wild card *). For example:

   ```
   OA*
   ```

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.

12.5.4 Viewing or Editing a Authentication Scheme

Users with valid Administrator credentials can use the following procedure to view or modify an existing authentication scheme.

See Also: "About the Authentication Schemes Page" on page 12-15

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 view or change.
3. Edit:
   a. On the Authentication Scheme page, modify values as needed (Table 12–3)
   b. Click Apply to submit the changes (or close the page without applying changes).
   c. Dismiss the Confirmation window.
4. Set as Default: Click the Set as Default button to automatically use this with new application domains, then close the Confirmation window.
5. Close the page when you finish.

12.5.5 Deleting an Authentication Scheme

Users with valid 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 12-15

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.

12.6 Configuring Challenge Parameters for Encrypted Cookies

This section provides the following topics:

- About ssoCookie Challenge Parameters for Encrypted Cookies
- Configuring Challenge Parameters for Encrypted Cookie Security
- Setting Challenge Parameters for Encrypted Cookie Persistence

12.6.1 About ssoCookie Challenge Parameters for Encrypted Cookies

In addition to the OAM Server cookie (OAM_ID), Oracle Access Manager implements single sign-on through an encrypted 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.

Oracle Access Manager provides the `ssoCookie` challenge parameter that you can use within any authentication scheme to control how Webgates set the flags of the encrypted cookie. For example:

- Securing Encrypted Cookie: Ensures that the encrypted cookie is sent only over an SSL connection and prevents the encrypted cookie from being sent back to a non-secure Web server.
- Persisting Encrypted Cookie: Allows the user to log in for a time period rather than a single session. Persistent cookie functionality works with Internet Explorer and Mozilla browsers.

Syntax between the parameter and values differs slightly depending upon your Webgate releases, as follows:

11g Webgate `ssoCookie=`

10g Webgate `ssoCookie:`

Multiple values must be separated by a semicolon (;). For example:

11g Webgate `ssoCookie=<value1>;<value2>;...`

10g Webgate `ssoCookie:<value1>;<value2>>;...`

The following example specifies sending the encrypted cookie over only an SSL connection and allows access to the encrypted cookie through client side scripts:

11g Webgate `ssoCookie=Secure;disablehttponly`

10g Webgate `ssoCookie:Secure;disablehttponly`

> **Note:** The value of the challenge parameter is case-sensitive. Be sure to enter an uppercase C in ssoCookie, and uppercase S in Secure.

Table 12–5 describes specific challenge parameters that control how Webgates set encrypted cookie flags for single sign-on.

Table 12–6 Challenge Parameters for Encrypted Cookies

Syntax for 11g Webgate and OAMAuthnCookie	Syntax for 10g Webgate and ObSSOCookie	Description
`ssoCookie=`	`ssoCookie:`	Parameter that controls flags for encrypted cookies.
`ssoCookie=httponly`	`ssoCookie:httponly`	Ensures that the encrypted cookie is not accessible to client side scripts such as JavaScript. Default: Enabled
`ssoCookie=disablehttponly`	`ssoCookie:disablehttponly`	Explicitly disables `httponly` functionality, making the encrypted cookie accessible to client side scripts. Once explicitly disabled, you must use the default value (`httponly`) to enable it.
`ssoCookie=Secure`	`ssoCookie:Secure`	Ensures that the encrypted cookie is sent only when the resource is accessed through HTTPS. A secure cookie is required only when a browser is visiting a server using HTTPS..
`ssoCookie=max-age=`*`time-in-seconds`*	`ssoCookie:max-age=`*`time-in-seconds`*	Creates a persistent cookie in Internet Explorer and Mozilla browsers, rather than one that lasts for a single session. Specifies the time interval *in-seconds* when the cookie expires. For example, to set the cookie to expire in 30 days (2592000 seconds): `max-age=2592000`

12.6.2 Configuring Challenge Parameters for Encrypted Cookie Security

The challenge parameter is case-sensitive. Be sure to enter an uppercase C in ssoCookie, and uppercase S in Secure.

> **See Also:** "About ssoCookie Challenge Parameters for Encrypted Cookies"

To secure the encrypted cookie

1. Create an authentication scheme, as described in "Creating an Authentication Scheme" on page 12-26.
2. In the Challenge Parameter field, specify the following to secure the encrypted cookie:

   11g Webgate `ssoCookie=Secure`

   10g Webgate `ssoCookie:Secure`
3. Confirm that the OAM Servers and clients (OAM Agents) are communicate securely across the Access Protocol channel, as described in Appendix E.

12.6.3 Setting Challenge Parameters for Encrypted Cookie Persistence

The challenge parameter is case-sensitive. Be sure to enter an uppercase C in ssoCookie.

> **See Also:** "About ssoCookie Challenge Parameters for Encrypted Cookies"

To define encrypted cookie persistence

1. Define an authentication scheme, as described in "Creating an Authentication Scheme" on page 12-26.
2. In the challenge parameter for this scheme, add the following:

11g Webgate	`ssoCookie=max-age=`*time-in-seconds*
10g Webgate	`ssoCookie:max-age=`*time-in-seconds*

13

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 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 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
- Validating Authentication and Authorization in an Application Domain
- Example: Pre-seeded IAM Suite Application Domain and Policies

13.1 Prerequisites

Review Chapter 11 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 9.
- Shared components for use in any application domain should be defined, as described in Chapter 12.

13.2 Introduction to Application Domain Creation

This section provides the following topics:

- About Automatic Application Domain Creation
- About Manually Creating Application Domains

13.2.1 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: IAMSuiteAgent is a pre-registered Java Agent filter that provides an application domain (IAMSuite) to protect Oracle Fusion Middleware console and other consoles.

For more information, see "Anatomy of an Application Domain and Policies" on page 13-3.

13.2.2 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 13-3.
2. Start the application domain registration as described in "Creating a Fresh Application Domain Manually" on page 13-8.
3. Add one or more resource definitions to the domain, set as Protected, Unprotected, or Excluded, as described in "Adding and Managing Resource Definitions for Use in Policies" on page 13-11.
4. Configure one or more authentication policies in the domain, as described in "Defining Authentication Policies for Specific Resources" on page 13-22.
5. Configure one or more authorization policies to the domain, as described in "Defining Authorization Policies for Specific Resources" on page 13-27.
6. Define one or more SSO Responses to your policies, as described in "Adding and Managing Policy Responses for SSO" on page 13-32.
7. Define one or more authorization constraints as described in "Defining Authorization Policy Constraints" on page 13-46.
8. Define Token Issuance Policies, if needed, as described in "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31.
9. Configure SSO settings as described in "Managing SSO Tokens and IP Validation" on page 8-4.
10. Validate the domain's operation, as described in "Validating Authentication and Authorization in an Application Domain" on page 13-50.

13.3 Anatomy of an Application Domain and Policies

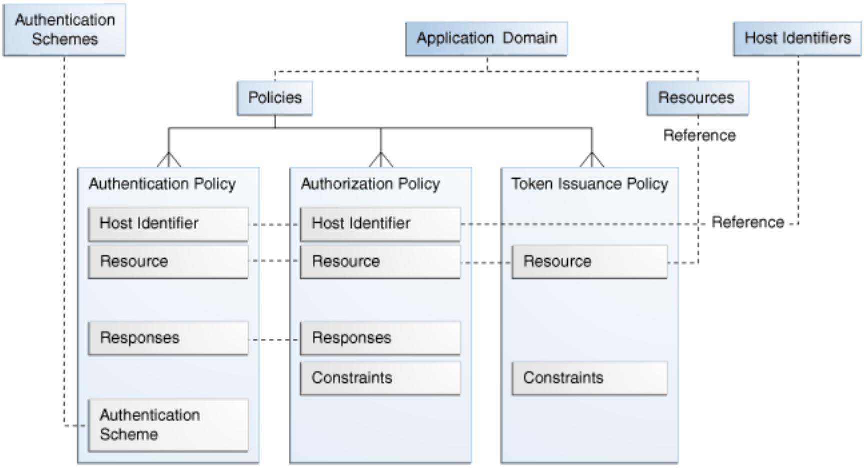
OAM 11g default behavior is to deny access when a resource is not protected by a policy that explicitly allows access.

> **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 13–1 illustrates Application Domain-specific components of the Oracle Access Manager 11g Policy Model.

Figure 13–1 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 and there is no Token Issuance Policy defined (just an empty container). After creating the application domain, you can add more resource definitions to it and then add resources to specific policies.

Whether you register an Agent using the Oracle Access Manager Console or the remote registration tool, the elements of an application domain are the same, as described in following topics:

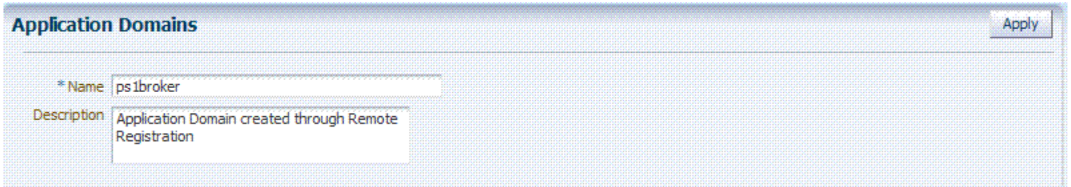
- Application Domain General Details
- Default Resources in a Generated Application Domain
- Default Authentication Policies in a Generated Application Domain
- Default Authorization Policies in a Generated Application Domain

> **See Also:** "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31 for details about the Token Issuance Policies.

13.3.1 Application Domain General Details

Figure 13–2 shows an application domain that is automatically created during Agent registration. The application domain name is highlighted in the navigation tree under the Application Domains node, and the Application Domain page with a name and description.

Figure 13–2 New Application Domain Generated During Agent Registration



* Name: ps1broker

Description: Application Domain created through Remote Registration

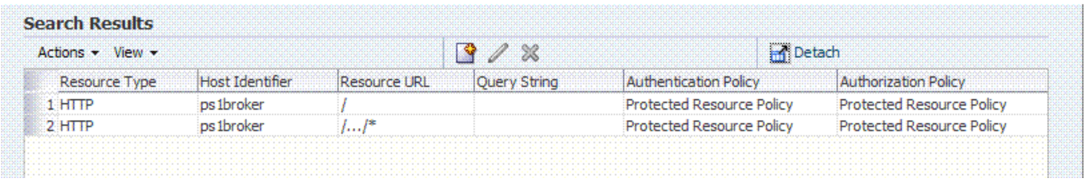
Beneath the application domain name in the navigation tree are the default resource definitions, and policies.

Administrators can modify the application domain to add more resources, and define policies, responses, and authorization constraints.

13.3.2 Default Resources in a Generated Application Domain

Figure 13–3 illustrates the default resources in the generated application domain, when you use the Auto Generate Policies function. The Host Identifier matches the Agent name that was specified during registration. The default Resource Type is HTTP; the default Resource URLs are /and /.../*.

Figure 13–3 Default Resource Definition in a Generated Application Domain



	Resource Type	Host Identifier	Resource URL	Query String	Authentication Policy	Authorization Policy
1	HTTP	ps1broker	/		Protected Resource Policy	Protected Resource Policy
2	HTTP	ps1broker	/.../*		Protected Resource Policy	Protected Resource Policy

You can add resources to this policy, drawing from list of the resources that have been added to the application domain. HTTP resource definitions enable you to enter a query string. This query string can contain only the Base URL, which can include optional pattern-matching special characters to represent a set of URLs. You can also search on a query string to locate a specific resource. The Search page includes a Create button to start a new resource definition.

> **See Also:** "Adding and Managing Resource Definitions for Use in Policies" on page 13-11

13.3.3 Default Authentication Policies in a Generated Application Domain

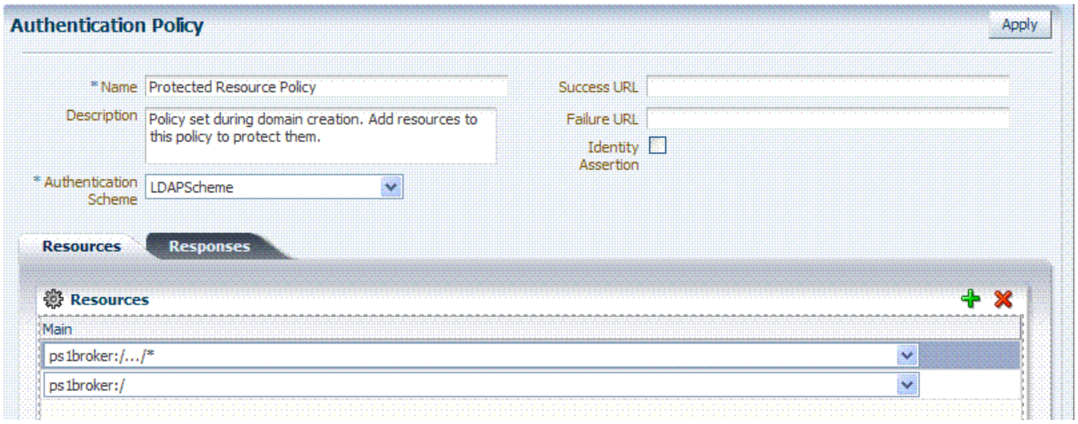
Each resource can be protected by only a single authentication policy. When an administrator creates an application domain manually she must also manually create all policies. However, when the application is generated automatically, during Agent registration for instance, the Protected Resource Policy is created are automatically generated:

The **Protected Resource Policy** is shown in Figure 13–4. 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 13–4 Default Authentication Policy for Protected Resources



Authentication Policy

* Name: Protected Resource Policy

Description: Policy set during domain creation. Add resources to this policy to protect them.

* Authentication Scheme: LDAPScheme

Success URL:

Failure URL:

Identity Assertion ☐

Resources

Main

ps1broker:/.../*

ps1broker:/

Authentication, **Public Resource Policy**: A second authentication policy is also created, which uses AnonymousScheme as its default for authentication. The description tells administrators "`Policy set during domain creation. Add resources to this policy to allow anyone access.`"

Note: This Public Resource Policy does not initially protect any Resources.

See Also: "Introduction to Policy Responses for SSO" on page 13-32

13.3.4 Default Authorization Policies in a Generated Application Domain

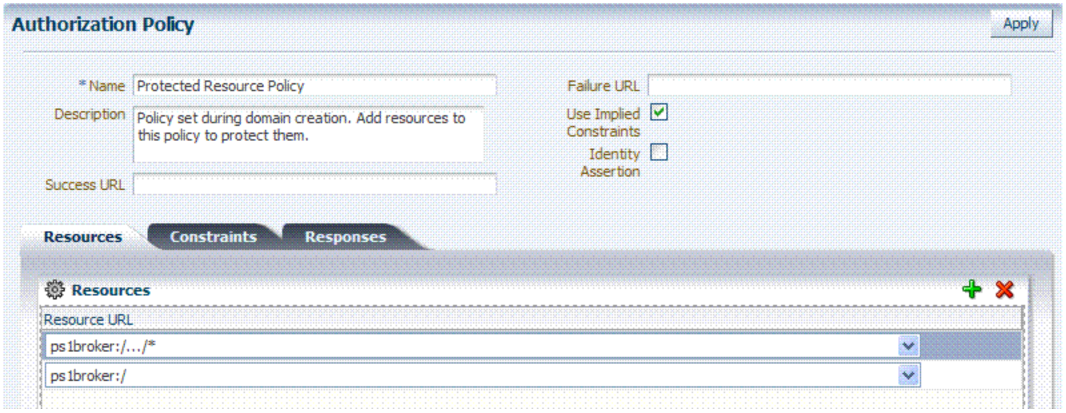
Each resource can be protected by only a single authorization policy. Administrators who manually create an application domain must manually create all policies. However when the application is generated automatically, during Agent registration for instance, the following authorization policies are automatically generated:

- Authentication Policy: Protected Resource Policy
- Authorization Policy: Protected 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 13–5. 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 13–5 Default Authorization Policy for Protecting Resources



* 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 ☑

Identity Assertion ☐

Resources

Resource URL
ps1broker:/.../*
ps1broker:/

In the previous release, a second authorization policy (Public) was also created, which initially did not protect any resources. Beginning with patch set 1, this second policy is not set.

See Also:

- "Introduction to Policy Responses for SSO" on page 13-32
- "Introduction to Authorization Constraints" on page 13-39

13.3.5 About Token Issuance Policies

By default, only a container for Token Issuance Policies is provided in an application domain that is generated when you register an agent.

For information on this policy type, resources, and possible constraints, see:

- "Managing TokenServiceRP Type Resources" on page 19-34
- "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31

13.4 Managing Application Domains using the Console

This section provides the following topics:

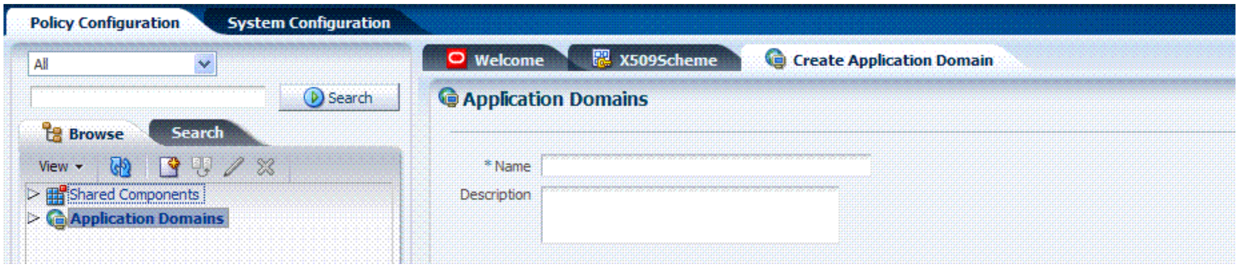
- About the Application Domains Page
- Creating a Fresh Application Domain Manually
- Searching for an Application Domain
- Viewing or Editing an Application Domain
- Deleting an Application Domain and Its Content

13.4.1 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 Oracle Access Manager Console, several pages are involved. Initially, you add general details (name and optional description) on the form show in Figure 13–6.

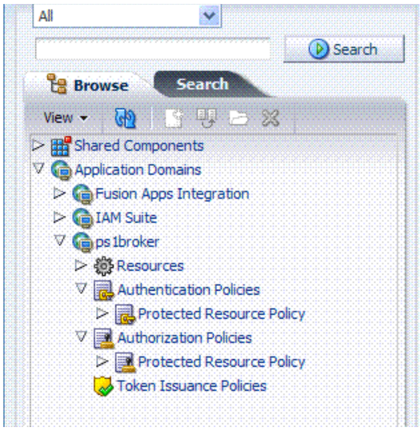
Figure 13–6 Fresh Application Domains General Page



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 13–7 illustrates the Application Domains navigation tree and the related containers for resources policies for one domain.

Figure 13–7 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.

13.4.2 Creating a Fresh Application Domain Manually

Users with valid Administrator credentials can perform the following task to manually create an application domain using the Oracle Access Manager Console.

Note: Application domains can be generated automatically during Agent Registration, as described in Chapter 9 and Chapter 10.

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 13-10.

> **See Also:** "Introduction to Application Domain Creation" on page 13-2

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 New Application Domain 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 13-11.
   - **Authentication Policies**: See "Defining Authentication Policies for Specific Resources" on page 13-22.
   - **Authorization Policies**: See "Defining Authorization Policies for Specific Resources" on page 13-27.
   - **Token Issuance Policies**: See "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31.

13.4.3 Searching for an Application Domain

Users with valid Administrator credentials can use the following procedure to search for a specific application domain.

> **See Also:** "About Policy Configuration Search Controls" on page 3-18

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.

13.4.4 Viewing or Editing an Application Domain

Users with valid Administrator credentials can perform the following task to view or modify an application domain (including its resources, policies, constraints, and responses) using theOracle Access Manager 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 13-11.
   - **Authentication Policies**: See "Defining Authentication Policies for Specific Resources" on page 13-22.
   - **Authorization Policies**: See "Defining Authorization Policies for Specific Resources" on page 13-27
   - **Token Issuance Policies**: See "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31.
3. Click Apply to submit the changes (or close the page without applying changes).

13.4.5 Deleting an Application Domain and Its Content

Users with valid Administrator credentials can perform the following task to delete an application domain (including its resources, policies, constraints, and responses) using the Oracle Access Manager Console.

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 Application Domain node.
2. In the navigation tree, open and review the application domain name.
3. Select the Application Domain 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. Check the navigation tree to confirm the application domain has been removed.

13.5 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 non-HTTP/HTTPS-based resources and HTTP/HTTPS-based resources such as:

- 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 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

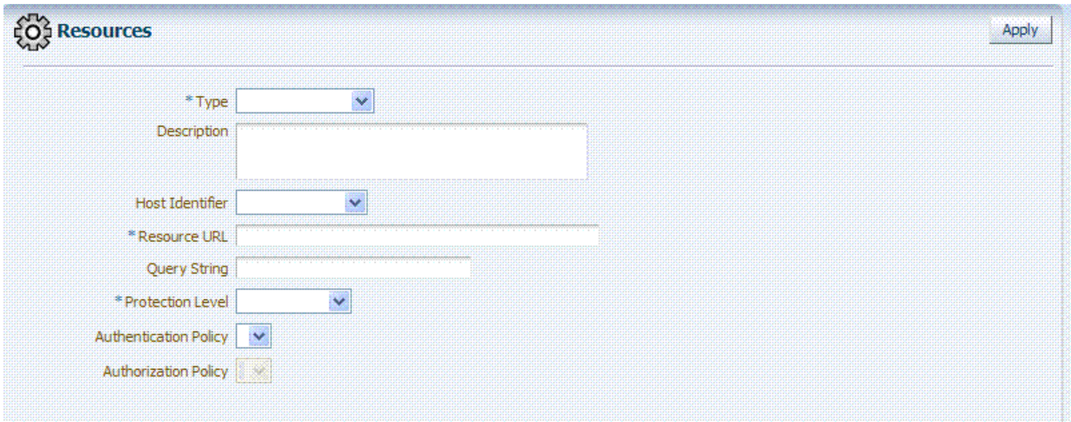
13.5.1 About the Resource Definition Page in an Application Domain

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. If a resource that is not explicitly marked as excluded, is not associated with a policy, then access is denied to all users because there is no policy match.

Figure 13–8 illustrates a fresh Resources definition page.

Figure 13–8 Resources Page in an Application Domain



Resources

Apply

* Type

Description

Host Identifier

* Resource URL

Query String

* Protection Level

Authentication Policy

Authorization Policy

Table 13–1 describes elements that comprise a resource definition.

Table 13–1 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. The `wl_authen` resource type is used for Fusion Middleware application scenarios, as described in the Oracle Fusion Middleware Application Security Guide. See Also "Managing Resource Types" on page 12-2. The `TokenServiceRP` resource type is used to represent the Token Service Relying Party as described in "Managing TokenServiceRP Type Resources" on page 19-34.
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 12-5.
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. 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 13–2.

Table 13–1 (Cont.) Resource Definition Elements

Elements	Description

Query String	The Policy Model supports query string-based HTTP resource definitions within Access Policies. The following formats are allowed:
	Supported: Resource protection based on literal full query string matching. A single Query String Pattern that would be matched against the entire input Query string (as opposed to matching only portions (selected name value pairs) of the query string. For example:
	`status=active&adminrole=*`
	A Query String pattern specified as a regular free form String with these extra features:
	■ Optional: Special character (*) that matches zero or more characters, which is applied to a set of names in the run time Query String.
	■ Two resource definitions can exist with same URL base path pattern and different Query String patterns. These two are independent and non-equal resources. For example, these are all valid and can exist at same time:
	`/foo`
	`/foo?bar=true`
	`/foo?bar=false`
	The Query String is free form with no restriction in terms of format or characters. It is not required to specify Query String as key/value pairs
	At run time, onlythe Query String that is part of HTTP GET requests is processed; Query String pattern does not apply to HTTP POST data.
	Resource Matching (at run time):
	■ The base URL path is matched and then the Query String is matched
	■ Multiple resource patterns that contain matching Query Strings: The best match is determined based on the number of tokens (pattern delimited by '*') and thegth length of the token at each position. Patterns with longer tokens in the beginning are preferred and then the pattern that contains more number of tokens. (If there are matching patterns that contain same number of tokens and same length at each position then the match would fail.)
	Conflicts:
	■ **Super Set**: The input resource definition contains a set of name-value Query String patterns that are a super set of patterns of an existing resource definition in the policy store.
	■ **Overlap**: The input resource specification contains a set of name-value Query string patterns that overlap a set of patterns of an existing resource definition in the policy store.
	Note: For OAM Agents, the remote registration tool (oamreg) accepts Query-string based HTTP resource definitions and generates the relevant policy objects for securing access of these resources. If any conflicts are encountered during policy provisioning, only policies for resources that do not have any conflicts are provisioned. This feature does not apply to 10g OSSO agent- based partners and applications. OSSO agents are not capable of enforcing authentication scheme per resource. Instead, a single authentication scheme is applied to all resources of an application.

Table 13–1 (Cont.) Resource Definition Elements

Elements	Description
Protection Level	Choose the appropriate protection level from the following: ■ Protected (the default) Protected resources are associated with a protected-level Authentication policy that uses a variety of authentication schemes (LDAP, or example). Authorization policies are allowed for protected resources. Responses, constraints, auditing, and session management are enabled for protected resources using a policy that protects the resource. ■ Unprotected Unprotected resources are associated with an unprotected-level Authentication policy (level 0) that can use a variety of authentication schemes (LDAP, for example). Authorization policies are allowed for unprotected resources, and a basic one is needed to allow such access. However, an elaborate policy with constraints and responses is irrelevant. Responses, constraints, and auditing are enabled for Unprotected resources using a policy that protects the resource. Only Session Management is not enabled. Access to Unprotected resources incur an OAM Server check from Webgate, which can be audited. ■ Excluded (these are public) Only HTTP resource types can be excluded. Typically security insensitive files like Images (*.jpg, *.png), protection level Excluded resources do not require an OAM Server check for Authentication, Authorization, Response processing, Session management, and Auditing. Excluded resources cannot be added to any user-defined policy in the console. The Webgate does not contact the OAM Server while allowing access to excluded resources; therefore, such access is not audited. Most regular resource validations apply to Excluded resources. However, excluded resources are not listed when you add resources to a policy. There is no Authentication or Authorization associated with the resource. Note: If a resource protection level is modified from "Protected" to "Excluded" and a policy exists for that resource, modification will fail until the resource is first disassociated with the policy.
Authentication Policy	A list of Authentication policies based on the specified resource protection level becomes available. Only policies within this domain, and that match the specified protection level, are listed.
Authorization Policy	A list of authorization policies defined in the domain become available from which you can choose.

Table 13–2 shows sample URL values for resources. For more information, see "About the Resource URL" on page 13-16.

Table 13–2 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

Table 13–2 (Cont.) HTTP Resources Sample URL Values

Resource	Sample URL Values
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

13.5.1.1 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.

13.5.1.2 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.

13.5.1.3 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:** No host identifier is 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, Query String, 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 13–3 illustrates a number of resource definitions within a single application domain. These are organized alphabetically according to the Host Identifier and Resource URLs.

Table 13–3 Resource URLs for.jsp

Resource URL	Correct
/bank/accounts/*	Yes
/bank/accounts/*.jsp	Yes
/bank/accounts/checking	Yes
/bank/.../checking.jsp	Yes
/.../*.jsp	Yes
/bank/accounts/checking*.jsp	No
/bank/accounts/c*.jsp	No
/bank/.../accounts/def.gif	No

13.5.1.4 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 8-8

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 13–4 describes the possible outcomes.

Table 13–4 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

Table 13–4 (Cont.) Resource Evaluation Outcomes

Outcome	Description
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, *xyz*`index.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"

13.5.2 Adding Resource Definitions to an Application Domain

Users with valid 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 12-2.

> **See Also:** "About the Resource Definition Page in an Application Domain" on page 13-11

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. Within the application domain, open the Resources node.
3. Click the New Resource button in the upper-right corner of the Search page.
4. On the Resource Definition page:

   a. Select or enter your details for a single resource (Table 13–1):

      Type
      Description (optional)
      Host Identifier
      Resource URL
      Query String
      Protection Level
      Authentication Policy (*if level is Protected*)
      Authorization Policy (if *level is Protected and Athentication Policy is chosen*)

   b. Click Apply to add this resource to the application domain.

   c. Repeat this procedure to add other resources to this application domain.

5. Proceed to adding a resource to specific policies as described in:

   - Defining Authentication Policies for Specific Resources
   - Defining Authorization Policies for Specific Resources
   - Managing Token Issuance Policies and Constraints with Oracle Access Manager

13.5.3 Searching for a Resource Definition

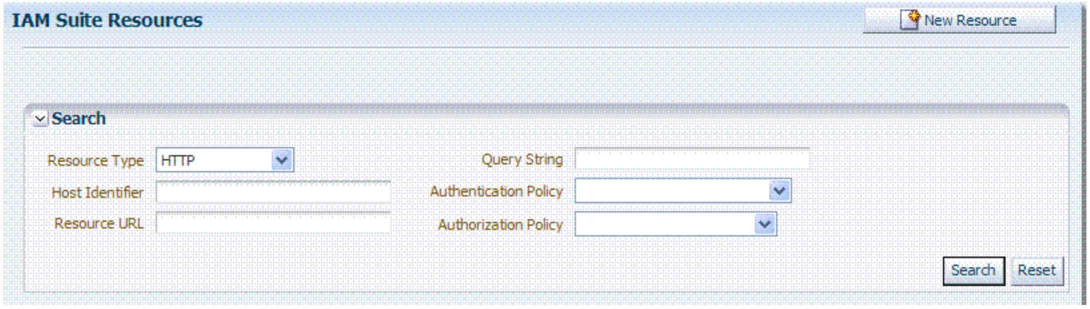
This section provides the following topics:

- About Searching for a Specific Resource Definition
- Searching for a Specific Resource Definition

13.5.3.1 About Searching for a Specific Resource Definition

Figure 13–9 shows the default Search elements and Search Results table for resource definitions in an application domain.

Figure 13–9 Searching for Resource Definitions within an Application Domain



Resource Type: HTTP

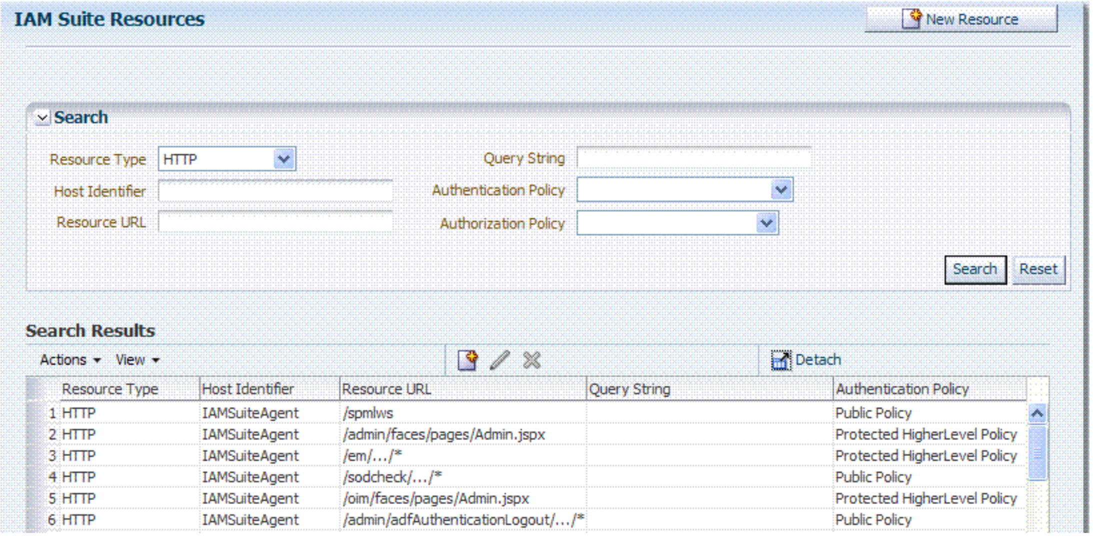
You can simply click the Search button using the defaults or refine your search by supplying as much or as little of the information in Table 13–5 as needed to find the resource.

Table 13–5 Search Elements for a Resource in an Application Domain

Search Elements	Description
Resource Type	Provides a list of defined resource types from which you can choose. You can also leave this blank. Default: HTTP
Host Identifier	Enter a host identifier here, if desired. You can leave this blank Default: blank
Resource URL	Enter a resource URL, if desired. You can leave this blank Default: blank
Query String	Enter a query string for the resource, or leave this blank. You can include this in the search criteria if a query string was defined for the resource when it was added to the application domain. Default: blank
Authentication Policy	Provides a list of defined authentication policies for this application domain. You can choose one or leave the space blank. Default: blank
Authorization Policy	Provides a list of defined authorization policies for this application domain. You can choose one or leave the space blank. Default: blank

You can click Reset to clear the form or Search to initiate the search. The results table is shown in Example 13–10. Each resource listed includes everything specified when it was added to the domain. The Actions and View menus are available for use with the table. Also you can click the Create command button to add a new resource definition to this domain.

Figure 13–10 Search Results Table for Resource Definitions in an Application Domain



Resource Type: HTTP
Host Identifier:
Resource URL:
Query String:
Authentication Policy:
Authorization Policy:

	Resource Type	Host Identifier	Resource URL	Query String	Authentication Policy
1	HTTP	IAMSuiteAgent	/spmlws		Public Policy
2	HTTP	IAMSuiteAgent	/admin/faces/pages/Admin.jspx		Protected HigherLevel Policy
3	HTTP	IAMSuiteAgent	/em/.../*		Protected HigherLevel Policy
4	HTTP	IAMSuiteAgent	/sodcheck/.../*		Public Policy
5	HTTP	IAMSuiteAgent	/oim/faces/pages/Admin.jspx		Protected HigherLevel Policy
6	HTTP	IAMSuiteAgent	/admin/adfAuthenticationLogout/.../*		Public Policy

13.5.3.2 Searching for a Specific Resource Definition

Users with valid Administrator credentials can use the following procedure to search for a specific resource definition.

> **See Also:** "About Policy Configuration Search Controls" on page 3-18

To find a resource definition

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
     Desired Domain

2. Open the Resources node to display related Search criteria and the Search Results table.
3. Fill in the search criteria as needed (Table 13–5), and click the Search button.
4. In the Search Results table, click the desired resource definition:
   - **Actions**: Select an Actions menu item to Create a new resource in the domain or to Edit or Delete the selected resource in the table.
   - **View**: Select a View menu item to alter the appearance of the results table.
   - **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.
5. Click the Browse tab to return to the navigation tree when you finish with the Search results.

13.5.4 Viewing or Editing a Resource Definition in an Application Domain

Users with valid Administrator credentials can use the following procedure to modify resource definitions within a specific application domain.

> **Note:** If a resource protection level is modified from "Protected" to "Excluded" while it is associated with a policy, the modification will fail. First, remove the resource from the policy, make the change, and add the resource to the policy.

Prerequisites

You must have the desired resource type defined as a shared component. For details, see "Managing Resource Types" on page 12-2.

> **See Also:** "About the Resource Definition Page in an Application Domain" on page 13-11

To view or modify resource definitions in an application domain

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

   Application Domains node
     Desired Domain

2. Find the Resource, as described in "Searching for a Resource Definition".
   - 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).

13.5.5 Deleting a Resource Definition from an Application Domain

Users with valid 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

2. Find the Resource, as described in "Searching for a Resource Definition"
3. Optional: Double-click the desired resource definition and confirm this is the one to be deleted, then close the page.
4. Click the name of the desired resource definition and then click the Delete button in the tool bar.
5. In the Confirmation window, click Delete (or click Cancel to dismiss the window).
6. Repeat this procedure as often as needed to delete other resources in an application domain.

13.6 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 13-3

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

13.6.1 About the Authentication Policy Page

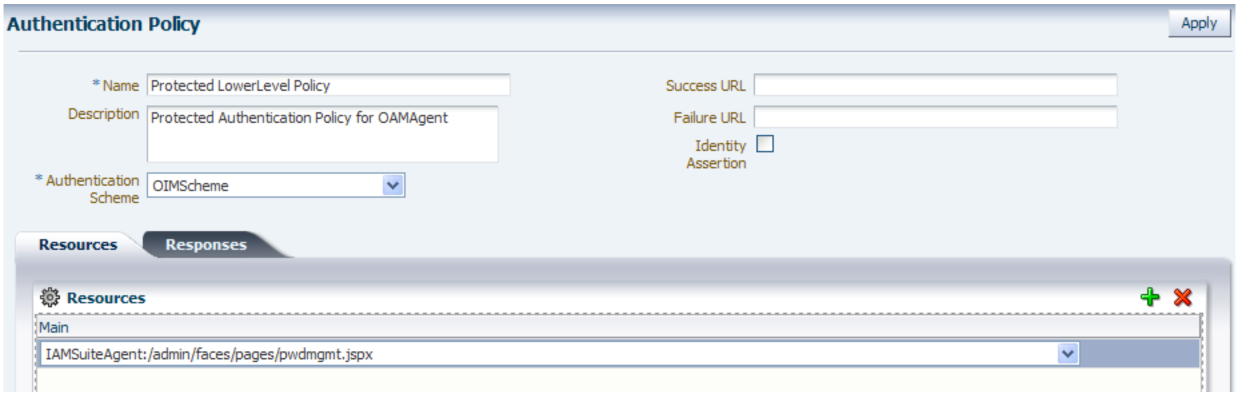
Administrators use authentication policies to protect specific resources. The authentication policy provides the sole authentication method for resources governed by the policy.

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.

Figure 13–11 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 IAM Suite application domain.

Figure 13–11 Authentication Policy: IAM Suite Application Domain



Authentication Policy

* Name: Protected LowerLevel Policy

Description: Protected Authentication Policy for OAMAgent

* Authentication Scheme: OIMScheme

Success URL:

Failure URL:

Identity Assertion □

Resources

Main

IAMSuiteAgent:/admin/faces/pages/pwdmgmt.jspx

Table 13–6 describes authentication policy elements. The IAM Suite application domain is shown simply as an example.

Table 13–6 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 12-15 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 13–6 (Cont.) Authentication Policy Elements and Descriptions

Element	Description
Identity Assertion	Required in the ID propagation use case scenario for any issued token from Oracle Access Manager that represents an end user (and possibly its OAM session). The Identity Assertion Token is generated and returned as a response (as an HTTP HEADER with a name of "OAM_IDENTITY_ASSERTION" and a value as a SAML token) after a successful authentication. For Oracle Security Token Service clients that are Web applications protected by OAM 11g requesting tokens to gain proxy access to a Relying Party (ID Propagation use case), Oracle Security Token Service requires clients to pass an OAM Identity Assertion token that represents the end user. The ID Provider (Oracle Access Manager) processes the request and returns an Authentication Token and an ID Assertion Token. An ID Assertion Token, in itself, does not represent a user session and cannot be used independently to request direct access to a resource or service. The ID Assertion Token Requestor uses this Token later, during the end user session, as part of a backend service processing request (on behalf of the end user). The ID Assertion Token Consumer (Oracle Security Token Service), as part of the request processing, first validates the ID Assertion Token and then (on validation success) processes the request in the context of the end user Identity. See Also: "Scenario: Identity Propagation with the OAM Token" on page 16-2.
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 13-24.
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 13-32.

13.6.1.1 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.

13.6.2 Adding an Authentication Policy and Resources

Users with valid 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 13-23
- "Managing Authentication Schemes" on page 12-15

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

2. In the navigation tree, click Authentication Policies, then click the Create button to open a fresh page.
3. **Add General Policy Details**:
   - Name
   - Authentication Scheme
4. **Add Global Policy Elements** and Specifications (Table 13–6):
   - Description (optional)
   - Success URL
   - Failure URL
   - Identity Assertion
5. **Add Resources**:
   - Click the Resources tab on the Authentication Policy page.
   - Click the Add button on the tab.
   - Choose 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 13-37.
8. Close the page when you finish.

13.6.3 Searching for an Authentication Policy

Users with valid Administrator credentials can use the following procedure to search for a specific authentication policy.

> **See Also:** "About Policy Configuration Search Controls" on page 3-18

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.

13.6.4 Viewing or Editing an Authentication Policy

Users with valid 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 13-23

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 13–6):
   - Description
   - Success URL
   - Failure URL
   - Identity Assertion for Oracle Security Token Service
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 13-37.
8. Close the page when you finish.

13.6.5 Deleting an Authentication Policy

Users with valid 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 13-23

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 13-37.
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.

13.7 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 13-3

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

13.7.1 About Authorization Policies for Specific Resources

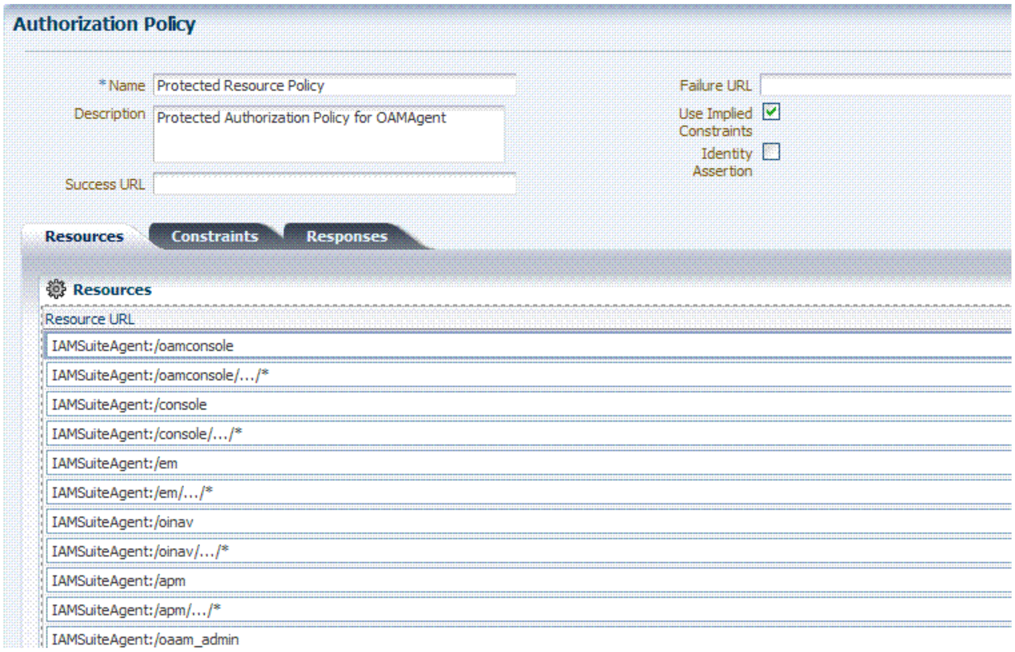
Administrators can create an authorization policy to protect access to one or more resources based on attributes of an authenticated user or the environment. The authorization policy provides the sole authorization protection for resources included in the policy.

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 13–12 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 IAM Suite Application Domain is shown simply as an example.

Figure 13–12 Authorization Policy Page: IAM Suite Application Domain



Authorization Policy

* Name: Protected Resource Policy

Description: Protected Authorization Policy for OAMAgent

Success URL:

Failure URL:

Use Implied Constraints ☑

Identity Assertion □

Resource URL
IAMSuiteAgent:/oamconsole
IAMSuiteAgent:/oamconsole/.../*
IAMSuiteAgent:/console
IAMSuiteAgent:/console/.../*
IAMSuiteAgent:/em
IAMSuiteAgent:/em/.../*
IAMSuiteAgent:/oinav
IAMSuiteAgent:/oinav/.../*
IAMSuiteAgent:/apm
IAMSuiteAgent:/apm/.../*
IAMSuiteAgent:/oaam_admin

Table 13–7 describes authorization policy elements. The elements are the same regardless of the domain; only the details will differ. The IAM Suite Application Domain application domain is shown simply as an example.

Table 13–7 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 13-39.

Table 13–7 (Cont.) Authorization Policy Elements and Descriptions

Element	Description
Identity Assertion	Required in the ID propagation use case scenario for any issued token from OAM that represents an end user (and possibly its OAM session). For OSTS clients that are Web applications protected by OAM 11g requesting tokens to gain proxy access to a Relying Party (ID Propagation use case), OSTS requires clients to pass an OAM Identity Assertion token that represents the end user. The ID Provider (Oracle Access Manager) processes the request and returns an Authentication Token and an ID Assertion Token. An ID Assertion Token, in itself, does not represent a user session and cannot be used independently to request direct access to a resource or service. The ID Assertion Token Requestor uses this Token later, during the end user session, as part of a backend service processing request (on behalf of the end user). The ID Assertion Token Consumer (Oracle Security Token Service), as part of the request processing, first validates the ID Assertion Token and then (on validation success) processes the request in the context of the end user Identity. See Also: "Scenario: Identity Propagation with the OAM Token" on page 16-2.
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 13-39.
Responses Tab	See Also "Introduction to Policy Responses for SSO" on page 13-32.

13.7.2 Adding an Authorization Policy and Specific Resources

Users with valid 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 13-28

To add an authorization policy and resources

1. From the Policy Configuration tab, navigation tree, expand the following nodes:

   Application Domains
   Desired Domain

2. In the navigation tree, click Authorization Policies and then the Create button.
3. Enter a unique name for this authorization policy.
4. **Global Policy Elements**: Enter your own details (Table 13–7):
   - Description (optional)
   - Success URL
   - Failure URL
   - Use Implied Constraints
   - Identity Assertion for Oracle Security Token Service
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 13-46.
8. **Responses**: Add or edit responses for SSO, as described in "Adding and Managing Policy Responses for SSO" on page 13-37.
9. Close the page when you finish.

13.7.3 Searching for an Authorization Policy

Users with valid Administrator credentials can use the following procedure to locate a specific authorization policy.

> **See Also:** "About Policy Configuration Search Controls" on page 3-18

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.

13.7.4 Viewing or Editing an Authorization Policy and Resources

Users with valid 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 13-28

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 13–7):
   - Description (optional)
   - Success URL
   - Failure URL
   - Use Implied Constraints
   - Identity Assertion for Oracle Security Token Service
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 13-49.
7. **Responses**: View or edit these as described in "Viewing, Editing, or Deleting a Policy Response for SSO" on page 13-38.
8. Close the page when you finish.

13.7.5 Deleting an Authorization Policy

Users with valid 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 13-28

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.

13.8 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.

> **Note:** There are no responses in Token Issuance Policies.

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

13.8.1 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).

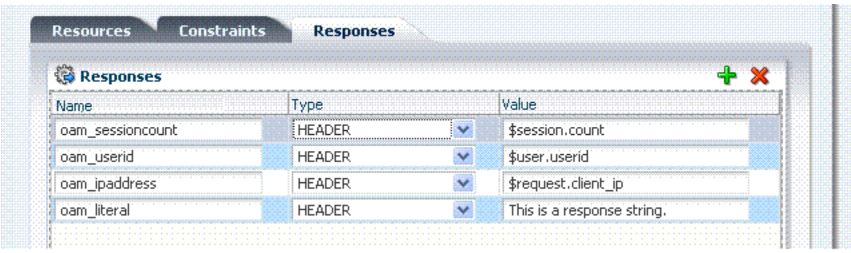
Policy responses enable the insertion of information into a session or application and the ability to withdraw the information at a later time to enable SSO. For instance, identity mappings can be inserted into the customer's application or actions can be carried out by the Agent or the application.

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 13–13 illustrates an Authorization Policy Response defined by an administrator in the Oracle Access Manager Console. Authorization responses can operate in conjunction with authorization constraints.

Figure 13–13 Authorization Policy Response in the 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 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 Oracle Access Manager Console, as described Table 13–8.

Table 13–8 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".

13.8.2 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`

13.8.3 About the Namespace and Variable Names for Policy Responses

With the namespace mechanism, the following variable types are 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 13–9, " Namespace Request Variables for Single Sign-On"
- Table 13–10, " Namespace Session Variables for Single Sign-On"
- Table 13–11, " Namespace User Variables"

Table 13–9 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 13–10 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 13–11 Namespace User Variables

Namespace	Description
attr	Reference to an arbitrary LDAP attribute, the name of which is passed to us as a variable attribute

Table 13–11 (Cont.) Namespace User Variables

Namespace	Description
groups	Comma-separated list of the user's group membership
userid	The user ID
user.id_domain	The user's identity domain (essentially the same as the identity store)
guid	A unique identifier that locates the user entry in an Identity Store

13.8.4 About Constructing a Policy Response for SSO

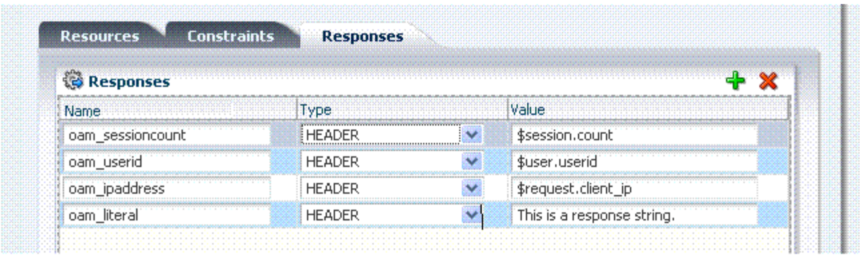
This section is divided as follows:

- Simple Responses
- Compound and Complex Responses

13.8.4.1 Simple Responses

After deciding on the response type and determining which namespace and variable, you simply enter the response attributes in the Oracle Access Manager Console. A simple response might look like one of the several authorization responses shown in Figure 13–14.

Figure 13–14 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 13–12 illustrates several simple responses and a description of what each one returns.

Table 13–12 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*

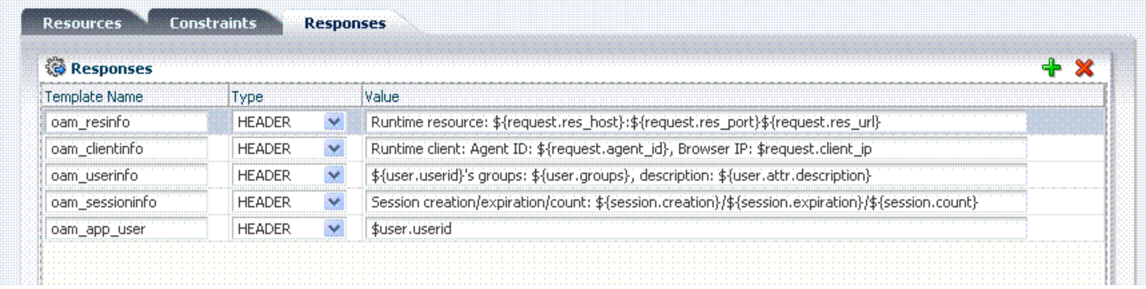
13.8.4.2 Compound and Complex Responses

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 13–15 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 13–15 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 13–13 describes the complex responses shown in Figure 13–15.

Table 13–13 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 2011/Wed Feb 24 01:47:42 PST 2011/7*
oam_app_user	$user.userid	HTTP_OAM_USERID *name*

For more information, see "About Policy Response Processing".

13.8.5 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.

13.9 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 13-32.

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

13.9.1 Adding a Policy Response for SSO

Users with valid 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 13-32

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 13-34

   - 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.

13.9.2 Viewing, Editing, or Deleting a Policy Response for SSO

Users with valid 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 13-32

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.

13.10 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

13.10.1 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.

13.10.2 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.

13.10.3 Guidelines for Authorization Responses Based on Constraints

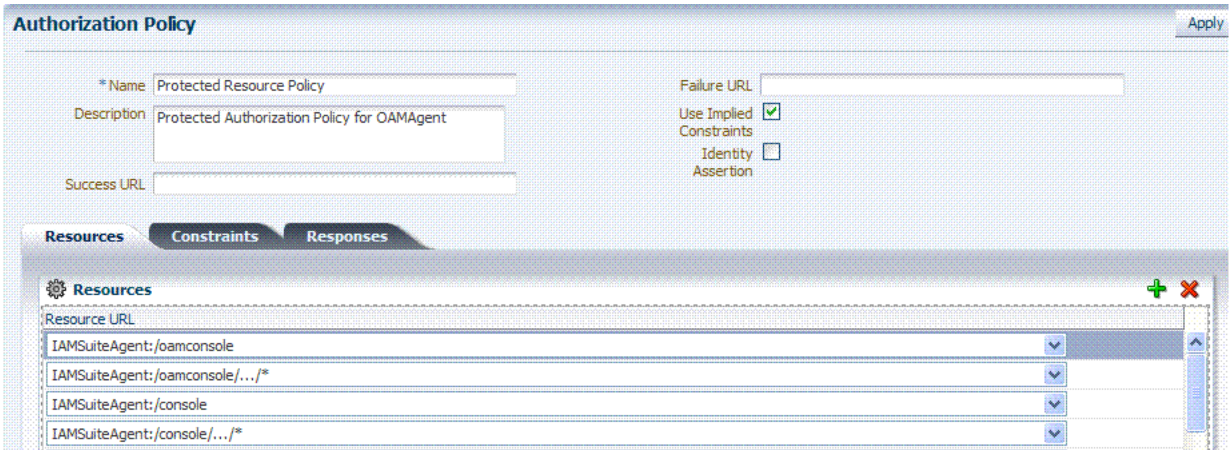
For each constraint class, consider the response actions that you want to occur for authorized users. For example, you may want the system to return user profile information and pass that information to a downstream application, as follows:

- If the user is authorized, you may want to pass the user's common name (cn) 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.

13.10.4 About Constraints and General Authorization Policy Details

All constraint definitions apply along with the general authorization policy details shown in Figure 13–16.

Figure 13–16 Authorization Policy Page, General Details



* Name	Protected Resource Policy
Description	Protected Authorization Policy for OAMAgent
Success URL	
Failure URL	
Use Implied Constraints	☑
Identity Assertion	□

Resources

Resource URL
IAMSuiteAgent:/oamconsole
IAMSuiteAgent:/oamconsole/.../*
IAMSuiteAgent:/console
IAMSuiteAgent:/console/.../*

Table 13–14 describes the common authorization policy general details.

Table 13–14 Authorization Policy General Details

Element	Description
Name	A unique name used as an identifier in the navigation tree.

Table 13–14 (Cont.) Authorization Policy General Details

Element	Description
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)
Identity Assertion	Required in the ID propagation use case scenario for any issued token from OAM that represents an end user (and possibly its OAM session). For OSTS clients that are Web applications protected by OAM 11g requesting tokens to gain proxy access to a Relying Party (ID Propagation use case), OSTS requires clients to pass an OAM Identity Assertion token that represents the end user. The ID Provider (Oracle Access Manager) processes the request and returns an Authentication Token and an ID Assertion Token. An ID Assertion Token, in itself, does not represent a user session and cannot be used independently to request direct access to a resource or service. The ID Assertion Token Requestor uses this Token later, during the end user session, as part of a backend service processing request (on behalf of the end user). The ID Assertion Token Consumer (Oracle Security Token Service), as part of the request processing, first validates the ID Assertion Token and then (on validation success) processes the request in the context of the end user Identity. See Also: "Scenario: Identity Propagation with the OAM Token" on page 16-2.

13.10.5 About the Add Constraint Window

When an administrator adds a constraint to an authorization policy, the window shown in Figure 13–17 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 13–17 Add Constraint Window

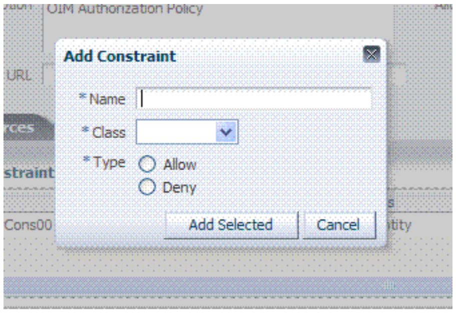


Table 13–15 describes the Add Constraint window elements.

Table 13–15 Add Constraint Window Elements

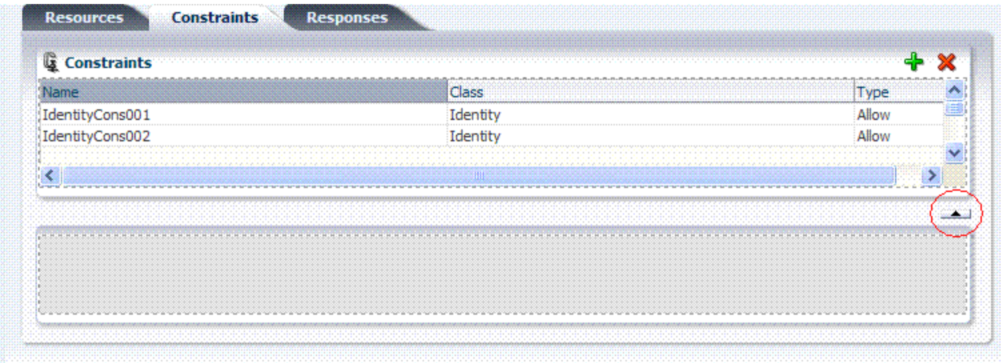
Element	Description
Name	A unique name for this constraint.
Class	Only one class can be specified (Identity, IP4 Range, or Temporal).

Table 13–15 (Cont.) Add Constraint Window Elements

Element	Description
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 13–18. 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 13–18 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

13.10.6 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 13–20.

Figure 13–19 Identity Class Constraint Details: Selected User and Groups Table

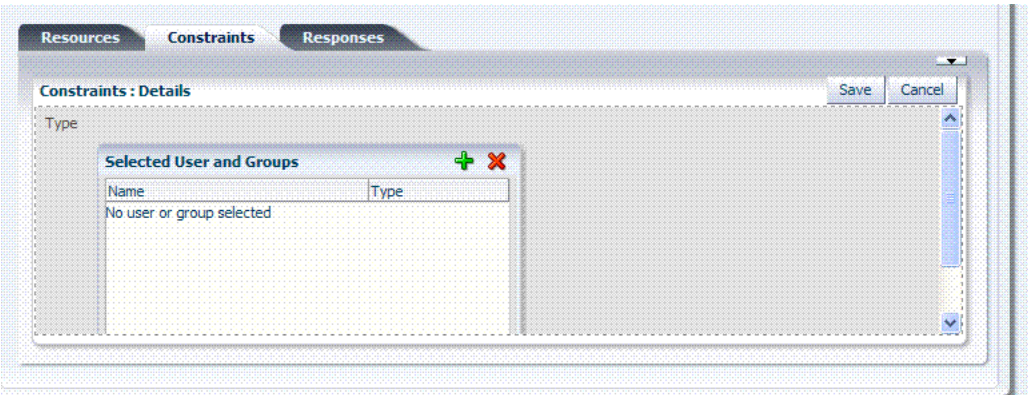
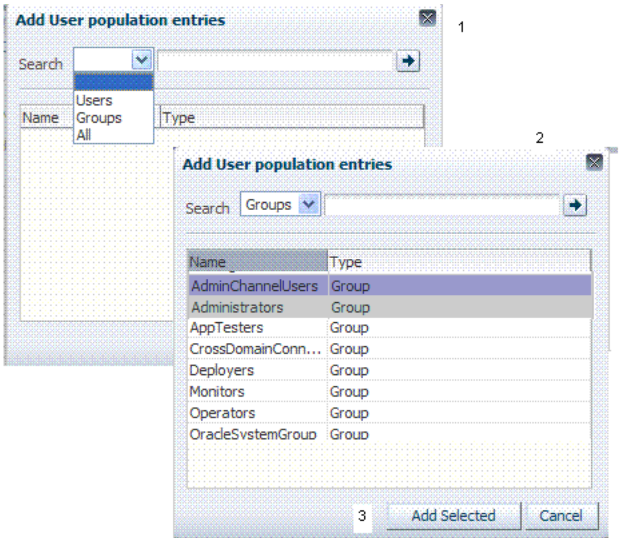


Figure 13–20 shows the Add User Population Entries window, which appears when you click the Add button on the Selected User and Groups table. From this table, you can choose from several populations: Users, Groups, or All.

Figure 13–20 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

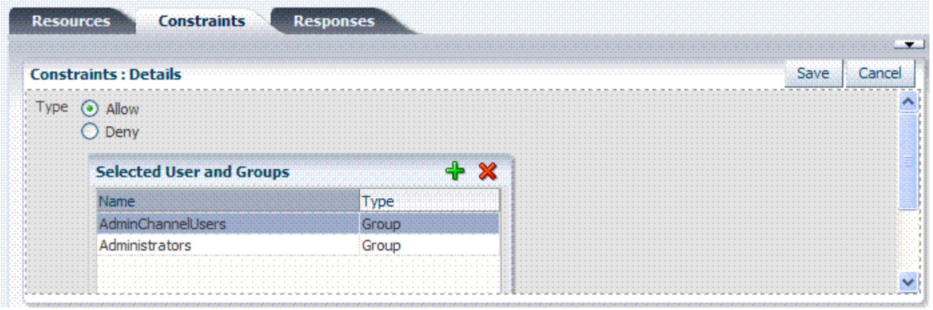
Table 13–16 describes the Add User Population Entries elements.

Table 13–16 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 13–21.

Figure 13–21 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.

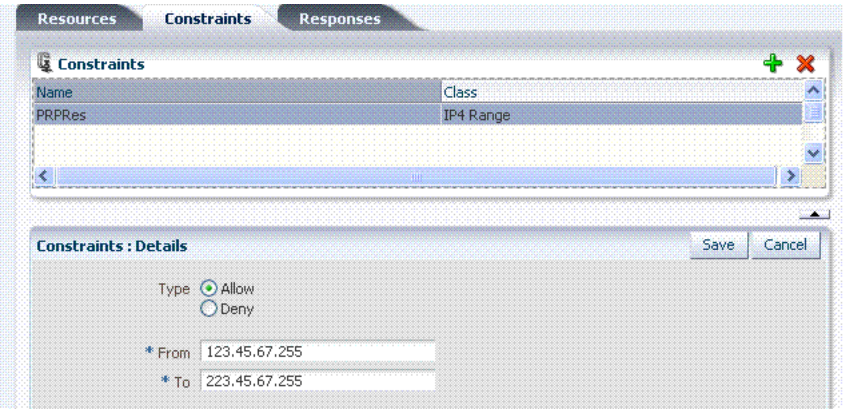
See Also: "Defining Identity Class Constraints" on page 13-46

13.10.7 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 13–22.

Figure 13–22 IP4Range Class Constraints



Name	Class
PRPRes	IP4 Range

Type ● Allow
○ Deny

* From 123.45.67.255

* To 223.45.67.255

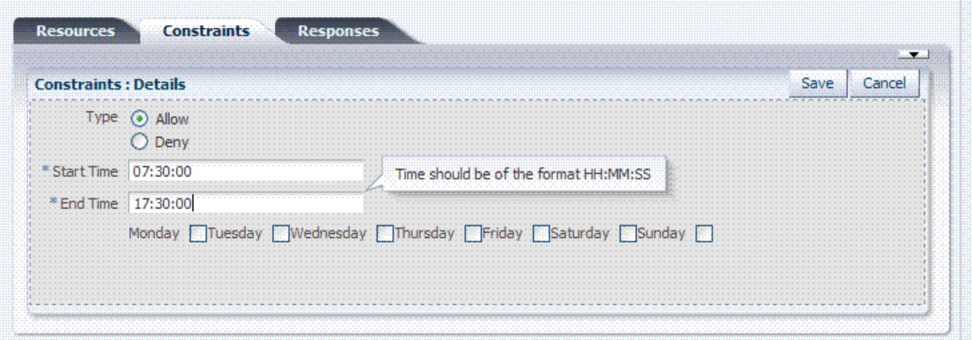
See Also: "Defining IP4Range Class Constraints" on page 13-47

13.10.8 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 13–20.

Figure 13–23 Temporal Constraint Class Details Page



Constraints : Details

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 13–17 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 13-48

13.11 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

13.11.1 Defining Identity Class Constraints

Users with valid 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 13-43

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 13–15) 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.

13.11.2 Defining IP4Range Class Constraints

Users with valid 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 13-45

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 13–15) 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 13–22):
   - 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.

13.11.3 Defining Temporal Class Constraints

Users with valid 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 13-45

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 13–15).
   - Click the option button beside the Type (Allow or Deny).
   - Click Add Selected in the Add Constraint window.
5. Add Temporal Details (Table 13–17): 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.

13.11.4 Viewing, Editing, or Deleting Authorization Policy Constraints

Users with valid 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 13-39

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 13-46

- "Defining IP4Range Class Constraints" on page 13-47
- "Defining Temporal Class Constraints" on page 13-48

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.

13.12 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 14, "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 Oracle Access Manager Console.

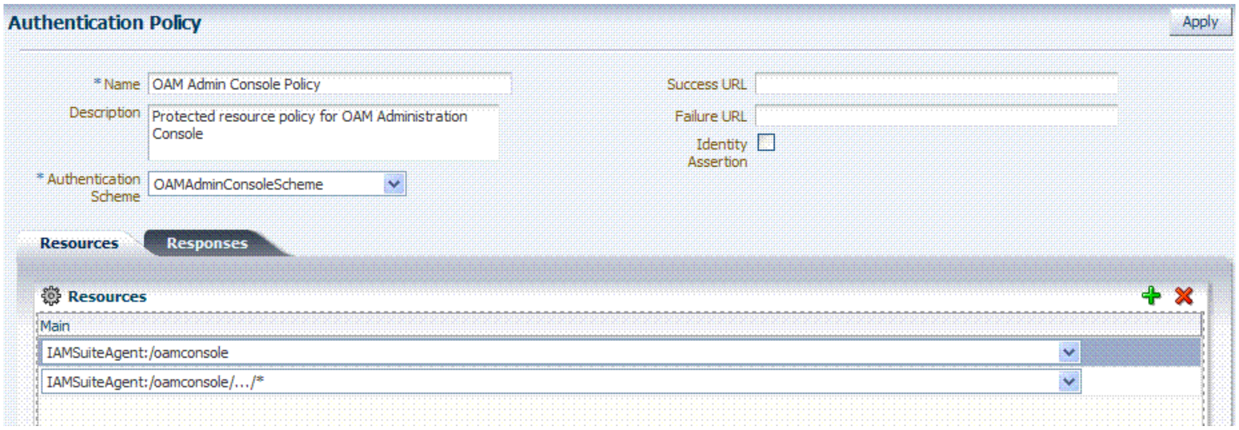
13.13 Example: Pre-seeded IAM Suite Application Domain and Policies

IAM Suite Authentication Policies, Schemes, and Resources

The following figures present Authentication Policies in the IAM Suite application domain:

- Figure 13–24, "OAM Admin Console Policy, Scheme, and Resources"
- Figure 13–25, "Protected HigherLevel Policy, LDAP Scheme, and Resources"
- Figure 13–26, "Protected LowerLevel Policy, Autentication Scheme, and Resources"
- Figure 13–27, "Public Policy, Anonymous Scheme, and Resources"

Figure 13–24 OAM Admin Console Policy, Scheme, and Resources



Authentication Policy

* Name: OAM Admin Console Policy

Description: Protected resource policy for OAM Administration Console

* Authentication Scheme: OAMAdminConsoleScheme

Success URL:

Failure URL:

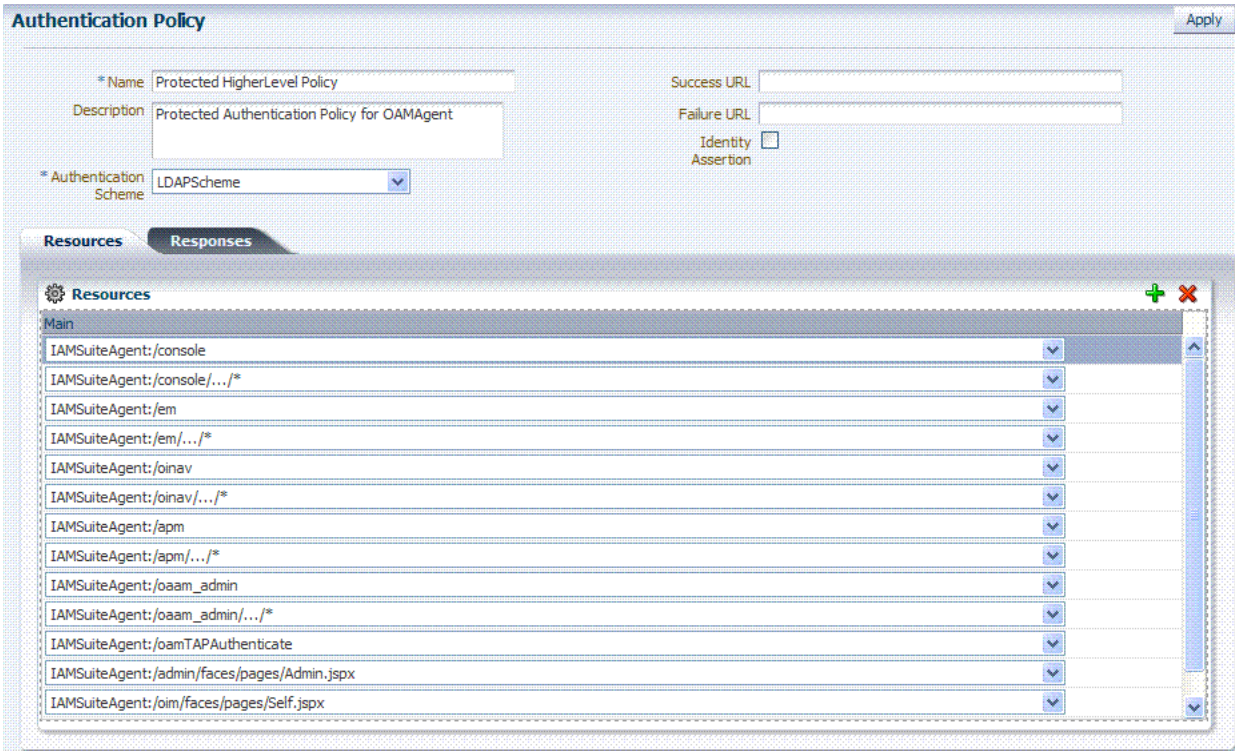
Identity Assertion: □

Main

IAMSuiteAgent:/oamconsole

IAMSuiteAgent:/oamconsole/…/*

Figure 13–25 Protected HigherLevel Policy, LDAP Scheme, and Resources



Authentication Policy

* Name: Protected HigherLevel Policy

Description: Protected Authentication Policy for OAMAgent

* Authentication Scheme: LDAPScheme

Success URL:

Failure URL:

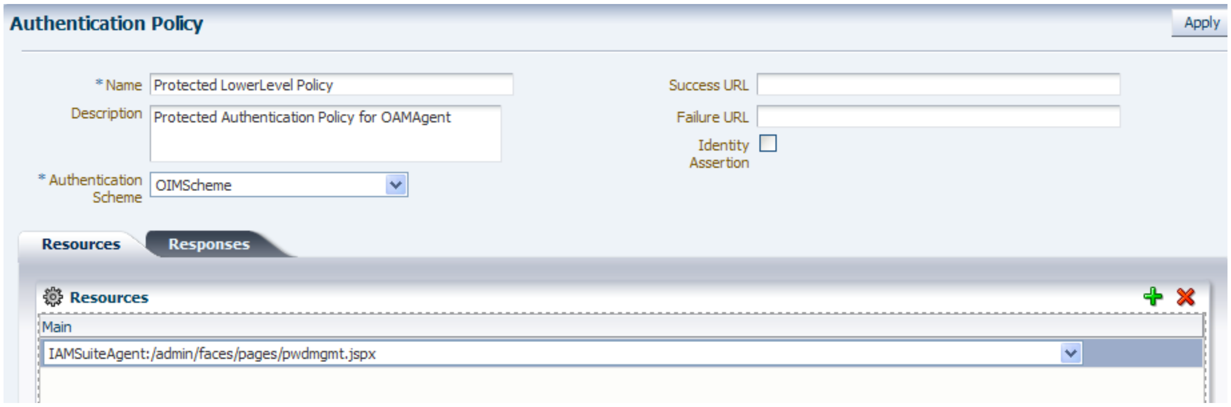
Identity Assertion: □

Resources

Main

IAMSuiteAgent:/console
IAMSuiteAgent:/console/…/*
IAMSuiteAgent:/em
IAMSuiteAgent:/em/…/*
IAMSuiteAgent:/oinav
IAMSuiteAgent:/oinav/…/*
IAMSuiteAgent:/apm
IAMSuiteAgent:/apm/…/*
IAMSuiteAgent:/oaam_admin
IAMSuiteAgent:/oaam_admin/…/*
IAMSuiteAgent:/oamTAPAuthenticate
IAMSuiteAgent:/admin/faces/pages/Admin.jspx
IAMSuiteAgent:/oim/faces/pages/Self.jspx

Figure 13–26 Protected LowerLevel Policy, Autentication Scheme, and Resources



Authentication Policy

* Name: Protected LowerLevel Policy

Description: Protected Authentication Policy for OAMAgent

* Authentication Scheme: OIMScheme

Success URL:

Failure URL:

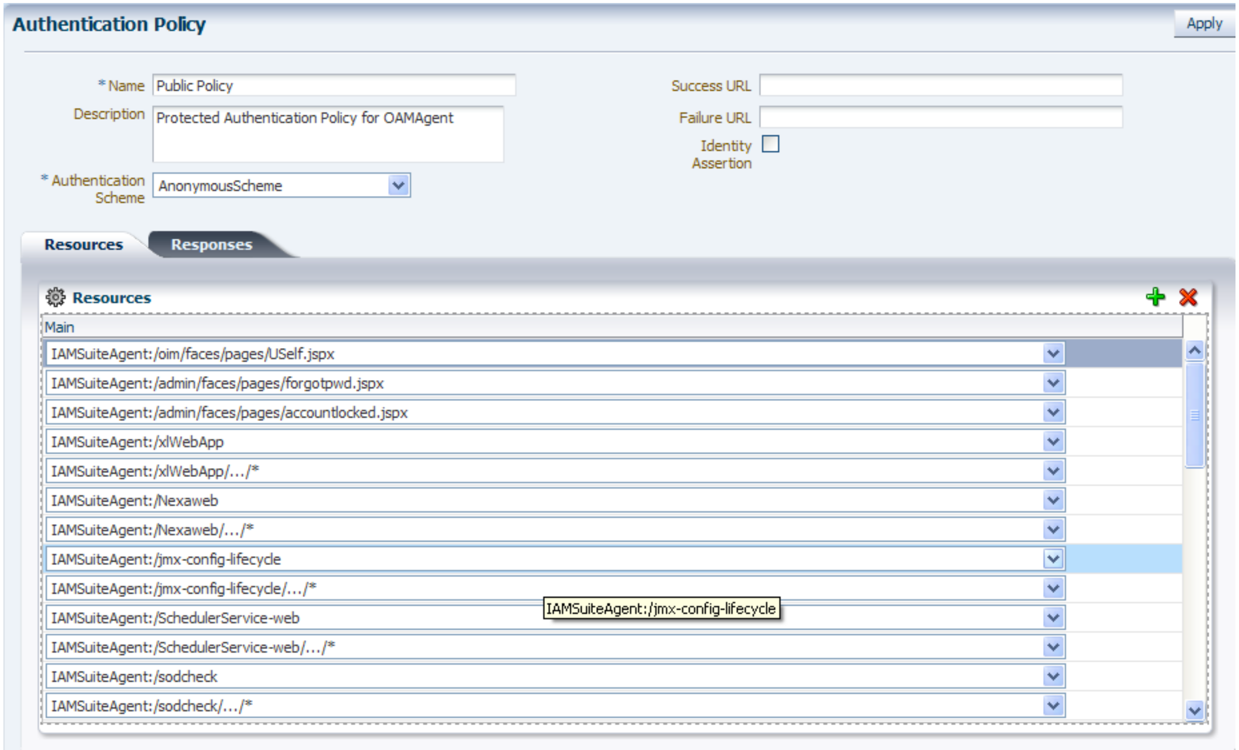
Identity Assertion: □

Resources

Main

IAMSuiteAgent:/admin/faces/pages/pwdmgmt.jspx

Figure 13–27 Public Policy, Anonymous Scheme, and Resources



Authentication Policy

Apply

* Name: Public Policy

Description: Protected Authentication Policy for OAMAgent

* Authentication Scheme: AnonymousScheme

Success URL:

Failure URL:

Identity Assertion: □

Resources | Responses

Resources

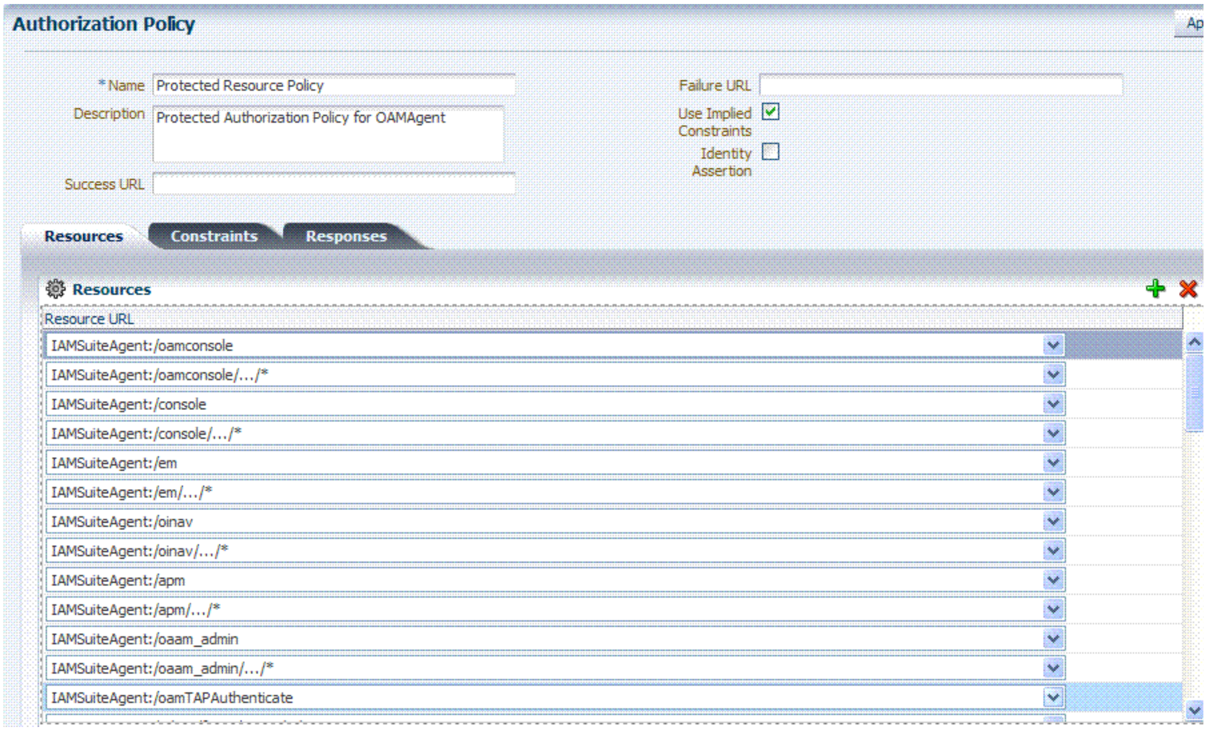
Main

- IAMSuiteAgent:/oim/faces/pages/USelf.jspx
- IAMSuiteAgent:/admin/faces/pages/forgotpwd.jspx
- IAMSuiteAgent:/admin/faces/pages/accountlocked.jspx
- IAMSuiteAgent:/xlWebApp
- IAMSuiteAgent:/xlWebApp/.../*
- IAMSuiteAgent:/Nexaweb
- IAMSuiteAgent:/Nexaweb/.../*
- IAMSuiteAgent:/jmx-config-lifecycle
- IAMSuiteAgent:/jmx-config-lifecycle/.../*
- IAMSuiteAgent:/SchedulerService-web
- IAMSuiteAgent:/SchedulerService-web/.../*
- IAMSuiteAgent:/sodcheck
- IAMSuiteAgent:/sodcheck/.../*

IAM Suite Authorization Policy

Figure 13–28 presents Authorization Policy in the IAM Suite application domain. There are no explicit constraints or responses. Use Implied Constraints is checked by default, which allows access in the absence of any authorization constraints of a particular class. There are no explicit constraints defined.

Figure 13–28 IAM Suite Authorization Policy



Authorization Policy

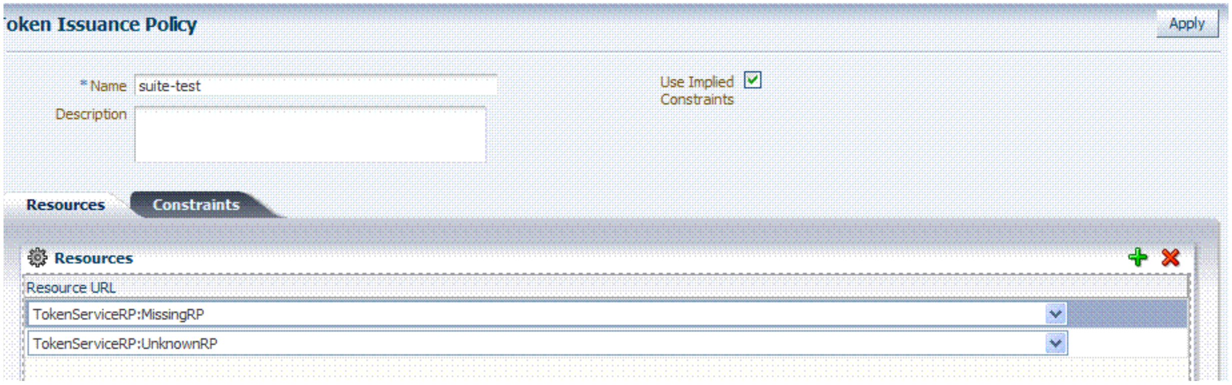
* Name	Protected Resource Policy
Description	Protected Authorization Policy for OAMAgent
Success URL	
Failure URL	
Use Implied Constraints	☑
Identity Assertion	□

Resource URL
IAMSuiteAgent:/oamconsole
IAMSuiteAgent:/oamconsole/…/*
IAMSuiteAgent:/console
IAMSuiteAgent:/console/…/*
IAMSuiteAgent:/em
IAMSuiteAgent:/em/…/*
IAMSuiteAgent:/oinav
IAMSuiteAgent:/oinav/…/*
IAMSuiteAgent:/apm
IAMSuiteAgent:/apm/…/*
IAMSuiteAgent:/oaam_admin
IAMSuiteAgent:/oaam_admin/…/*
IAMSuiteAgent:/oamTAPAuthenticate

IAM Suite Token Issuance Policy

Figure 13–29 presents IAM Suite Token Issuance Policy in the IAM Suite application domain. Use Implied Constraints is checked by default, which allows access in the absence of any authorization constraints of a particular class. There are no explicit constraints defined.

Figure 13–29 IAM Suite Token Issuance Policy and Resource URLs



oken Issuance Policy

* Name	suite-test
Description	
Use Implied Constraints	☑

Resource URL
TokenServiceRP:MissingRP
TokenServiceRP:UnknownRP

See Also: "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31 for details about the Token Issuance Policies.

14

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:

- Prerequisites
- Introduction to the OAM 11g Access Tester
- Installing and Starting the Access Tester
- Introduction to the Access Tester Console and Navigation
- Testing Connectivity and Policies from the Access Tester Console
- Creating and Managing Test Cases and Scripts
- Evaluating Scripts, Log File, and Statistics

14.1 Prerequisites

Before you can perform tasks in this chapter:

- Ensure that the Oracle Access Manager Console and OAM Server are running.
- Confirm the application domain and policies for one or more resources, as described in Chapter 13.

14.2 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 having a network connection to the OAM Server. Both a graphical user interface (known as the Tester 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
- Stress test the policy server by running multiple concurrent tests (multi-threaded mode) with command-line mode only.
- 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.

Additionally, the Access Tester provides a multi-threaded capability designed to stress test the policy server. In the multi-threaded approach, you identify the number of virtual test clients to connect to the policy server and the number of iterations that each virtual client should execute a test script. This enables you to stress test the policy server.

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

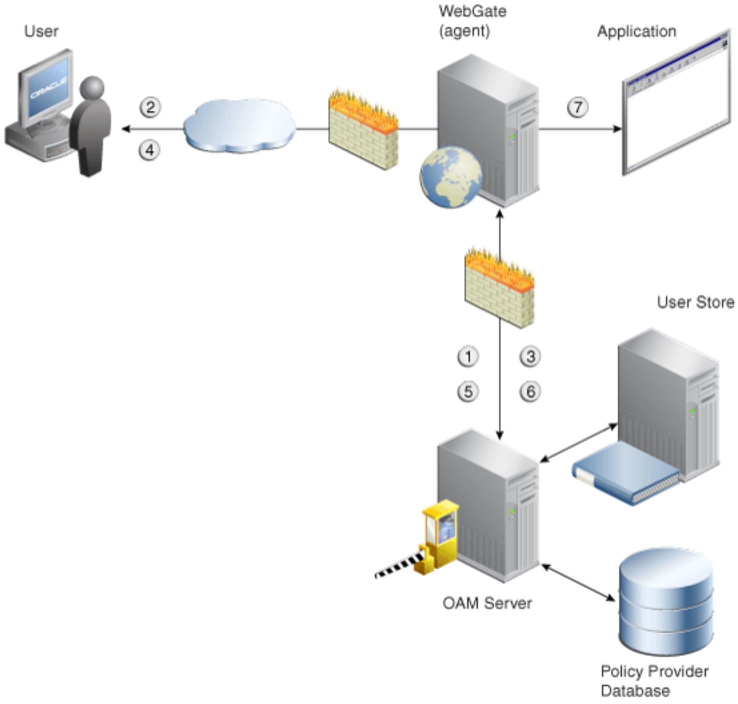
14.2.1 About OAM Agent and Server Interoperability

The two primary types of actors in the OAM architecture are the policy servers (OAM Servers) and OAM policy enforcement agents (Webgates or Access Clients). 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 inter-operability and the key decision points, Figure 14–1 illustrates a typical OAM Agent and OAM Server interaction during a user's request for a resource.

Figure 14–1 OAM Agent (PEP) and OAM Server (PDP) Inter-operability



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.

14.2.2 About Access Tester Security and Processing

This topic provides information about secure communications, connections, storage, input, iogging, and Analysis.

Secure Communication: The Access Tester supports Open, Simple, or Cert connection modes for communication with the OAM Server:

- Open mode: No security on the physical connection
- Simple mode: The physical connection is encrypted using built-in certificates. With Simple mode, you are asked to enter the Global Pass Phrase that is configured for the OAM Server.
- Cert mode: The physical connection is encrypted using a field-provided certificates. Access Tester Cert Mode requires:
  - Configuring the agent (either existing or new) for Cert mode communication.
  - Obtaining certificates for the agent being emulated.

Access Tester Cert Mode requires two JKS key stores, created using the importcert tool from the supplied PEM (BASE64-encoded ASCII) certificates: aaa_trust.pem, aaa_key.pem, aaa_cert.pem:

- A Trust Store (file containing the JKS key store with the root CA certificate) is required.
- A Key Store (file containint the JKS key store with the agent's private key and certificate) is required.
- A Key Store Password is used to encrypt the Key Store with the agent certificates.

See Also:

- Appendix E, "Securing Communication for Oracle Access Manager 11g" for details about Simple and Cert mode configuration for OAM Server and clients (Webgates)
- "Introduction to the Access Tester Console and Navigation" on page 14-12

Connections: The Access Tester encrypts all password-type values that it saves to configuration files and test cases. Access Tester validates whether the pool contains valid connections. Cache flush requests are sent over an established connection (not an out-of-band connection to delete the user session (to simulate logout) over OAP. Using an already established connection can improve performance.

Persistent Storage: 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 Access Tester uses a single XML schema to define all the XML documents it generates. 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 14-32.
- 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 14-33.
- 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 14-34. 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 14-36.
- Execution Log: lamtest_log.log is generated together with the output script. For details, see "About the Execution Log" on page 14-38.

For more information, see "About Access Tester Modes and Administrator Interactions".

14.2.3 About Access Tester Modes and Administrator Interactions

This topic describes modes, interactions, and the jar files needed to start and run the Access Tester.

Console: 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.

Command Line and Scripts: You can use the Access Tester command line and develop test scripts, which you can run interactively or in batches for computerized execution to maximize productivity and minimize costs and resources.

Startup and Run Time JAR Files: The Access Tester requires nap-api.jar in the same directory as the main jar oamtest.jar, which is used to start the application.

Interactions: 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 14–2 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:** Command-line mode enables complete automation of test script execution in single or multi-client mode environments. The Access Tester exposes a control mechanism to configure test runs without having to change "known good" input test scripts which are available in read-only mode.

Figure 14–2 User Interactions with the Access Tester

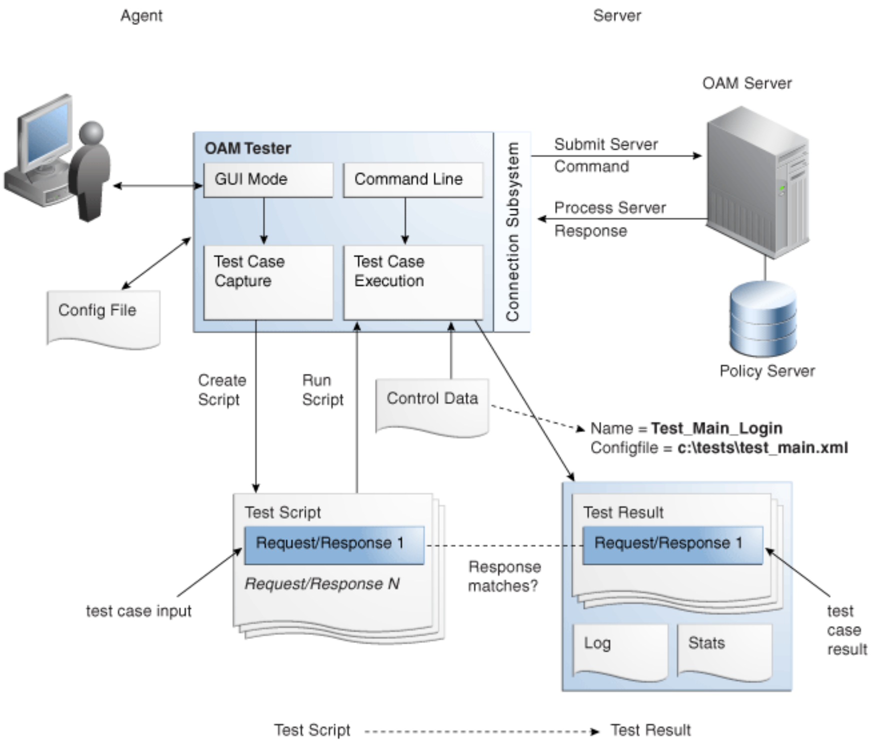


Table 14–1 describes the process flow of information during both Tester Console mode operations and command-line mode operations.

Table 14–1 User Interactions Using Tester Console Mode versus Command Line Mode Operations

Tester 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. Cert mode for secure communication: The keystores are specified in the OamTestConfiguration.xml file containing previously saved configuration information.
The user opens a previously saved OamTestConfiguration.xml file to populate the application fields and minimize data entry, including server connection fields. **Alternatively**, the user can use the Tester 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.
In Cert mode, you will be prompted to identify the necessary keystores.	In Cert mode, the keystores are specified in the XML file containing previously saved configuration information.

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"

14.3 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: Tester 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 Tester Without System Properties For Use in Tester Console Mode
- Starting the Access Tester with System Properties For Use in Command Line Mode

14.3.1 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. Cert Mode: If the OAM Server communication mode is Cert, ensure that the computer from which you will run the Access Tester includes the same keystores

that are defined on the agent registration page of the Oracle Access Manager Console. See Chapter 9.

5. Proceed as follows, depending on your environment and requirements:

   - Starting the Tester Without System Properties For Use in Tester 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"

14.3.2 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 14–2, which describes all supported system properties.

Table 14–2 Access Tester Supported System Properties

Property	Access Tester Mode	Description and Command Syntax
log.traceconnfile	Tester Console and Command Line modes	Logs connection details to the specified file name. -Dlog.traceconnfile="<file-name>"
display.fontname	Tester 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	Tester 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	Tester 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>"

Table 14–2 (Cont.) Access Tester Supported System Properties

Property	Access Tester Mode	Description and Command Syntax
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>"
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.displayiterationstats	Command Line mode	Controls whether or not to display intermediate statistics after each iteration of the test run. -Dcontrol.displayiterationstats=”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"

14.3.3 Starting the Tester Without System Properties For Use in Tester Console Mode

To manually drive (and capture) requests and view real-time response through the graphical user interface, start the tester in Tester Console mode. This procedure omits all system properties, even though several can be used with Tester 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. Use the -help option to list all the options available for the oamtest command-line tool.

   ```
   java -jar oamtest.jar -help
   ```

3. 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

14.3.4 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 Tester Without System Properties For Use in Tester Console Mode

14.3.4.1 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 14-9

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.

14.3.4.2 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 14-9

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:

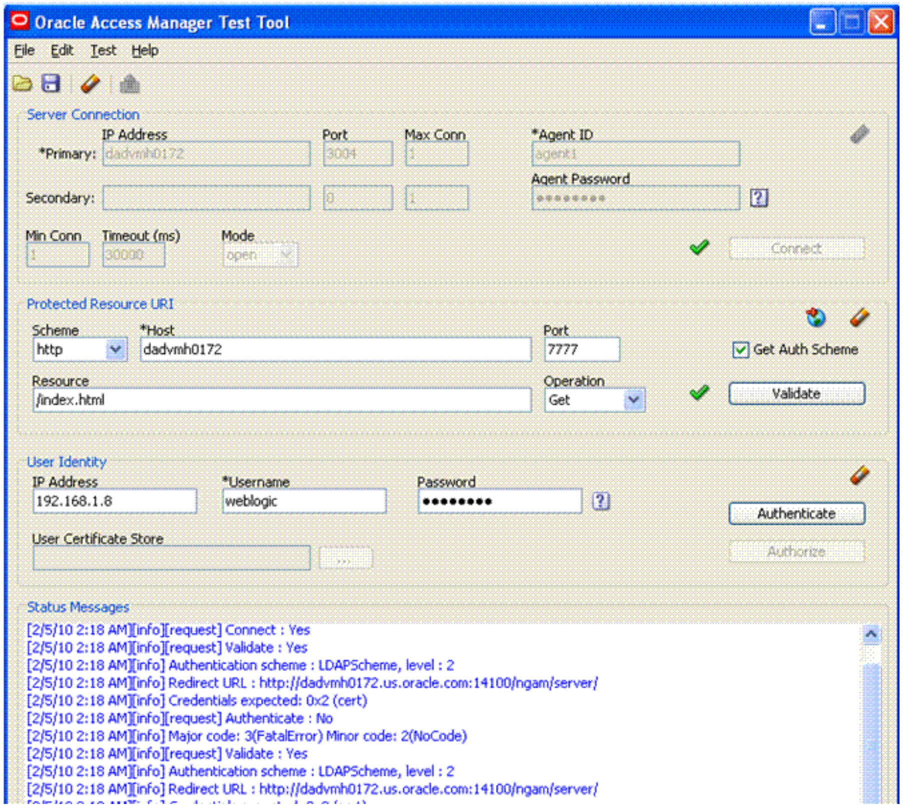
   - Testing Connectivity and Policies from the Access Tester Console
   - Creating and Managing Test Cases and Scripts

14.4 Introduction to the Access Tester Console and Navigation

This section introduces the Access Tester Console, navigation, and controls.

Figure 14–3 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 14–3 Access Tester Console



*Primary: IP Address: dadvmh0172, Port: 3004, Max Conn: 1

Secondary: IP Address: (empty), Port: 0, Max Conn: 1

*Agent ID: agent1

Agent Password: ●●●●●●●●

Min Conn: 1

Timeout (ms): 30000

Mode: open

Scheme: http

*Host: dadvmh0172

Port: 7777

☑ Get Auth Scheme

Resource: /index.html

Operation: Get

IP Address: 192.168.1.8

*Username: weblogic

Password: ●●●●●●●●●

User Certificate Store: (empty)

```
[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 14–3.

Table 14–3 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 14-16.
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 14-18.
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 14-20.
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 14-23.

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 14–4.

Table 14–4 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"

14.4.1 Access Tester Menus and Command Buttons

Table 14–5 identifies additional Access Tester Console buttons and their use. All command buttons provide a tip when the cursor is on the button.

Table 14–5 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.

Table 14–5 (Cont.) Additional Access Tester Buttons

Command Buttons	Description
💾	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.
⇧ ⇧	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 14–6. 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 14–6 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

Table 14–6 (Cont.) Access Tester Menus

Menu Title	Menu Commands
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	The command About, which displays usage information.

14.5 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 14-24.

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 14-8.
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 14-16.
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 14-18.
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 14-20.
5. After successful authentication, click Authorize in the User Identity panel, as described in "Testing User Authorization from the Access Tester Console" on page 14-22.
6. Check the latency of requests, as described in "Observing Request Latency" on page 14-23.

14.5.1 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

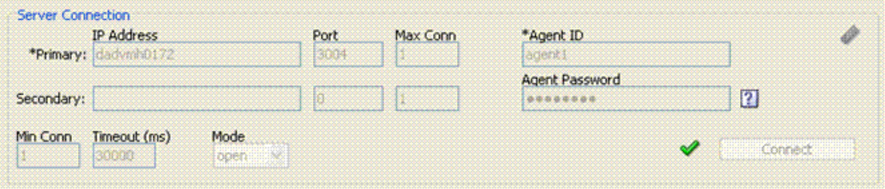
14.5.1.1 About the Connection Panel

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 14–4 illustrates the Server Connection panel and controls. This panel contains information needed to establish a connection to the OAM Server's Proxy port.

Figure 14–4 Server Connection Panel in the Access Tester



Server Connection

	IP Address	Port	Max Conn	*Agent ID
*Primary:	dadvmh0172	3004	1	agent1
				Agent Password
Secondary:		0	1	●●●●●●●●

Min Conn	Timeout (ms)	Mode
1	30000	open

Table 14–7 describes the information needed to establish the connection. The source of your values is the Oracle Access Manager Console, System Configuration tab.

Table 14–7 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.

Table 14–7 (Cont.) Connection Panel Information

Fields	Description
Mode	The level of communication security that is designated for the Agent to be emulated. ▪ Open--No special configuration needed for this mode. ▪ Simple--Presents a field for the global pass phrase set for the OAM Server. See Also: "Retrieving the Global Passphrase for Simple Mode" on page E-13. ▪ Cert--Presents a Configure Certs ... button that opens a dialog asking for the following: Trust Store (Root Store Alias): A file containing the JKS key store with the root CA certificate. Key Store: A file containing the JKS key store with the agent's private key and certificate. Currently, the agent certificate is used for encrypting the connection and not the agent identification. Key Store Password: The password used to encrypt the Key Store with the agent certificates. See Also: "About Access Tester Security and Processing" on page 14-4, and "Generating Client Keystores for OAM Tester in Cert Mode" on page E-5.
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.
?	Click ? beside the Agent Password field for help.
✓	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.
⊗	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"

14.5.1.2 Connecting the Access Tester with the OAM Server

Use the following procedure to submit your connection details for the OAM Server.

> **Note:** Cert mode requires the presense of keystores generated as described in Appendix E, "Securing Communication for Oracle Access Manager 11g"

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. Start the Access Tester, as described in "Installing and Starting the Access Tester" on page 14-8.

2. In the Server Connection Panel (Table 14–7), enter:
   - Primary and secondary OAM Proxy details
   - Timeout period
   - Communication encryption mode
   - Agent details
3. Click the Connect button.
4. Beside the Connect button, look for the green check mark indicating the connection is established.
5. In the Status Messages panel, verify a Yes response.

   Not Successful: If there is a problem connecting to the OAM Server, ensure that you entered all connection information correctly (IP address and port, Agent name and password, connection mode and related certificates and passwords, as needed).

   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 Administrator of the OAM Server to review the policy server log.

14.5.2 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

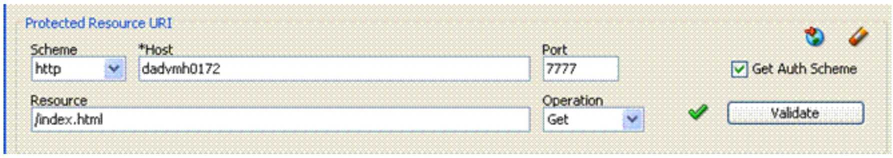
14.5.2.1 About the Protected Resource URI Panel

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 14–5 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 14–5 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 14–8 describes the information needed to perform this validation.

Table 14–8 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 Oracle Access Manager 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 Oracle Access Manager Console. **Note**: If protected, the resource identifier that you provide here must match the one specified in an authorization policy in the Oracle Access Manager 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.
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 14-24

14.5.2.2 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 14–8).
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 14-24.
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"

14.5.3 Testing User Authentication from the Access Tester Console

This topic provides the following information:

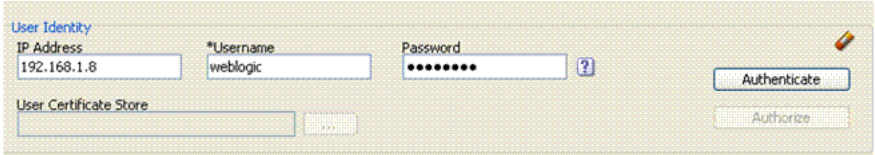
- About the User Identity Panel
- Testing User Credential Authentication

14.5.3.1 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 14–6 illustrates the Access Tester panel where you enter the information needed to test authentication.

Figure 14–6 Access Tester User Identity Panel



User Identity

IP Address: 192.168.1.8

*Username: weblogic

Password: ●●●●●●●●

User Certificate Store:

Table 14–9 describes the information you must provide.

Table 14–9 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 username and password fields if the resource is protected by an authentication scheme that requires those credentials. Similarly the Access Tester enables or disables the certificate field if the resource is protected by an authentication scheme that requires a user's X509 certificate.
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 PEM format file containing the X.509 certificate of the user whose credentials should be authenticated. If the URI is protected by the X509 Authentication Scheme then the Tester will use the PEM-formatted X509 certificate as a credential instead of or in addition to the username/password. The X509 cert may also be used for authorization if security policies are so configured on the OAM Server. 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.
...	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. Note: For certificate-based authentication, the OAM Server deployment must be properly configured with certificates as described in Appendix E.
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. This request submits information collected in the URI and Identity panels to the OAM Server to decide if the user defined on the Identity panel can access the resource defined on the URI panel. The server returns Yes (user can access the resource) or No (user can not access the resource). The OAM Server might return additional information such as actions (responses) that the real Agent would normally handle.

Table 14–9 (Cont.) Access Tester User Identity Panel Fields and Controls

Field or Control	Description
✓	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 14-24

14.5.3.2 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 14–9).
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 Oracle Access Manager Console for an active user session that you might need to end, as described in Chapter 7.
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 14-24.
5. Retain the URI and user identity information and proceed to "Testing User Authorization from the Access Tester Console".

14.5.4 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 14–9).
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 Oracle Access Manager 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 14-24.
6. Proceed to:
   - Observing Request Latency
   - Creating and Managing Test Cases and Scripts
   - Evaluating Scripts, Log File, and Statistics

14.5.5 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 14-8

Task overview: Observing request latency includes

1. "Validating Resource Protection" on page 14-20
2. "Testing User Authentication from the Access Tester Console" on page 14-20
3. "Testing User Authorization from the Access Tester Console" on page 14-22
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

14.6 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
- Capturing Test Cases
- Generating an Input Test Script
- Personalizing an Input Test Script
- Executing a Test Script

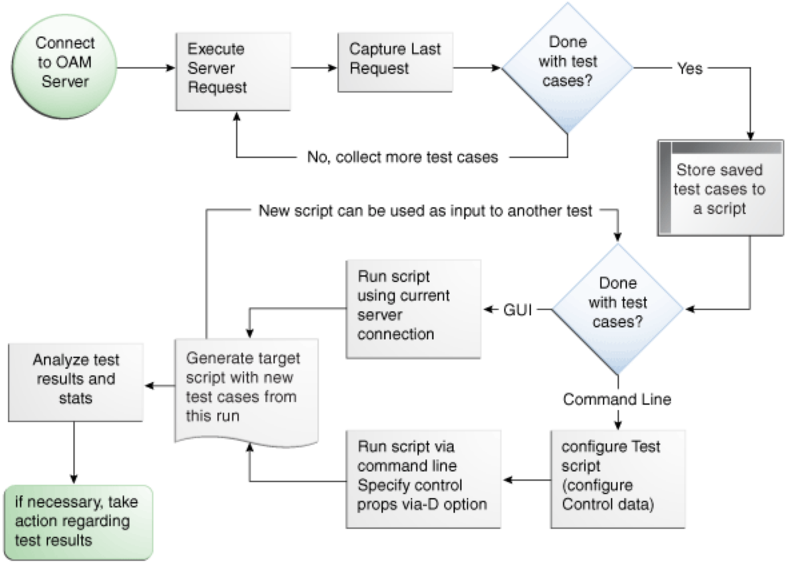
14.6.1 About Test Cases and Test Scripts

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 14–7.

Figure 14–7 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
Done with test cases?
GUI
Run script using current server connection
Generate target script with new test cases from this run
Analyze test results and stats
Command Line
configure Test script (configure Control data)
Run script via command line Specify control props via-D option
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.

14.6.2 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 14–10 describes the location of the capture options.

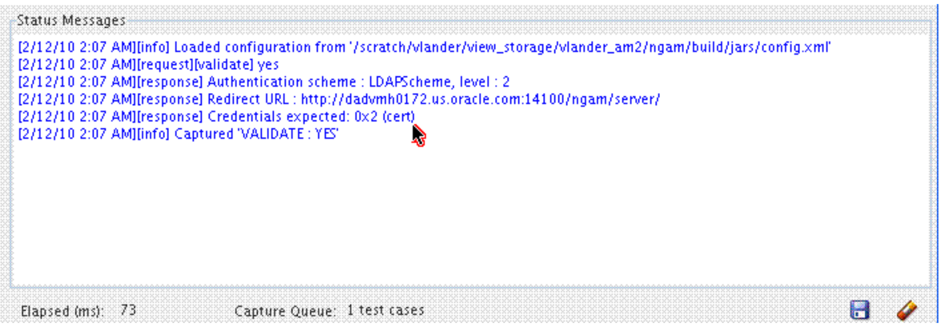
Table 14–10 Access Tester Capture Request Options

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 14-15.
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".

14.6.3 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.

14.6.3.1 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 14–11 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.

14.6.3.2 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 14-25.
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.

14.6.4 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

14.6.4.1 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 14–12 describes the control elements and how to customize these.

Table 14–12 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="1"	Indicates the number of threads (virtual clients) that will be started by the Access Tester to run multiple copies of the test script. Each thread opens its own pool of connections to the policy server. This feature is designed for stress testing the Policy Server, and is available only in command line mode. Default: 1 Note that when running a test script in GUI mode, the number of threads is ignored and only one thread is started to perform a single iteration of the test script.
numiterations="1"	Indicates the number of iterations that will be performed by the Access Tester. This feature is designed for stress testing and longevity testing the Policy Server and is available only in command line mode. Default: 1

14.6.4.2 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".

14.6.5 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

14.6.5.1 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 14–13 describes the commands to execute a test script.

Table 14–13 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. When running in multi-threaded mode, only the statistics XML file and execution log file will be generated.

   e. In command line mode, the Access Tester exits with the exit code as described in "About the Access Tester Command Line Mode" on page 14-11.

14.6.5.2 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 14-26.
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 14-11.

     ```
     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".

14.7 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

14.7.1 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

> **Note:** The target script is not created if the Access Tester is configured to run in multi-threaded mode.

- 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 14-33.

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 14–14.

Table 14–14 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.

14.7.2 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 14–1 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>
        <certstore></certstore>
        <ipaddr>111.222.3.4</ipaddr>
    </identity>
</oamtestconfig>
```

14.7.3 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 14–2 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>
                    <certstore></certstore>
                    <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>
                    <certstore></certstore>
                    <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>
```

14.7.4 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 14–3. As shown in the execution log, this test run found no mismatches, and shows that 4 out of 4 requests matched.

Example 14–3 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>
                    <certstore></certstore>
                    <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>
                    <certstore></certstore>
                    <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>
```

14.7.5 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 14–4. The various sections that provide statistics for this run, which you can compare against statistics for an earlier "known good" run.

Example 14–4 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>
```

14.7.6 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 14–5 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'
```

15

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 IAMSuiteAgent
- Configuring Centralized Logout for 10g Webgate with OAM 11g Servers
- Configuring Centralized Logout for Oracle ADF-Coded Applications
- Removing Custom mod_osso Cookies on Logout
- Validating Global Sign-On and Centralized Logout

> **Caution:** Oracle recommends using the logout mechanism provided by Oracle Access Manager, not custom logout scripts.

15.1 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
  - IAMSuiteAgent 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

15.2 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 15–1 describes the circumstances under which centralized logout occurs.

Table 15–1 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 IAMSuiteAgent
- About Centralized Logout with OSSO Agents (mod_OSSO) and OAM 11g

- About Centralized Logout for Applications Using Oracle ADF Security

15.2.1 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".

15.2.2 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".

15.2.3 About Centralized Logout with the IAMSuiteAgent

The IAMSuiteAgent is a domain-wide agent that provides single sign-on functionality for the IDM Administration Console. The IAMSuiteAgent 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 IAMSuiteAgent" on page 15-6.

15.2.4 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.
5. Delete the custom mod_osso agent cookies on logout.

15.2.5 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 15-12.

15.3 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

15.3.1 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 15–2.

Table 15–2 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.

Table 15–2 (Cont.) Logout Elements in OAM 11g Webgate Registration

Element	Description
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.
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 logged out 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 logged out cookie.

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

15.3.2 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 9, "Registering Partners (Agents and Applications) by Using the Console"
   - Chapter 10, "Registering Partners (Agents and Applications) Remotely"
2. When creating or editing an agent registration, include appropriate logout values for your environment (Table 15–2):

   > **Note:** If the `LogOutUrl` parameter is already configured for the 11g Webgate (with a value other than "`/oamsso/logout.html`"), then ensure that "`/oamsso/logout.html`" is also present as part of the `LogOutUrl` parameter.

   - 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 15-16.

15.4 Configuring Centralized Logout for the IAMSuiteAgent

The IAMSuiteAgent is pre-configured with the logout parameters needed to perform central logout against the OAM 11g Server. While similar to a 10g Webgate, the IAMSuiteAgent does not have a local logout.html page to be configured. Instead, the IAMSuiteAgent is delivered with a pre-deployed application oamsso_logout), that is used by the agent to perform the logout.

The logout functionality for the IAMSuiteAgent requires that the oamsso_logout application is deployed in the Server where the IAMSuiteAgent 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 IAMSuiteAgent.

To configure logout for the IAMSuiteAgent

1. Log in to the WebLogic Server Administration Console.
2. Navigate to Domain, Deployments, oamsso_logout, Targets.
3. Select all the Servers where the IAMSuiteAgent is enabled and where logout is performed. For example, oim_server, oaam_admin, oaam_server, and so on.
4. Click Save.

15.5 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

15.5.1 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 Creating and Editing Webgate Registration" on page 9-11.

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".

15.5.2 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 15-4.

Example 15–1 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 15–1 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 15-12:
>
> ```
> /<app context root>/adfAuthentication?logout=true&end_url=<any uri>
> ```

Table 15–3 illustrates how a logout link in the logout.html file might be specified:

Table 15–3 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/welcome.html`

15.5.3 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 27, "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 15–1, "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 `LogOutUrl` 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 `LogOutUrl`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 15-10.

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>
```

15.6 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 IAMSuiteAgent 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 15-13.
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

15.6.1 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 15–2.

   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.

15.6.2 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 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 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 Administrator. Here, `logouturi` value is the URI of the logout script /logout.html. The value could either begin with "logout." (exceptions are logout.gif and logout.jpg) or it could be any other value configured by the 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 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 15-6.

5. **OAM 10g Webgate**: Perform steps in "Configuring Centralized Logout for 10g Webgates with OAM 11g" on page 15-10.

   > **See Also:** "Scenario: Identity Propagation with the OAM Token" on page 16-2 for details about setting up providers for Oracle Access Manager 11g Identity Assertion.

15.7 Removing Custom mod_osso Cookies on Logout

On user logout, some custom cookies set by OAM Server through authentication response settings might not get deleted. However, you can edit oam-config.xml to configure the OAM Server to delete custom cookies set during authentication when a user logs out of OAM. For instance, when integrating with Oracle E-Business Suite, the ORASSO_AUTH_HINT cookie is set by the application and should be included in the `CookieNames` list (or the UCM cookie, for example).

Syntax (beneath `PluginClass" Type=`...):

```
<Setting Name="CookieDelMap" Type="htf:map">
  <Setting Name="CookieNames" Type="xsd:string">COOKIE_NAME</Setting>
</Setting>
```

The following procedure guides as you edit the CookieDelMap element and add CookieNames as a single value or a comma-separated list of custom cookies to delete when a user logs out. This procedure also explains how to increment the oam-config.xml file version to propagate your change to all managed servers without restarting.

> **Caution:** Work carefully. In general, Oracle recommends that you do not edit the oam-config.xml file. This, however, is a rare exception.

To delete custom mod_osso cookies on logout

1. Back up *DOMAIN_HOME*/config/fmwconfig/oam-config.xml.
2. In oam-config.xml, add (or edit) the `CookieDelMap` element and `CookieNames`. For example:

```
<Setting Name="ResponsePluginSetting" Type="htf:map">
  <Setting Name="PluginClass" Type=... </Settings>
  <Setting Name="CookieDelMap" Type="htf:map">
    <Setting Name="CookieNames" Type="xsd:string">ORASSO_AUTH_HINT
    </Setting>
  </Setting>
</Setting>
```

3. **Configuration Version:** Increment the `Version xsd:integer` as shown in the next to last line of this example (existing value (25, here) + 1):

   Example:

```
<Setting Name="Version" Type="xsd:integer">
  <Setting xmlns="http://www.w3.org/2001/XMLSchema"
    Name="NGAMConfiguration" Type="htf:map:>
  <Setting Name="ProductRelease" Type="xsd:string">11.1.1.3</Setting>
    <Setting Name="Version" Type="xsd:integer">25</Setting>
</Setting>
```

4. Save the file.

15.8 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

15.8.1 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.

15.8.2 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.

15.8.3 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 V

Oracle Security Token Service

This section provides information to help administrators manage the Oracle Security Token Services available with Oracle Access Manager.

Part V contains the following chapters:

- Chapter 16, "Oracle Security Token Service Implementation Scenarios"
- Chapter 17, "Managing Oracle Security Token Service Settings and Set Up"
- Chapter 18, "Managing Oracle Security Token Service Certificates and Keys"
- Chapter 19, "Managing Templates, Endpoints, and Policies"
- Chapter 20, "Managing Token Service Partners and Partner Profiles"
- Chapter 21, "Troubleshooting Oracle Security Token Services"

16

Oracle Security Token Service Implementation Scenarios

This chapter introduces several Oracle Security Token Service implementation and processing scenarios. Regardless of scenario specifics, there are many similarities in both configuration tasks and token handling. This chapter provides the following sections:

- Prerequisites
- Typical Token Ecosystem
- Scenario: Identity Propagation with the OAM Token
- Scenario: Web Service Security With On Behalf Of Username Token

16.1 Prerequisites

"Introduction to Oracle Security Token Service" on page 1-10.

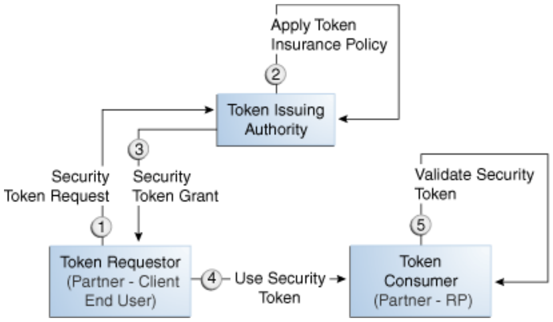
16.2 Typical Token Ecosystem

The abstract model chosen here is of interest because of the requirements placed on Oracle Security Token Service to support such models.

The phrase security token ecosystem is used here to represent a typical environment where security tokens are in use. In such environments the security token, based on the security model required for the environment, could be used to serve an end goal such as to enable brokered trust or single-sign-on and so on. Regardless of the environment and the type of security token, several aspects are common across all models, as shown and described here.

Figure 16–1 illustrates a typical token ecosystem, which includes: Token Issuing Authority, Token Requestor, Token Consumer, and the Security Token.

Figure 16–1 Typical Token Ecosystem



Actors and Process overview: In a typical token ecosystem

1. The Token Requestor places a request for a security token at the Token Issuing Authority.

   This security token is required to communicate and request access to a service provided by a Service Provider (a Token Consumer who accepts the security token).

   - A Token Requestor could be a Partner of the Token Issuing Authority (generally registered with the Token Issuing Authority).
   - A Token Requestor could be an End User (generally not registered with the Token Issuing Authority).

2. The Token Issuing Authority (OAM and OSTS, for example) receives and processes the security token request and returns a security token, as follows:

   - Authenticate the input credentials.
   - Authorize the security token request based on a Token Issuance Policy that specifies which Token Requestors are authorized to request a security token for a given Token Consumer.

3. The Token Consumer (typically a service provider).

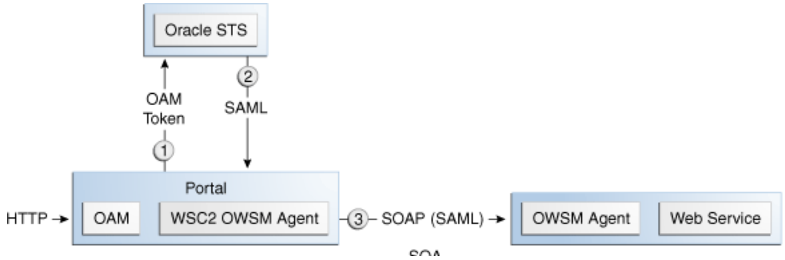
   - Accepts the security token as part of the service request and provides service based on the validity of the input security token.
   - Validates the input security token with Token Issuing Authority.

   Note: A Token Consumer is typically a registered Partner of the Token Issuing Authority. A Token Consumer is also known as a Relying Party, because it trusts and relies on the Token Issuing Authority for Token Requestor authentication. Token Consumers (Relying Party Partner) are Web Applications (for OAM, OSTS is the Token Issuing Authority) or STS Relying Party Web Services.

16.3 Scenario: Identity Propagation with the OAM Token

This is a deployment where the user's Identity information needs to be propagated from a Web application to a Web service provider. The Web service provider can reside in the same security domain as the web application or in a different security domain.

Figure 16–2 Identity Propagation with the OAM Token



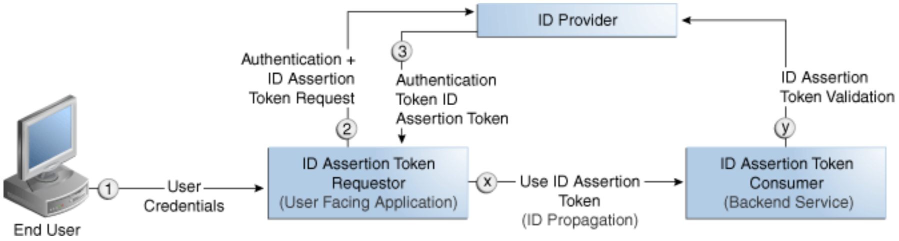
Identity propagation means that the original user context becomes visible outside its original security tier or domain boundaries. The user security context is propagated across different security tiers or domains to support tier-specific or domain-specific security needs such as step-up authentication, authorization, audit and/or internal application-specific business logic.

ID Propagation is said to occur in a distributed processing of a request when the identity context established in the first node is propagated to subsequent nodes to enable further processing of the request in the context of that identity.

ID Propagation can be achieved in several ways. One of them is based on a brokered-trust model where an ID provider acts as a trust-broker for ID Assertions. The discussion here is pertains to this model.

Figure 16–3 illustrates an ID Propagation scenario in a brokered-trust model, where a user-facing application needs to request processing by a backend service application in the context of the end user. To bring out the main aspects of ID propagation all other interaction and relationship details between end user, application, and backend service application are ignored.

Figure 16–3 Process Flow During Identity Propagation


ID Provider
Authentication + ID Assertion Token Request
2
3
Authentication Token ID Assertion Token
ID Assertion Token Validation
y
End User
1
User Credentials
ID Assertion Token Requestor (User Facing Application)
x
Use ID Assertion Token (ID Propagation)
ID Assertion Token Consumer (Backend Service)

Actors and Process overview: Identity Propagation

1. The ID Assertion Token Requestor (an End User-Facing Application), upon end user access, requests authentication and ID Assertion Token at the identity Provider.

> **Note:** Examples of ID Assertion Token Requestors include Web applications that are protected by OAM. The ID Assertion Token request could be either implicit or could be driven by a policy at the ID Provider.

2. The ID Provider (Oracle Security Token Service) processes the request and returns an Authentication Token and an ID Assertion Token. An ID Assertion Token, in

itself, does not represent a user session and cannot be used independently to request direct access to a resource or service.

3. The ID Assertion Token Requestor uses this Token later, during the end user session, as part of a backend service processing request (on behalf of the end user).
4. The ID Assertion Token Consumer (Oracle Security Token Service), as part of the request processing, first validates the ID Assertion Token and then (on validation success) processes the request in the context of the end user Identity

For more information, see the following topics:

- Component Processing: Identity Propagation with the OAM Token
- RST Attributes and Run Time Processing
- Configuration Requirements: Identity Propagation with the OAM Token

16.3.1 Component Processing: Identity Propagation with the OAM Token

Figure 16–4 illustrates a typical deployment topology for Identity propagation using Oracle Security Token Service with Oracle Access Manager.

Figure 16–4 Identity Propagation Deployment

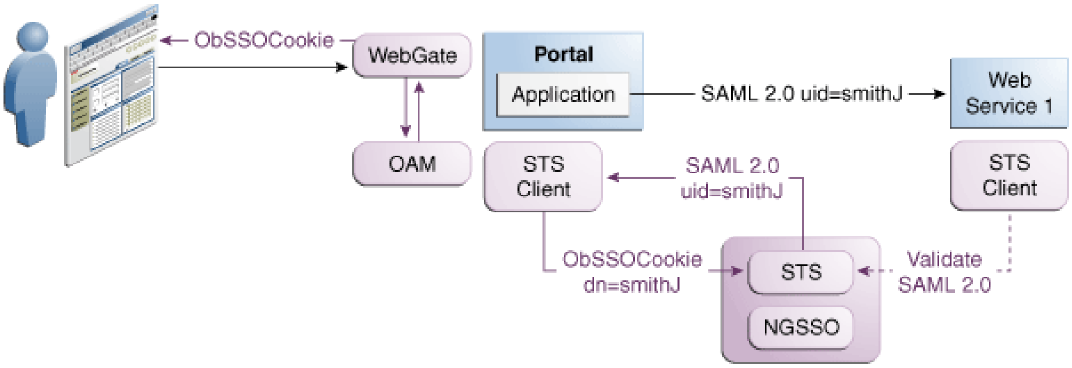
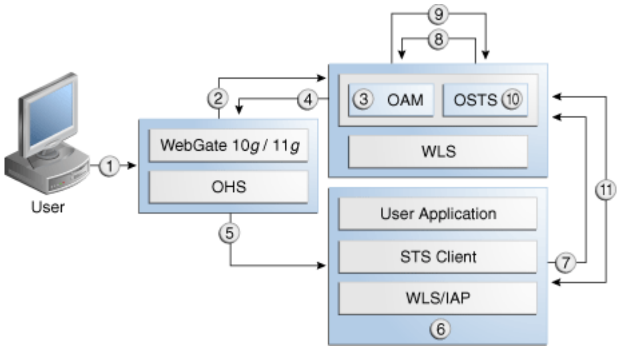


Figure 16–5 illustrates a processing of Identity propagation using Oracle Security Token Service with Oracle Access Manager. Details follow the figure.

Figure 16–5 Identity Propagation Processing



Process overview: Component interactions for Identity Propagation

1. User attempts to access a protected resource.
2. Webgate is protecting the resource; it sends request to Oracle Access Manager for authentication and authorization.
3. Oracle Access Manager authenticates the user using the policy configured for this Webgate Application Domain. It sees the response type "IDENTITY_ASSERTION" is configured for this Webgate, so it generates ID Assertion token as well.
4. Oracle Access Manager sends authentication and ID assertion token to Webgate
5. Webgate processes the response; sets ID assertion token to the header; (OHS where Webgate is installed on) then redirect the request to the WLS that hosts the resource.
6. IAP (Oracle Access Manager Identity Asserter on WebLogic Server) sees OAM_ IDENTITY_ASSERTION header is set, processes the headers, then sets ID assertion token to Subject's private credential as `OamIdentity`.
7. When the resource is finally accessed, a Web Service Client can then obtain the ID assertion token from current user's Subject, generates a OnBehalfOf (OBO) token with it, then creates and sends Request Security Token (RST) to OSTS.
8. Oracle Security Token Service sees the ID assertion Token inside OBO token, it sends validation/authentication request to Oracle Access Manager using Oracle Access Manager library.
9. Oracle Access Manager validates and authenticates the ID Assertion Token, then sends response (user identity) to Oracle Security Token Service.
10. Oracle Security Token Service uses this user identity to do further processing: policy evaluation, token issuance, and so on. It then generates Request Security Token Response.
11. Oracle Security Token Service sends Request Security Token Response to the client, which can then use the token inside the Request Security Token Response (RSTR) to create a web service request to access a service hosted on a relying party.

16.3.2 RST Attributes and Run Time Processing

For an incoming Request Security Token (RST) with the following attributes, Oracle Security Token Service must be configured to process a request and issue a token):

RST Attributes for Identity Propagation with the OAM Token

- The SOAP header contains a Username token referencing a WS Requester. The Username token contains at least a username and a password.
- The SOAP body contains a WS-Trust RST message
- The RST contains a OAM ID Propagation token in the OnBehalfOf field referencing a user in LDAP. The token included in the OnBelahfOf element is a BinarySecrityToken, whose text value is the Base 64 encoded format of the OAM Session Propagation Token, and whose ValueType attribute is `http://xmlns.oracle.com/am/2010/11/token/session-propagation` and whose EncodingType attribute is `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soapmessage-security-1.0#Base64Binary`
- The RST can possibly contain an AppliesTo field holding a URL pointing to the endpoint of the Relying Party Web Service

- The RST can possibly contain a TokenType field holding the type of token that needs to be returned
- The RST can possibly contain an Entropy field holding random data that will be used when creating the SecretKey when a symmetric proof key is required in the SAML Assertion
- The RST can possibly contain a UseKey field holding the certificate or public key to be used as an asymmetric proof key in the SAML Assertion, but this field will be ignored by OSTS

Process overview: Identity Propagation with the OAM Token

1. Client prepares the request by:
   a. Creating the SOAP message
   b. Creating the Username token referencing the client and including it in the SOAP header
   c. Creating the WS-Trust RST message
   d. Creating the OAM ID Propagation token referencing the user and including it in the OnBehalfOf field of the RST
   e. Including the RST message in the SOAP body
2. Client sends the message to the OSTS, to an endpoint protected by a WS-Security User Name Token (UNT) Policy, with that endpoint being mapped to an OSTS WSS Validation Template.
3. OSTS will process the incoming request
4. OSTS validates the token included in the SOAP header by using the settings contained in the WS-Security Validation Template:
   a. Validates the format of the Username token
   b. Validates the credentials contained in the Username token against the OSTS Partner store, thus mapping this token to a Requester Partner
   c. Knowing the Requester Partner, OSTS will retrieve the Requester Partner Profile associated with this Requester
5. OSTS then validates the token present in the OnBehalfOf field:
   a. Determines the type of token present in the OnBehalfOf field
   b. Retrieves the WS-Trust Validation Template to be used for OAM Token Type, from the Requester Partner Profile
   c. Validates the format of the OAM token
   d. Validates the OAM token, and maps the token to a user
   e. Creating the OAM ID Propagation token referencing the user and including it in the OnBehalfOf field of the RST
6. OSTS then examines the AppliesTo field:
   - If present, OSTS will attempt to map the AppliesTo URL to a Relying Party Partner, using the WS Endpoint Mapping of the Relying Party Partner. If the mapping is successful, then the AppliesTo field has been mapped to a Relying Party Partner, and OSTS will retrieve the Relying Party Partner Profile from this Partner. If mapping was not successful, then the AppliesTo field could not

be mapped to a Relying Party Partner, and OSTS will retrieve the Default Relying Party Partner Profile from the Requester Partner Profile

- If absent, OSTS will retrieve the Default Relying Party Partner Profile from the Requester Partner Profile

7. OSTS then examines the TokenType field:
   - If present, OSTS will map the TokenType string to a local token type value using the Requester Partner Profile, and it will then use the Relying Party Partner Profile to retrieve the Issuance Template to be used to create the outgoing token
   - If absent, OSTS will retrieve the default token type from the Relying Party Partner Profile, and it will then use the Relying Party Partner Profile to retrieve the Issuance Template to be used to create the outgoing token
8. OSTS will perform an Authorization evaluation to check that the Requester Partner is authorized to request a token for the Relying Party referenced in the flow (see Authorization Trust Policy for more information)
9. OSTS will then create the token
   - If the token to be issued is of SAML type, then the Issuance Template will list how to populate the NameID, the Relying Party Partner Profile will list which attributes need to be sent in the token, the Issuance Template will indicate whether or not to translate the names and values of the attributes, the Issuance Template will indicate whether or not to sign/encrypt the token.
   - If the token to be issued if of SAML type, the OSTS server will examine the KeyType to determine the Subject Confirmation Method of the Assertion. If it is missing, it will use the Default Confirmation Method from the Issuance Template
10. OSTS will create the Response that the client will process:
    a. Creates the WS-Trust RSTRC
    b. Includes the returned token
    c. Includes proof key if necessary

16.3.3 Configuration Requirements: Identity Propagation with the OAM Token

This topic walks through the configuration requirements for the identity propagation scenario. It includes:

- Configuration overview: Identity Propagation with the OAM Token
- WebLogic Server Identity Assertion Providers
- Oracle Access Manager Identity Asserter Details
- LDAP Authentication Provider Details
- Default Identity Store Configuration
- Token Issuance Policy
- Authentication Policy for Identity Assertion by Webgate
- Endpoint Configuration
- Issuance Template Configuration
- Partner Configuration: Requester

- Partner Profile: Relying Party
- Partner Profile: Requester
- Validation Template for WS-TRUST
- Cookies and Headers (Truncated)
- Request Security Token Sent By the Client (Truncated)
- Request Security Token Response sent by the OSTS Server (Truncated)

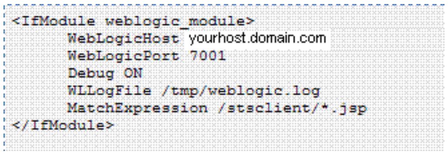
Configuration overview: Identity Propagation with the OAM Token

Following is an overview of the Identity Propagation environment and implementation tasks:

- A custom application module that will act as a client to:
  - Retrieve the OAM Session Propagation token from the HTTP request
  - Send a WS-Trust request to the Oracle Security Token Service server with OAM Session Propagation token as the OnBehalfOf element
- A web application that will be protected by Webgate and will invoke the client web application that will send a WS-Trust request to Oracle Security Token Service
- OSTS Server URL: http://*yourhost.domain.com*:14100/sts
- An OHS 11*g* with Webgate protecting the web application

  Provision (register) a Webgate (11g or 10g) to protect the application deployed in WebLogic Server. The `OAMSuite` application domain is pre-seeded and delivered with OAM 11g. When you provision an OAM Agent to use this (or another existing) application domain, decline the option of having policies automatically created.

  Reverse Proxy mapping for Webgate in the OHS Server mod_wl_ohs.conf, is shown here.



```
<IfModule weblogic_module>
      WebLogicHost yourhost.domain.com
      WebLogicPort 7001
      Debug ON
      WLLogFile /tmp/weblogic.log
      MatchExpression /stsclient/*.jsp
</IfModule>
```

The following Oracle Security Token Service configuration is required to implement token processing for identity propagation:

- One Requester Partner Profile
- One Relying Party Partner Profile
- One Issuance Template
- One WS-Trust Validation Template
- OSTS Endpoint
- An LDAP server is required for Oracle Security Token Service to map the Username token referencing the user to an LDAP User record, and thus use that record to populate the outgoing token.

  Ensure that the desired LDAP server is configured as the Default Identity Store for Oracle Access Manager with Oracle Security Token Service.

WebLogic Server Identity Assertion Providers

Deploy the Identity Assertion Providers jar. The Oracle Access Manager Identity Asserter is available in the following path with Oracle Fusion Middleware installed:

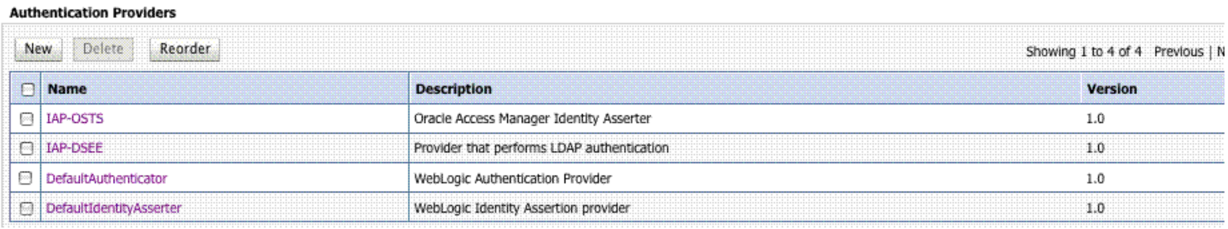
```
ORACLE_INSTANCE/modules/oracle.oamprovider_11.1.1/oamauthenticationprovider.war
```

Copy oamauthenticationprovider.war to the following location:

```
BEA_HOME/wlserver_10.x/server/lib/console-ext/autodeploy/oamauthentication
provider.war
```

Figure 16–6 illustrates the required WebLogic Server Identity Assertion Providers configuration for this scenario.

Figure 16–6 Required v1.0 WebLogic Server Identity Assertion Providers



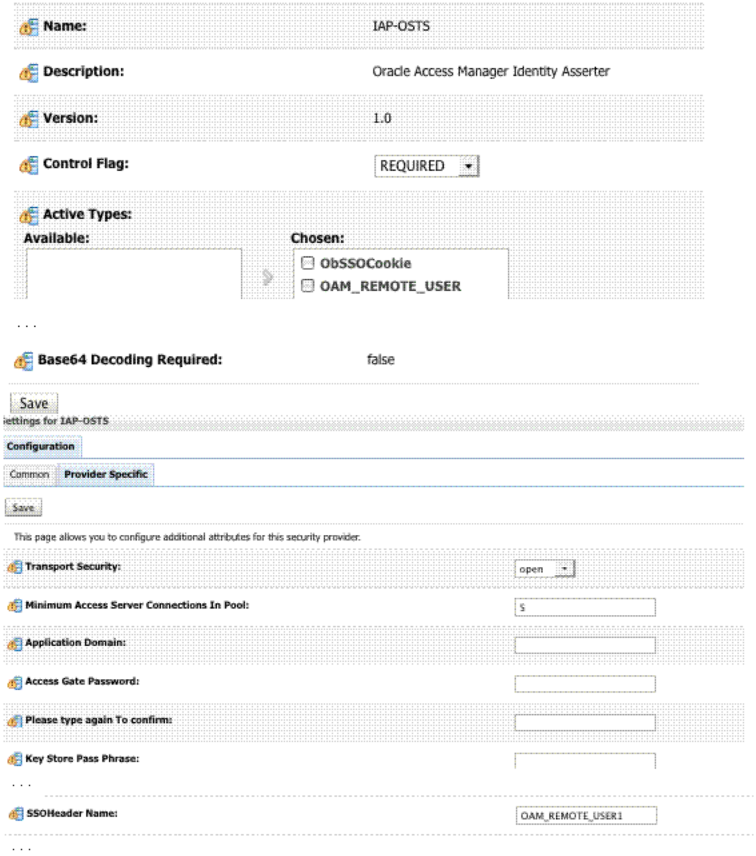
□	Name	Description	Version
□	IAP-OSTS	Oracle Access Manager Identity Asserter	1.0
□	IAP-DSEE	Provider that performs LDAP authentication	1.0
□	DefaultAuthenticator	WebLogic Authentication Provider	1.0
□	DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0

Oracle Access Manager Identity Asserter Details

The IAP-OSTS identity asserter must be the first, and set using the REQUIRED Control flag. The Active Types should be set as ObSSOCookie and OAM_REMOTE_USER, with an SSO Header name of OAM_REMOTE_USER1.

Figure 16–7 illustrates the configuration.

Figure 16–7 IAP-OSTS Details



Name:	IAP-OSTS
Description:	Oracle Access Manager Identity Asserter
Version:	1.0
Control Flag:	REQUIRED

Active Types:

Available:

Chosen:
□ ObSSOCookie
□ OAM_REMOTE_USER

. . .

Base64 Decoding Required:	false

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

Transport Security:	open
Minimum Access Server Connections In Pool:	5
Application Domain:	
Access Gate Password:	
Please type again To confirm:	
Key Store Pass Phrase:	

. . .

SSOHeader Name:	OAM_REMOTE_USER1

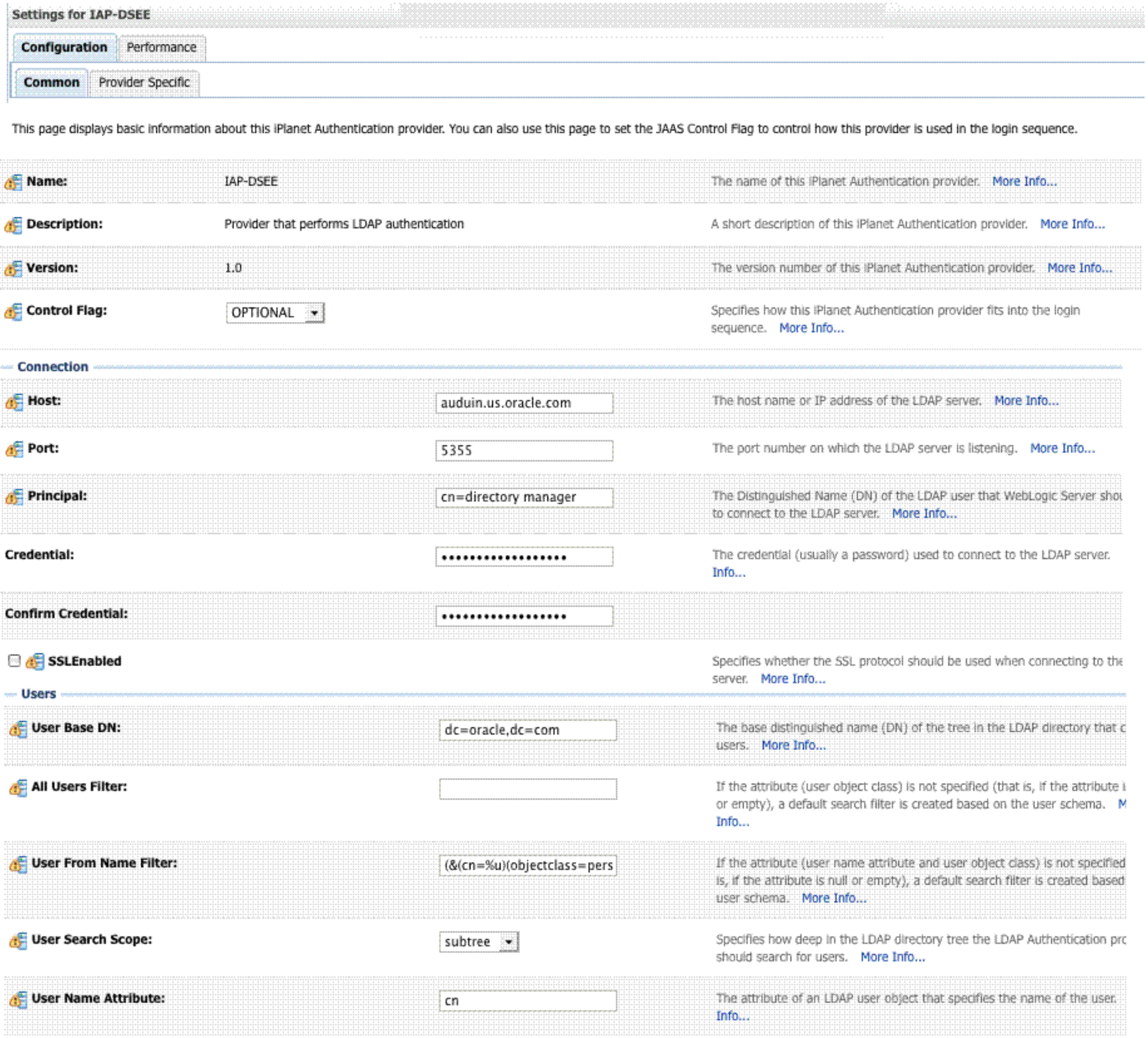
. . .

LDAP Authentication Provider Details

Create the Authenticator for the LDAP with the OPTIONAL JAAS flag. This will point to the Default System Store of Oracle Access Manager, which provides the OAM token.

Figure 16–8 illustrates this.

Figure 16–8 LDAP Provider: IAP-DSEE



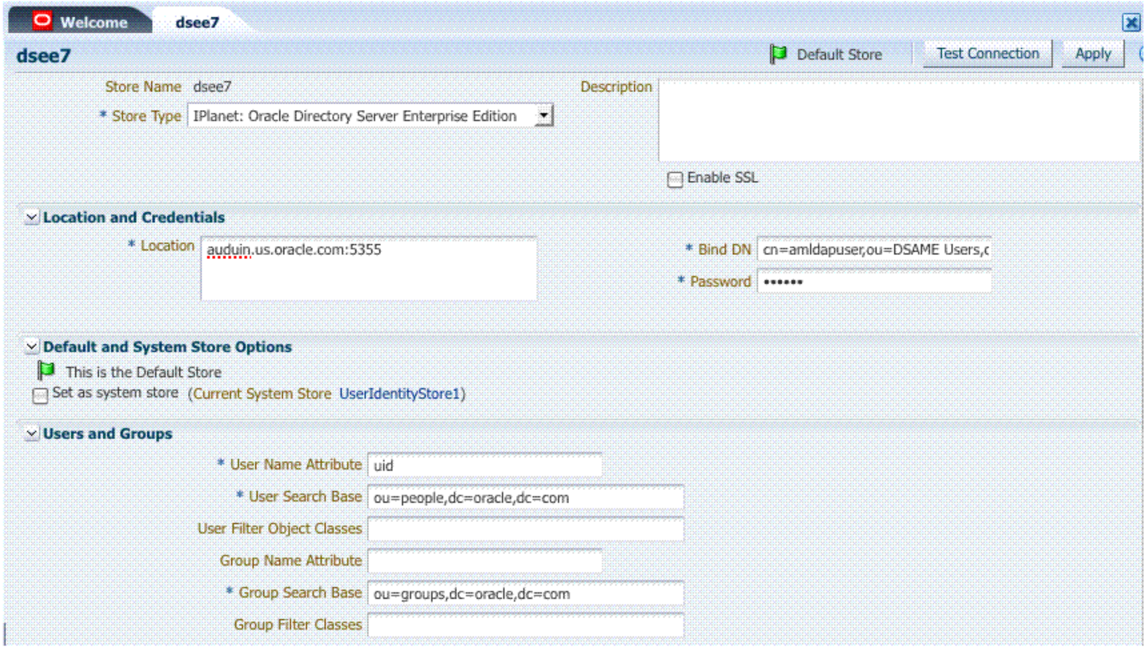
This page displays basic information about this iPlanet Authentication provider. You can also use this page to set the JAAS Control Flag to control how this provider is used in the login sequence.

Name:	IAP-DSEE	The name of this iPlanet Authentication provider. More Info...
Description:	Provider that performs LDAP authentication	A short description of this iPlanet Authentication provider. More Info...
Version:	1.0	The version number of this iPlanet Authentication provider. More Info...
Control Flag:	OPTIONAL	Specifies how this iPlanet Authentication provider fits into the login sequence. More Info...
Connection		
Host:	auduin.us.oracle.com	The host name or IP address of the LDAP server. More Info...
Port:	5355	The port number on which the LDAP server is listening. More Info...
Principal:	cn=directory manager	The Distinguished Name (DN) of the LDAP user that WebLogic Server shou to connect to the LDAP server. More Info...
Credential:	••••••••••••••••••	The credential (usually a password) used to connect to the LDAP server. Info...
Confirm Credential:	••••••••••••••••••	
□ SSLEnabled		Specifies whether the SSL protocol should be used when connecting to the server. More Info...
Users		
User Base DN:	dc=oracle,dc=com	The base distinguished name (DN) of the tree in the LDAP directory that c users. More Info...
All Users Filter:		If the attribute (user object class) is not specified (that is, if the attribute i or empty), a default search filter is created based on the user schema. M Info...
User From Name Filter:	(&(cn=%u)(objectclass=pers	If the attribute (user name attribute and user object class) is not specified is, if the attribute is null or empty), a default search filter is created based user schema. More Info...
User Search Scope:	subtree	Specifies how deep in the LDAP directory tree the LDAP Authentication prc should search for users. More Info...
User Name Attribute:	cn	The attribute of an LDAP user object that specifies the name of the user. Info...

Default Identity Store Configuration

Figure 16–9 illustrates the Default Identity Store configuration within Oracle Access Manager Console.

Figure 16–9 Default Identity Store Defined in Oracle Access Manager



dsee7

Store Name dsee7

* Store Type IPlanet: Oracle Directory Server Enterprise Edition

Description

□ Enable SSL

Location and Credentials

* Location auduin.us.oracle.com:5355

* Bind DN cn=amldapuser,ou=DSAME Users,c

* Password ••••••

Default and System Store Options

This is the Default Store

□ Set as system store (Current System Store UserIdentityStore1)

Users and Groups

* User Name Attribute uid

* User Search Base ou=people,dc=oracle,dc=com

User Filter Object Classes

Group Name Attribute

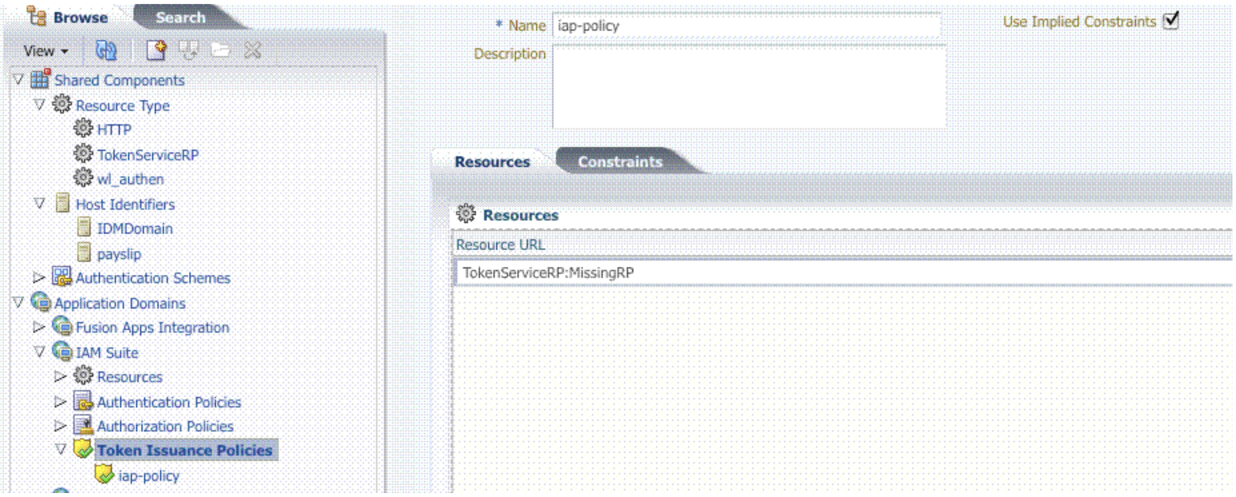
* Group Search Base ou=groups,dc=oracle,dc=com

Group Filter Classes

Token Issuance Policy

Create the Token Issuance Policy for the resource URL within the IAM Suite Application Domain, as shown in Figure 16–10.

Figure 16–10 Token Issuance Policy for Identity Propagation



- Shared Components
  - Resource Type
    - HTTP
    - TokenServiceRP
    - wl_authen
  - Host Identifiers
    - IDMDomain
    - payslip
  - Authentication Schemes
- Application Domains
  - Fusion Apps Integration
  - IAM Suite
    - Resources
    - Authentication Policies
    - Authorization Policies
    - Token Issuance Policies
      - iap-policy

* Name iap-policy

Description

Use Implied Constraints ☑

Resources

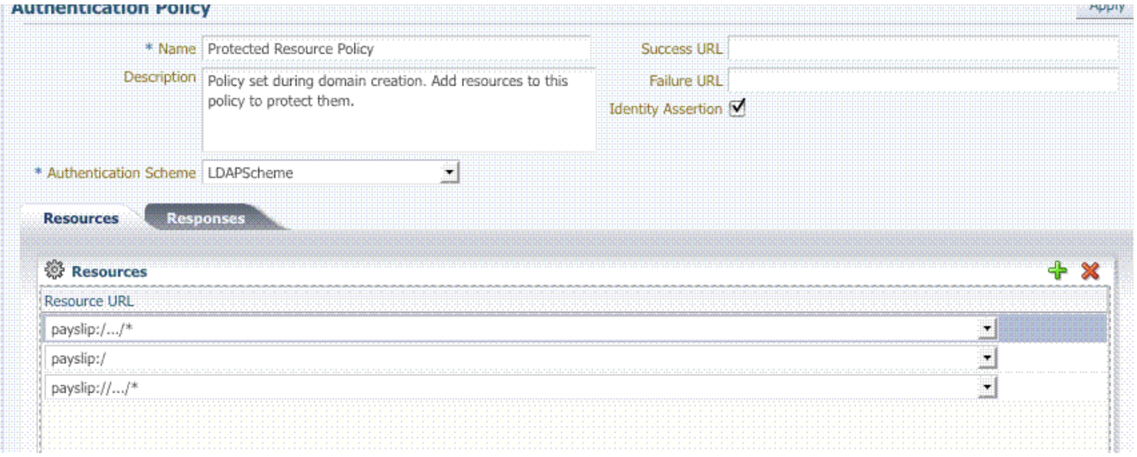
Resource URL

TokenServiceRP:MissingRP

Authentication Policy for Identity Assertion by Webgate

Confirm that the Identity Assertion box is checked in the Authentication Policy within the IAM Suite Application Domain. This enables Webgate to perform Identity Assertion for protected resources, as shown in Figure 16–10.

Figure 16–11 Authentication Policy for Identity Assertion by Webgate



Authentication Policy

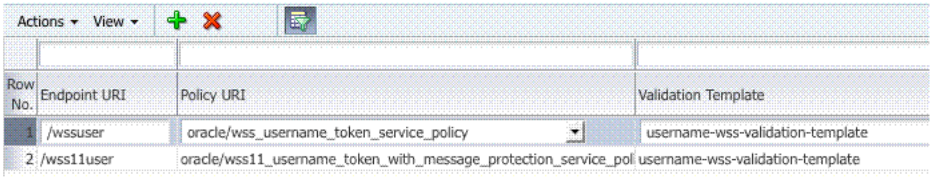
* Name	Protected Resource Policy	Success URL	
Description	Policy set during domain creation. Add resources to this policy to protect them.	Failure URL	
		Identity Assertion	☑
* Authentication Scheme	LDAPScheme		

Resource URL
payslip:/.../*
payslip:/
payslip://.../*

Endpoint Configuration

The /wss10user Endpoint is needed, as shown in Figure 16–12. This endpoint is protected by the default WS-Security Validation Template. This is the one that will be used in the Web application to post the RST.

Figure 16–12 /wssuser Endpoint for Identity Assertion



Row No.	Endpoint URI	Policy URI	Validation Template
1	/wssuser	oracle/wss_username_token_service_policy	username-wss-validation-template
2	/wss11user	oracle/wss11_username_token_with_message_protection_service_pol	username-wss-validation-template

Issuance Template Configuration

The Issuance Template requires the following configuration for Identity Propagation:

- Name: iap-issuance-template
- Description: Custom issuance template
- Token Type: SAML 2.0
- Signing Key Id: osts_signing
- Description: Custom issuance template

Partner Configuration: Requester

Create a new Requester Partner configuration for Identity Propagation with the OAM token as follows:

- Partner Name: iap-request-partner
- Requester Type: STS_REQUESTER
- Partner Profile: iap-requestor-profile
- Description: Custom requester
- Trusted

- Username Token Authentication
  - *Username* <enter username used by the Web Service Client>
  - *Password* <enter password used by the Web Service Client>
  - Confirm *Password* <enter password used by the Web Service Client>
- Identity Attribute values for:
  - httpbasicusername
  - sslclientcertdn

Partner Profile: Relying Party

Create a new Relying Party Profile for Identity Propagation as follows:

- Profile ID: iap-relyingparty-profile
- Description: iap-issuance-template
- Default Token Type: SAML 2.0
- Default Template: iap-issuance-template

Partner Profile: Requester

Create a new Requester Profile for Identity Propagation as follows:

- Profile ID: iap-requestor-profile
- Description: iap-requestor-profile partner profile
- Default Relying Party Profile: iap-relyingparty-profile

Validation Template for WS-TRUST

The Validation Template requires the following configuration for Identity Propagation:

- Validation Template Name: iap_wstrust_validation_template
- Description: iap_wstrust_validation_template
- Token Protocol: WS-Trust
- Token Type: OAM
- Timestamp Lifespan:

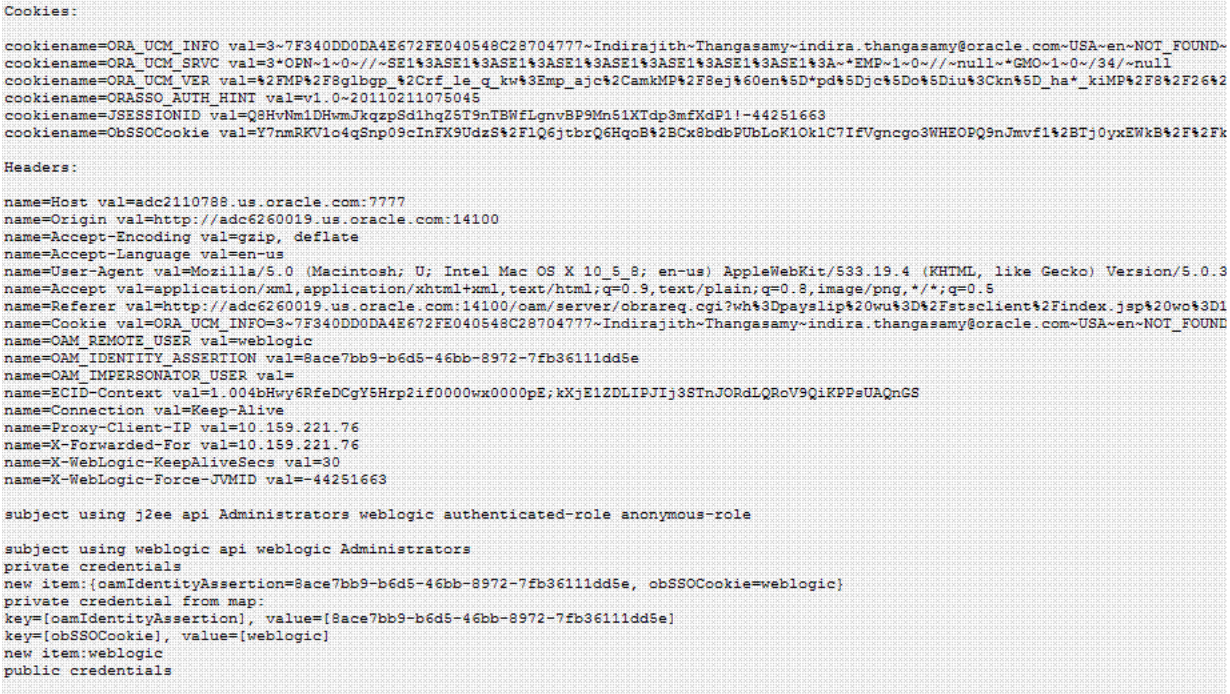
This completes the configuration requirements for the Identity Propagation with OAM Token scenario.

16.3.4 Testing Your Implementation

Following configuration, you can try to access the resource to confirm your implementation is working properly.

Webgate should redirect to the OAM Server if there is no existing session for the user Upon successful authentication, you should be able to see the RST and RSTR sent by the STS server.

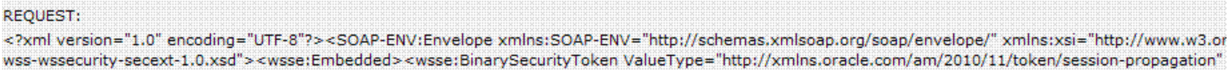
Cookies and Headers (Truncated)



```
Cookies:

cookiename=ORA_UCM_INFO val=3~7F340DD0DA4E672FE040548C28704777~Indirajith~Thangasamy~indira.thangasamy@oracle.com~USA~en~NOT_FOUND~
cookiename=ORA_UCM_SRVC val=3*OPN~1~0~//~SE1%3ASE1%3ASE1%3ASE1%3ASE1%3ASE1%3ASE1%3ASE1%3A~*EMP~1~0~//~null~*GMO~1~0~/34/~null
cookiename=ORA_UCM_VER val=%2FMP%2F8glbgp_%2Crf_le_q_kw%3Emp_ajc%2CamkMP%2F8ej%60en%5D*pd%5Djc%5Do%5Diu%3Ckn%5D_ha*_kiMP%2F8%2F26%2
cookiename=ORASSO_AUTH_HINT val=v1.0~20110211075045
cookiename=JSESSIONID val=Q8HvNm1DHwmJkqzpSd1hqZ5T9nTBWfLgnvBP9Mn51XTdp3mfXdP1!-44251663
cookiename=ObSSOCookie val=Y7nmRKV1o4qSnp09cInFX9UdzS%2F1Q6jtbrQ6HqoB%2BCx8bdbPUbLoK1Ok1C7IfVgncgo3WHEOPQ9nJmvf1%2BTj0yxEWkB%2F%2Fk

Headers:

name=Host val=adc2110788.us.oracle.com:7777
name=Origin val=http://adc6260019.us.oracle.com:14100
name=Accept-Encoding val=gzip, deflate
name=Accept-Language val=en-us
name=User-Agent val=Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_8; en-us) AppleWebKit/533.19.4 (KHTML, like Gecko) Version/5.0.3
name=Accept val=application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
name=Referer val=http://adc6260019.us.oracle.com:14100/oam/server/obrareq.cgi?wh%3Dpayslip%20wu%3D%2Fstsclient%2Findex.jsp%20wo%3D1
name=Cookie val=ORA_UCM_INFO=3~7F340DD0DA4E672FE040548C28704777~Indirajith~Thangasamy~indira.thangasamy@oracle.com~USA~en~NOT_FOUND
name=OAM_REMOTE_USER val=weblogic
name=OAM_IDENTITY_ASSERTION val=8ace7bb9-b6d5-46bb-8972-7fb36111dd5e
name=OAM_IMPERSONATOR_USER val=
name=ECID-Context val=1.004bHwy6RfeDCgY5Hrp2if0000wx0000pE;kXjE1ZDLIPJIj3STnJORdLQRoV9QiKPPsUAQnGS
name=Connection val=Keep-Alive
name=Proxy-Client-IP val=10.159.221.76
name=X-Forwarded-For val=10.159.221.76
name=X-WebLogic-KeepAliveSecs val=30
name=X-WebLogic-Force-JVMID val=-44251663

subject using j2ee api Administrators weblogic authenticated-role anonymous-role

subject using weblogic api weblogic Administrators
private credentials
new item:{oamIdentityAssertion=8ace7bb9-b6d5-46bb-8972-7fb36111dd5e, obSSOCookie=weblogic}
private credential from map:
key=[oamIdentityAssertion], value=[8ace7bb9-b6d5-46bb-8972-7fb36111dd5e]
key=[obSSOCookie], value=[weblogic]
new item:weblogic
public credentials
```

Request Security Token Sent By the Client (Truncated)

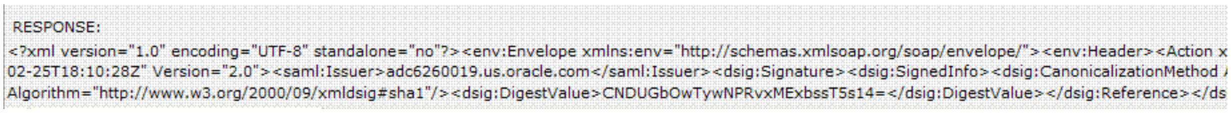
Here is a (truncated) request for security token sent by the client.



```
REQUEST:
<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.or
wss-wssecurity-secext-1.0.xsd"><wsse:Embedded><wsse:BinarySecurityToken ValueType="http://xmlns.oracle.com/am/2010/11/token/session-propagation"
```

Request Security Token Response sent by the OSTS Server (Truncated)

Here is a (truncated) response to the RST sent by the OSTS Server.



```
RESPONSE:
<?xml version="1.0" encoding="UTF-8" standalone="no"?><env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"><env:Header><Action x
02-25T18:10:28Z" Version="2.0"><saml:Issuer>adc6260019.us.oracle.com</saml:Issuer><dsig:Signature><dsig:SignedInfo><dsig:CanonicalizationMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><dsig:DigestValue>CNDUGbOwTywNPRvxMExbssT5s14=</dsig:DigestValue></dsig:Reference></ds
```

16.4 Scenario: Web Service Security With On Behalf Of Username Token

This section provides the following topics:

- Component interactions for Identity Propagation with Username Token
- RST Attributes and Processing for Identity Propagation with a Username Token
- Configuration Requirements: Identity Propagation with the Username Token

16.4.1 Component interactions for Identity Propagation with Username Token

Process overview: Component interactions for Identity Propagation

1. User attempts to access a protected resource.
2. User is authenticated.
3. The WebLogic container sets the user's identity into a Subject for this session.
4. When the resource is finally accessed, a Web Service Client can then obtain the user's identity from current user's Subject, generates a OnBehalfOf (OBO) token with it, then creates and sends Request Security Token (RST) to OSTS.
5. Oracle Security Token Service authenticates the Web Service Client as a Requester Partner.
6. Oracle Security Token Service sees the Username Token inside OBO token, it maps the maps the user's identity to a user record in LDAP.
7. Oracle Security Token Service then generates Request Security Token Response.
8. Oracle Security Token Service sends Request Security Token Response to the client, which can then use the token inside the Request Security Token Response (RSTR) to create a web service request to access a service hosted on a relying party.

16.4.2 RST Attributes and Processing for Identity Propagation with a Username Token

For an incoming Request Security Token (RST) with the following attributes, Oracle Security Token Service must be configured to process a request and issue a token.

RST Attributes for Identity Propagation with a Username Token

- The SOAP header contains a Username token referencing a WS Requester. The Username token contains at least a username and a password.
- The SOAP body contains a WS-Trust RST message.
- The RST contains a Username Token in the OnBehalfOf field referencing a user in LDAP.
- The RST can possibly contain an AppliesTo field holding a URL pointing to the endpoint of the Relying Party Web Service.
- The RST can possibly contain a TokenType field holding the type of token that needs to be returned.
- The RST can possibly contain an Entropy field holding random data that will be used when creating the SecretKey when a symmetric proof key is required in the SAML Assertion.
- The RST can possibly contain a UseKey field holding the certificate or public key to be used as an asymmetric proof key in the SAML Assertion, but this field will be ignored by Oracle Security Token Service.

Process overview: Identity Propagation with the OAM Token

1. Client prepares the request by:
   - Creating the SOAP message
   - Creating the Username token referencing the client and including it in the SOAP header.
   - Creating the WS-Trust RST message.

- Creating the Username Token referencing the user and including it in the OnBehalfOf field of the RST.
- Including the RST message in the SOAP body.

2. Client sends the message to the Oracle Security Token Service, to an endpoint protected by a WS-Security User Name Token (UNT) Policy, with that endpoint being mapped to an Oracle Security Token Service WSS Validation Template.
3. Oracle Security Token Service will process the incoming request.
4. Oracle Security Token Service validates the token included in the SOAP header by using the settings contained in the WS-Security Validation Template:
   - Validates the format of the Username token.
   - Validates the credentials contained in the Username token against the Oracle Security Token Service Partner store, thus mapping this token to a Requester Partner.
   - Knowing the Requester Partner, Oracle Security Token Service will retrieve the Requester Partner Profile associated with this Requester.
5. Oracle Security Token Service then validates the token present in the OnBehalfOf field:
   - Determines the type of token present in the OnBehalfOf field.
   - Retrieves the WS-Trust Validation Template to be used for Username Token Type, from the Requester Partner Profile.
   - Validates the Username Token, and maps the token to a user.
6. Oracle Security Token Service then examines the AppliesTo field:
   - If present, Oracle Security Token Service will attempt to map the AppliesTo URL to a Relying Party Partner, using the WS Endpoint Mapping of the Relying Party Partner. If the mapping is successful, then the AppliesTo field has been mapped to a Relying Party Partner, and Oracle Security Token Service will retrieve the Relying Party Partner Profile from this Partner. If mapping was not successful, then the AppliesTo field could not be mapped to a Relying Party Partner, and Oracle Security Token Service will retrieve the Default Relying Party Partner Profile from the Requester Partner Profile.
   - If absent, Oracle Security Token Service will retrieve the Default Relying Party Partner Profile from the Requester Partner Profile.
7. Oracle Security Token Service then examines the TokenType field:
   - If present, Oracle Security Token Service will map the TokenType string to a local token type value using the Requester Partner Profile, and it will then use the Relying Party Partner Profile to retrieve the Issuance Template to be used to create the outgoing token.
   - If absent, Oracle Security Token Service will retrieve the default token type from the Relying Party Partner Profile, and it will then use the Relying Party Partner Profile to retrieve the Issuance Template to be used to create the outgoing token.
8. Oracle Security Token Service will perform an Authorization evaluation to check that the Requester Partner is authorized to request a token for the Relying Party referenced in the flow (see Authorization Trust Policy for more information).
9. Oracle Security Token Service will then create the token:

- If the token to be issued is of SAML type, then the Issuance Template will list how to populate the NameID, the Relying Party Partner Profile will list which attributes need to be sent in the token, the Issuance Template will indicate whether or not to translate the names and values of the attributes, the Issuance Template will indicate whether or not to sign/encrypt the token.
- If the token to be issued if of SAML type, the Oracle Security Token Service server will examine the KeyType to determine the Subject Confirmation Method of the Assertion. If it is missing, it will use the Default Confirmation Method from the Issuance Template.

10. Oracle Security Token Service will create the Response that the client will process:
    - Creates the WS-Trust RSTRC
    - Includes the returned token
    - Includes proof key if necessary

16.4.3 Configuration Requirements: Identity Propagation with the Username Token

This topic walks through the configuration requirements for the identity propagation scenario. It includes:

- Configuration overview: Identity Propagation with the Username Token
- Default Identity Store Configuration
- Token Issuance Policy
- Endpoint Configuration
- Issuance Template Configuration
- Partner Configuration: Requester
- Partner Profile: Relying Party
- Partner Profile: Requester
- Validation Template for WS-TRUST
- Example 16–1, "Sample exchange: Request Security Token Sent By the Client"
- Example 16–2, "Request Security Token Response sent by the OSTS Server"

Configuration overview: Identity Propagation with the Username Token

Following is an overview of the Identity Propagation environment and implementation tasks:

- A web application where the user will request. This web application will authenticate the user, then attempt to send a SOAP message to a remote Web Service Provider. As part of that SOAP exchange, the WS-Security client will download the WS-Security policy of the Web Service Provider, connect to the Oracle Security Token Service to retrieve the token requested by the Web Service Provider, send the Security Token with the SOAP message to the Web Service Provider.
- OSTS Server URL: http://*yourhost.domain.com*:14100/sts

The following Oracle Security Token Service configuration is required to implement token processing for identity propagation:

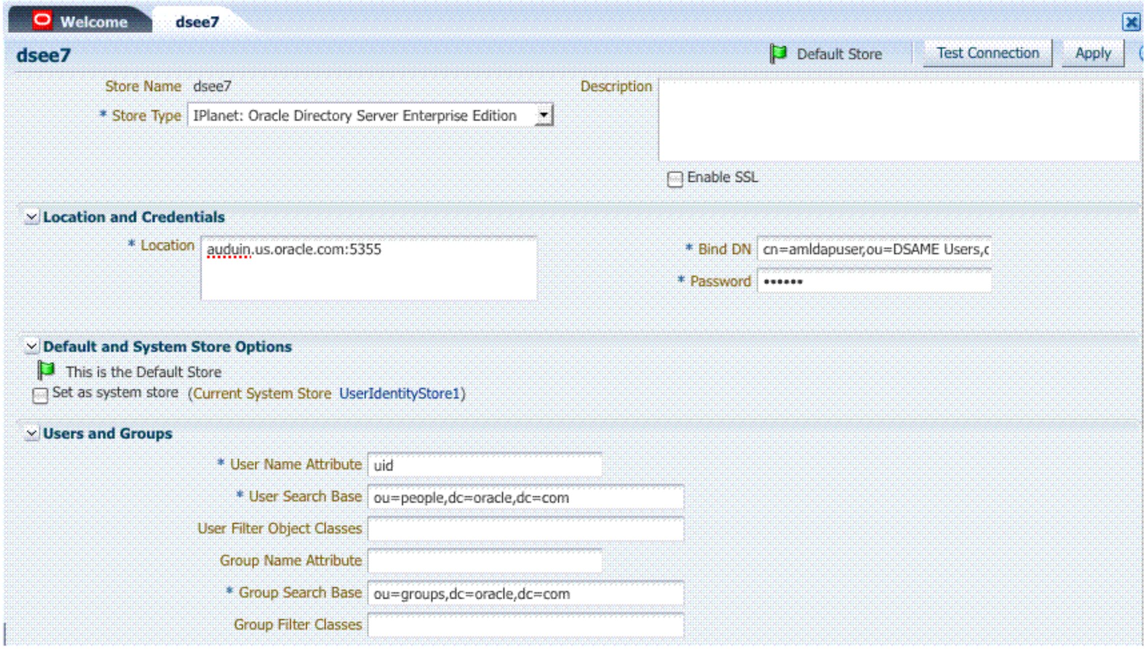
- One Requester Partner Profile

- One Relying Party Partner Profile
- One Issuance Template
- One WS-Trust Validation Template
- Oracle Security Token Service Endpoint
- An LDAP server is required for Oracle Security Token Service to map the Username token referencing the user to an LDAP User record, and thus use that record to populate the outgoing token.
- Ensure that the desired LDAP server is configured as the Default Identity Store for Oracle Access Manager with Oracle Security Token Service.

Default Identity Store Configuration

Figure 16–13 illustrates the Default Identity Store configuration within Oracle Access Manager Console.

Figure 16–13 Default Identity Store Defined in Oracle Access Manager



dsee7

Store Name dsee7
* Store Type IPlanet: Oracle Directory Server Enterprise Edition
Description
□ Enable SSL

Location and Credentials

* Location auduin.us.oracle.com:5355
* Bind DN cn=amldapuser,ou=DSAME Users,c
* Password ••••••

Default and System Store Options

This is the Default Store
□ Set as system store (Current System Store UserIdentityStore1)

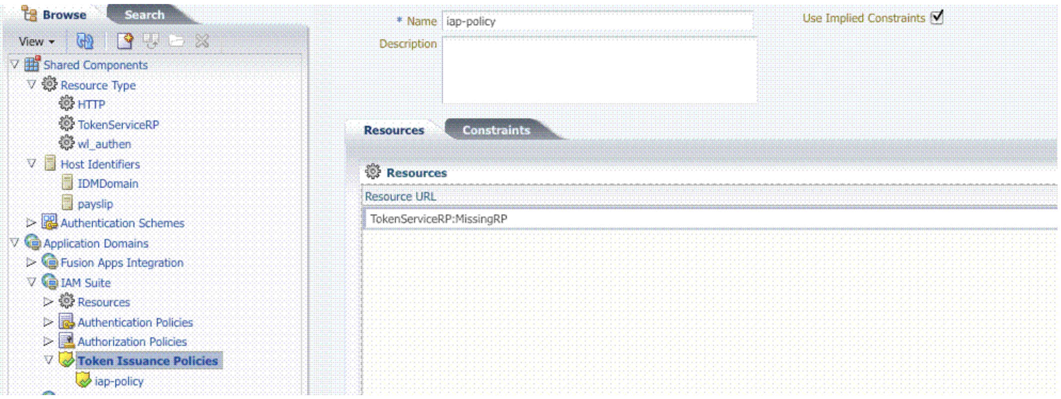
Users and Groups

* User Name Attribute uid
* User Search Base ou=people,dc=oracle,dc=com
User Filter Object Classes
Group Name Attribute
* Group Search Base ou=groups,dc=oracle,dc=com
Group Filter Classes

Token Issuance Policy

Create the Token Issuance Policy for the resource URL within the IAMSuite Application Domain, as shown in Figure 16–14.

Figure 16–14 Token Issuance Policy for Identity Propagation



* Name iap-policy

Description

Use Implied Constraints ☑

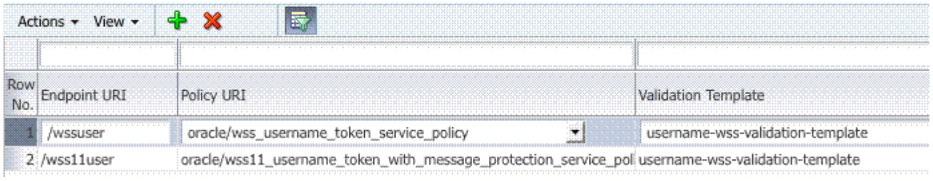
Resources

Resource URL
TokenServiceRP:MissingRP

Endpoint Configuration

The /wss11user Endpoint is needed, as shown in Figure 16–15. This endpoint is protected by the default WS-Security Validation Template. This is the one that will be used in the Web application to post the RST.

Figure 16–15 /wss11user Endpoint for Identity Assertion



Row No.	Endpoint URI	Policy URI	Validation Template
1	/wssuser	oracle/wss_username_token_service_policy	username-wss-validation-template
2	/wss11user	oracle/wss11_username_token_with_message_protection_service_pol	username-wss-validation-template

Issuance Template Configuration

The Issuance Template requires the following configuration for Identity Propagation:

- Name: saml-issuance-template
- Description: SAML issuance template
- Token Type: SAML 2.0
- Signing Key Id: osts_signing

Partner Configuration: Requester

Create a new Requester Partner configuration for Identity Propagation with the OAM token as follows:

- Partner Name: requester-partner
- Partner Type: Requester
- Partner Profile: requester-profile
- Description: Requester
- Trusted
- Username Token Authentication
  - *Username* <enter username used by the Web Service Client>
  - *Password* <enter password used by the Web Service Client>

  - Confirm *Password* <enter password used by the Web Service Client>
- Identity Attribute values for:
  - httpbasicusername
  - sslclientcertdn

Partner Profile: Relying Party

Create a new Relying Party Profile for Identity Propagation as follows:

- Profile ID: relying-party-profile
- Description: Relying Party Profile
- Default Token Type: SAML 2.0
- Issuance Template: iap-issuance-template for SAML 2.0

Partner Profile: Requester

Create a new Requester Profile for Identity Propagation as follows:

- Profile ID: requester-profile
- Description: Requester Partner Profile
- Default Relying Party Profile: relying-party-profile

Validation Template for WS-TRUST

The Validation Template requires the following configuration for Identity Propagation:

- Validation Template Name: username_wstrust_validation_template
- Description: Username WS-Trust Template
- Token Protocol: WS-Trus
- Token Type: Username
- Timestamp Lifespan: 600
- Enable Credential Validation: unchecked
- Token Mapping:
  - Map Token To User: checked
  - Enable Simple User Mapping: checked
  - Datastore Attribute: uid

This completes the configuration requirements for the Identity Propagation with Username Token scenario.

Example 16–1 Sample exchange: Request Security Token Sent By the Client

Here is a request for security token sent by the client.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
<SOAP-ENV:Header><wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-sec
ext-1.0.xsd">
<wsse:UsernameToken><wsse:Username>requester-test</wsse:Username><wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profi
le-1.0#PasswordText">welcome1</wsse:Password></wsse:UsernameToken></wsse:Security>
```

```
<wsa:Action
xmlns:wsa="http://www.w3.org/2005/08/addressing">http://docs.oasis-open.org/ws-sx/
ws-trust/200512/RST/Issue</wsa:Action></SOAP-ENV:Header>
<SOAP-ENV:Body><wst:RequestSecurityToken
xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"><wst:RequestType>http
://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
<wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAM
LV1.1</wst:TokenType><wst:OnBehalfOf>
<wsse:UsernameToken
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-sec
ext-1.0.xsd"><wsse:Username>user-alice</wsse:Username>
</wsse:UsernameToken></wst:OnBehalfOf></wst:RequestSecurityToken></SOAP-ENV:Body><
/SOAP-ENV:Envelope>
```

Example 16–2 Request Security Token Response sent by the OSTS Server

Here is a response to the RST sent by the OSTS Server.

```
<env:Envelope
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"><env:Header><Action
xmlns="http://www.w3.org/2005/08/addressing">http://docs.oasis-open.org/ws-sx/ws-t
rust/200512/RSTRC/IssueFinal</Action>
</env:Header><env:Body><wst:RequestSecurityTokenResponseCollection
xmlns:ns6="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-sec
ext-1.0.xsd" xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-util
ity-1.0.xsd">
<wst:RequestSecurityTokenResponse><wst:TokenType>http://docs.oasis-open.org/wss/oa
sis-wss-saml-token-profile-1.1#SAMLV1.1</wst:TokenType><wst:RequestedSecurityToken
><saml:Assertion AssertionID="id-1LNkSUVcpbH7O0oQwbHJ5JOd5fs-"
IssueInstant="2011-04-22T18:48:05Z" Issuer="adc2110618.us.oracle.com"
MajorVersion="1" MinorVersion="1" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:enc="http://www.w3.org/2001/04/xmlenc#"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<saml:Conditions NotBefore="2011-04-22T18:48:05Z"
NotOnOrAfter="2011-04-22T19:48:05Z"/><saml:AttributeStatement><saml:Subject><saml:
NameIdentifier
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">user-alice@oracle.
com</saml:NameIdentifier><saml:SubjectConfirmation>
<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</saml:Confi
rmationMethod></saml:SubjectConfirmation>
</saml:Subject><saml:Attribute AttributeName="sn"
AttributeNamespace="urn:oracle:security:fed:attrnamespace"><saml:AttributeValue
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xsi:type="xs:string">user-alice-last</saml:AttributeValue></saml:Attribute>
</saml:AttributeStatement><dsig:Signature><dsig:SignedInfo><dsig:CanonicalizationM
ethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<dsig:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/><dsig:Reference
URI="#id-1LNkSUVcpbH7O0oQwbHJ5JOd5fs-"><dsig:Transforms><dsig:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<dsig:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/></dsig:Transforms><dsig:Diges
tMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>1GF2ZT9h+gs8sxyO+/yG/N6jxk8=</dsig:DigestValue></dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>InZVb5aRM5+KKI1VqXg9HiIgLjKyGm0VkD6sMJ/8SIbFbbxuNm7Mnky5W35p2
```

```
P0c5bCPRx02uzLEE4KhLkyM2GsLsVaDNkRztGMphQW/Mcg7DprJIEyR2OYMYDOQSipa/k2K98C4zO/RNiv
o1KvyJsd6a3h6CBHwoO1RKip039w=</dsig:SignatureValue>
<dsig:KeyInfo><dsig:X509Data><dsig:X509Certificate>MIIB/DCCAWWgAwIBAgIBCjANBgkqhki
G9w0BAQQFADAjMSEwHwYDVQQDExhhZGMyMTEwNjE4LnVzLm9yYWNsZS5jb20wHhcNMTEwNDE5MTUxNTI2W
hcNMjEwNDE2MTUxNTI2WjAjMSEwHwYDVQQDExhhZGMyMTEwNjE4LnVzLm9yYWNsZS5jb20wgZ8wDQYJKoZ
IhvcNAQEBBQADgY0AMIGJAoGBAJnSxVc86TGcwewieaueIVG33C3Qouve6HuJxHsoM8cRRkJcmv+0auyvD
LJfYAEOfHo5OsF4+za11iNPln9ZFaOjUy/Y8JC0kSVxatgU36RveIrpOJvp978Oa6IlMNUtdFf8q3Trsiz
spE2hnbLY+0SMofgnAPcJEKPxkd6b0b0ZAgMBAAGjQDA+MAwGA1UdEwEB/wQCMAAwDwYDVR0PAQH/BAUDA
wfYADAdBgNVHQ4EFgQU47ZqWHgTOmZO67uw4YzsbrRMNOswDQYJKoZIhvcNAQEEBQADgYEAH0QIHaLMN/7
hD2VP0SLOCtNdEmY5IqLY1CDW+GpUZZ9e+MCgE/rvr34566D9Q8lvET6T9u+sg3h+hSkb3gE4a4wgShH/V
7nfHzx8ZntlxccvCZK6ePVDMt0Lfj2iVnE7IJxou4bO0w0m9DrvyKop7ncnSEzaVpxIZgCDo7+8Zdw=</d
sig:X509Certificate>
</dsig:X509Data></dsig:KeyInfo></dsig:Signature></saml:Assertion></wst:RequestedSe
curityToken><wst:RequestedAttachedReference><wsse:SecurityTokenReference><wsse:Key
Identifier
ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0#SAMLAss
ertionID">id-1LNkSUVcpbH7O0oQwbHJ5JOd5fs-</wsse:KeyIdentifier></wsse:SecurityToken
Reference></wst:RequestedAttachedReference>
<wst:RequestedUnattachedReference><wsse:SecurityTokenReference><wsse:KeyIdentifier
ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0#SAMLAss
ertionID">id-1LNkSUVcpbH7O0oQwbHJ5JOd5fs-</wsse:KeyIdentifier></wsse:SecurityToken
Reference></wst:RequestedUnattachedReference>
<wst:Lifetime><wsu:Created>2011-04-22T18:48:05Z</wsu:Created><wsu:Expires>2011-04-
22T19:48:05Z</wsu:Expires></wst:Lifetime></wst:RequestSecurityTokenResponse></wst:
RequestSecurityTokenResponseCollection></env:Body></env:Envelope>
```

17

Managing Oracle Security Token Service Settings and Set Up

This chapter introduces how to manage components involved in the protection of Oracle Security Token Service endpoints. This chapter provides the following topics:

- Prerequisites
- Introduction to Oracle Security Token Service Configuration
- Enabling and Disabling Oracle Security Token Service
- Defining Security Token Service Settings Using Oracle Access Manager Console
- Using and Managing WSS Policies for Oracle WSM Agents
- Configuring OWSM for WSS Protocol Communication
- Managing and Migrating Oracle Security Token Service Policies
- Introduction to Logging Oracle Security Token Service Messages
- Introduction to Auditing for Oracle Security Token Service
- Auditing Oracle Security Token Service Administrative and Run-time Events

17.1 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 Oracle Security Token Service" on page 1-10
- Chapter 3, "Getting Started with Common Administration and Navigation"
- Chapter 6, "Managing Common OAM Server Registration"

17.2 Introduction to Oracle Security Token Service Configuration

Oracle Security Token Service a Web Service co-existing with Oracle Access Manager 11g. Oracle Security Token Service invokes some Oracle Access Manager components to validate and issue security tokens. Typically, the Web client can use the Oracle Security Token Service to request an outbound token, such as SAML, by providing a security token, like a Username Token or an X.509 Token.

Oracle Security Token Service is integrated with the Oracle Access Manager Console to provide a unified and consistent administration experience. All Oracle Security Token Service system configuration is done using the Oracle Access Manager Console.

Oracle Security Token Service provides:

- Tokens:
  - Validation Tokens: Standard (Username, X.509, Kerberos, SAML 1.1/2.0) and custom tokens. OnBehalfOf use cases (OAM Session ID Propagation Token and custom tokens through the integration engine) also supports the following standard tokens along with OAM sessionID propagation token and custom token (Username, X.509, SAML 1.1/2.0).
  - Issuance Tokens: Standard (Username, SAML 1.1/2.0) and custom tokens through the integration engine
- Configuration-driven token issuance and validation
- Enhanced auditing through identity propagation across multiple tiers and domains
- Consolidated shared-platform service interacts with internal (Oracle Access Manager SSO, Federation, Oracle Web Services Manager) and external services

This section provides the following topics:

- Post-Installation Configuration
- About Servers and Oracle Security Token Service
- About Oracle Security Token Service Clients
- About Agents and Oracle Security Token Service

> **See Also:** *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*

17.2.1 Post-Installation Configuration

After installation and server startup, you can access the Oracle Access Manager Console on the OAM Server. For example, if the URL to the OAM Server is http://*machine:14100*/oam, you might access:

- Oracle Access Manager Console at http://*machine:7001*/oamconsole
- Oracle Security Token Service: http://machine:14100/sts/wss11user?wsdl to view the WSDL of the /sts/wss11user endpoint that is available by default, to ensure that OSTS is correctly installed

  By default, OSTS is disabled and as such all runtime functionality as well as Web Service endpoints are disabled. To access those endpoints, OSTS must first be enabled using the Oracle Access Manager Console. Afterwards, the endpoints might be accessed

Post-installation configuration includes the tasks in the following outline, which point to other areas in this book for details.

Task overview: Oracle Security Token Service configuration requires

1. **Server Side Configuration**: Use the Oracle Access Manager Console for the following tasks.

   a. Service enablement "Enabling and Disabling Services for Oracle Security Token Service" on page 17-10

   b. Settings configuration "Managing Security Token Service Settings" on page 17-12

   c. Endpoint registration "Managing EndPoints" on page 19-30
   d. Token Issuance Template configuration "Managing Token Issuance Templates" on page 19-7
   e. Token Validation Template configuration "Managing Token Validation Templates" on page 19-15
   f. Partner Profile creation "Managing Token Service Partner Profiles" on page 20-7
   g. Partner configuration "Managing Token Service Partners" on page 20-2
   h. Token Issuance Policies to define an authorization rule for issuing tokens with OSTS, for a specific Relying Party "Managing Token Issuance Policies and Constraints with Oracle Access Manager" on page 19-31

2. Set up interactions with the Oracle WSM Agent as described in following topics:
   a. Using and Managing WSS Policies for Oracle WSM Agents
   b. Configuring OWSM for WSS Protocol Communication
3. Set up message logging, as described in "Introduction to Logging Oracle Security Token Service Messages" on page 17-20.
4. Configure event auditing, as described in "Auditing Oracle Security Token Service Administrative and Run-time Events" on page 17-22.
5. Configure lifecycle management
   a. Register the Oracle Security Token Service trust endpoint, as described in item 1c.
   b. Register the Requester or Relying Party Partner with Oracle Security Token Service, as described in "Managing Token Service Partners" on page 20-2.
   c. Monitor performance, as described in Chapter 26, "Monitoring Performance and Logs with Fusion Middleware Control".

17.2.2 About Servers and Oracle Security Token Service

With Oracle Access Manager 11g, all Oracle Security Token Service instances are installed on OAM Servers (also known as Managed Servers). Each server must be registered with Oracle Access Manager.

Oracle Security Token Service leverages the common infrastructure for shared services and the Oracle Access Manager 11g administration model.

Oracle Security Token Service support Web Services Security protocol 1.0 and 1.1 and process the following tokens, if present in the Security SOAP headers:

- Username token (UNT)
- SAML 1.1 or SAML 2.0 Assertion
- Kerberos
- X.509

Note: Managed servers hosting Oracle Access Manager with Oracle Security Token Service must be registered with Oracle Access Manager as described in Chapter 6, "Managing Common OAM Server Registration".

Third-Party Servers: Oracle Security Token Service interoperates with third party security token servers. For instance, a third party security token service can create a valid Security Assertion Markup Language (SAML) Assertion that can be consumed by Oracle Security Token Service.

17.2.3 About Oracle Security Token Service Clients

Oracle Security Token Service provides services to various Oracle clients (Oracle Web Services Manager client) or third party clients (Microsoft and IBM are two).

Oracle WSM Client: Oracle Web Services Manager client bindings are the responsibility of Oracle Web Services Manager (and out of scope for this book). For more information, see "WS-Trust Policies and Configuration Steps" in Oracle Fusion Middleware Security and Administrator's Guide for Web Services.

Third Party Clients: Require a secure key exchange between the Oracle WSM client and server. You simply import the Oracle Security Token Service certificate to the client.

During SOAP interactions, the WS-Security protocol might require the client to trust the signing/encryption certificate used for WSS operations by the OWSM Agent protecting the OSTS endpoint. In those cases, the OSTS administrator should extract the OSTS OWSM signing/encryption certificate used for WSS operations and provide it to the WS Client. For more information, see "Extracting the Oracle STS/Oracle WSM Signing and Encryption Certificate" on page 17-17.

17.2.4 About Agents and Oracle Security Token Service

Oracle Web Services Manager communicates through agents. This topic introduces the agents that operate with Oracle Security Token Service.

Oracle WSM Agent: The Oracle Web Services Manager (Oracle WSM) Agent is integrated with Oracle Security Token Service. This agent provides the Web Services Security support for Oracle Security Token Service Web Services endpoints.

- Protects Web Services endpoints of Oracle Security Token Service
- Provides WS-Security support for sending SOAP messages to Relying Parties. As part of that process, the OWSM Client might interact with Oracle Security Token Service to get a security token that will be presented to the Relying Party
- Interacts with Oracle Security Token Service for token acquisition and token validation

Oracle Security Token Service supports token acquisition and token validation by Oracle Web Services Manager (Oracle WSM) agents. Oracle Web Services Manager Agents are not required to use Oracle Security Token Service as part of their inbound or outbound security policy enforcement. Oracle Web Services Manager client bindings are the responsibility of Oracle Web Services Manager administrators.

The Oracle WSM Agent is used by Oracle Security Token Service to enforce message protection of the SOAP communication channel between Oracle Security Token Service and the client. The Oracle WSM Agent caches the OPSS Keystore (by default the default-keystore.jks keystore located in $DOMAIN_HOME/config/fmwconfig directory) which contains the trusted certificates involved when validating the WSS clients' certificates. Subsequent changes to the contents of the keystore or to its name, require a restart of the Managed Server using Oracle Enterprise Manager Fusion Middleware Control or WebLogic Server console, or NodeManager.

The Oracle WSM Agent available to Oracle Security Token Service must be configured to protect the Oracle Oracle Security Token Service endpoints, to perform the following tasks:

- Decrypt the request, if necessary
- Verify any digital signatures present in the request
- Validate any certificate used to create the request's digital signatures, if the signatures were created with a private key
- Validate any X.509 token, if present, in the SOAP headers
- Validate the Kerberos token, if present, in the SOAP headers
- Sign the outgoing response, if needed
- Encrypt the outgoing response, if required

Oracle WSM Agent Keystore: The Oracle WSM Agent uses a keystore for various cryptographic operations. For these operations, the Oracle WSM Agent uses the keystore configured for Oracle WSM tasks.

OAM Webgate: Oracle Security Token Service uses Webgate for the OAM 11g session propagation token. This, identity propagation, use case is more advanced. It requires the Identity Assertion Provider in WebLogic Server and some custom integration.

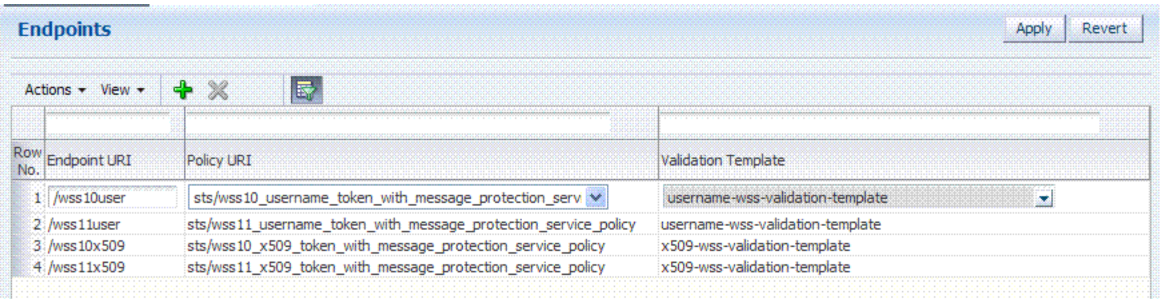
See Also:

- Chapter 16, "Oracle Security Token Service Implementation Scenarios"
- Chapter 16, "Oracle Security Token Service Implementation Scenarios"
- "About the Oracle Web Services Manager Keystore (default-keystore.jks)" on page 18-3

17.2.5 About Oracle Security Token Service End Points and Policies

When you add an endpoint, you can choose from a list of Policy URI's and validation templates with which to associate the Oracle Security Token Service endpoint. By default, Oracle Security Token Service is configured with the endpoints shown in Figure 17–1.

Figure 17–1 Default Endpoints, Policies, and Validation Templates



Row No.	Endpoint URI	Policy URI	Validation Template
1	/wss10user	sts/wss10_username_token_with_message_protection_serv	username-wss-validation-template
2	/wss11user	sts/wss11_username_token_with_message_protection_service_policy	username-wss-validation-template
3	/wss10x509	sts/wss10_x509_token_with_message_protection_service_policy	x509-wss-validation-template
4	/wss11x509	sts/wss11_x509_token_with_message_protection_service_policy	x509-wss-validation-template

The ORAPROVIDER is integrated with the Oracle WSM Agent, which provides Web Services Security support on the SOAP messages being exchanged between the client and Oracle Security Token Service. Oracle Security Token Service leverages ORAPROVIDER for Web Services to:

- publish Web Services endpoints dynamically
- invoke Oracle Security Token Service to process SOAP messages
- publish a WSDL file for each WS endpoint

Oracle WSM Agent WSS Policy Stores: The Oracle WSM Agent requires a repository to retrieve the Web Services Security (WSS) policies it needs. Oracle Security Token Service supports two types of repositories:

- **JAR file with WSS Policies**: Used when the WLS Domain is configured for classpath.
- **Oracle WSM Policy Manager** available from the SOA deployment

> **See Also:**
>
> - "Configuring OWSM for WSS Protocol Communication" on page 17-15
> - Managing Web Service Security policies for Oracle Security Token Service
> - *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for details about the policies for Oracle Security Token Service

Policy Assertions: Out of the box, Oracle Security Token Service provides a set of security policy assertions for use with the WS-Policy framework to describe how messages are to be secured in the context of Web Service Security: SOAP Message Security and WS-Trust.

- Oracle Security Token Service makes its associated security policy files publicly available by attaching them to its deployed WSDL.
- Oracle Security Token Service runtime uses the private key and X.509 certificate pairs, stored in the keystores defined by the jps-config.xml file, for its WS-Security encryption and digital signature operations.

The following paragraphs and tables identify the policies that are available out of the box for Oracle Security Token Service and the Oracle WSM Agent.

Message-level Security Not Required: When message level-security is not required, use an Oracle Security Token Service policy that does not specify message_protection in its name. This authenticates users using credentials provided in tokens in the WS-Security SOAP header. The credentials in the Fusion Applications token are mapped based on the rules specified in the validation template. Both plain text and digest mechanisms are supported.

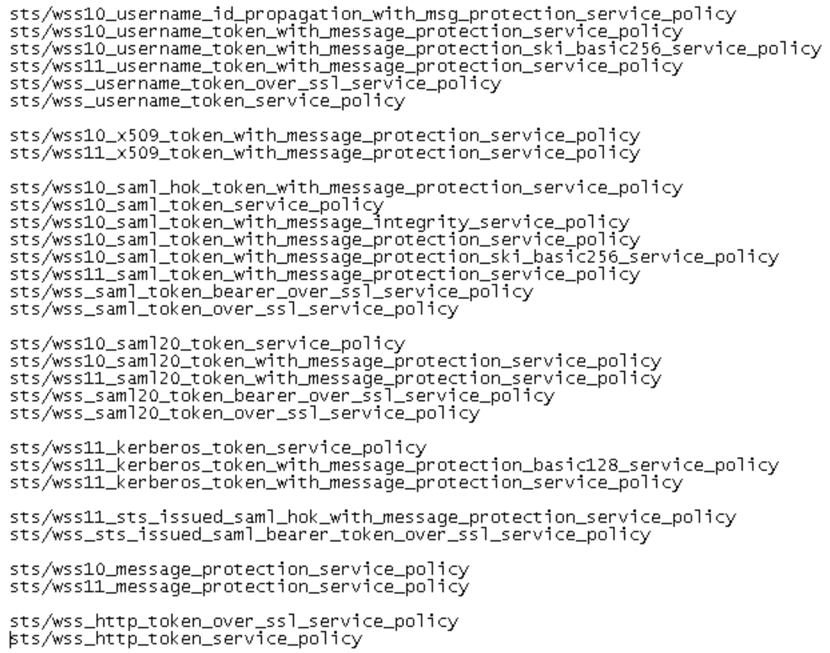
Transport Security when Message-level Security Not Required: You can configure two-way SSL where both the client applications and WebLogic server present certificates to each other. To configure two-way or one-way SSL for the core WebLogic Server security see "Configuring SSL" in *Oracle Fusion Middleware Securing Oracle WebLogic Server* guide. Use the policies described in Table 17–1.

Table 17–1 Policies Transport Security when Message-level Security Not Required

Policy	Description
sts/wss_username_token_over_ssl_service_policy	Maps users using credentials provided in UNT tokens in the WS-Security SOAP header. The credentials in the UNT token are mapped based on the rules specified in the validation template. Both plain text and digest mechanisms are supported.The policy verifies that the transport protocol provides SSL message protection.
sts/wss_saml_token_over_ssl_service_policy sts/wss_saml20_token_over_ssl_service_policy	Maps users using credentials provided in SAML SV tokens in the WS-Security SOAP header. The credentials in the SAML token are mapped based on the rules specified in the validation template. The policy verifies that the transport protocol provides SSL message protection
sts/wss_saml_token_bearer_over_ssl_service_policy sts/wss_saml20_token_bearer_over_ssl_service	Maps users using credentials provided in SAML tokens with confirmation method 'Bearer' in the WS-Security SOAP header. The credentials in the SAML token are mapped based on the rules specified in the validation template. The policy verifies that the transport protocol provides SSL message protection
sts/wss_sts_issued_saml_bearer_token_over_ssl_service_policy	

Interoperability WS-Security 1.0 and 1.1 Policies: Use policies in Figure 17–2 if you require interoperability with WS-Security 1.0 or 1.1 (depending on your authentication requirements and credential availability). Use WS-Security 1.1 policies if you have strong security requirements.

Figure 17–2 WS-Security 1.0 and 1.1 Policies



```
sts/wss10_username_id_propagation_with_msg_protection_service_policy
sts/wss10_username_token_with_message_protection_service_policy
sts/wss10_username_token_with_message_protection_ski_basic256_service_policy
sts/wss11_username_token_with_message_protection_service_policy
sts/wss_username_token_over_ssl_service_policy
sts/wss_username_token_service_policy

sts/wss10_x509_token_with_message_protection_service_policy
sts/wss11_x509_token_with_message_protection_service_policy

sts/wss10_saml_hok_token_with_message_protection_service_policy
sts/wss10_saml_token_service_policy
sts/wss10_saml_token_with_message_integrity_service_policy
sts/wss10_saml_token_with_message_protection_service_policy
sts/wss10_saml_token_with_message_protection_ski_basic256_service_policy
sts/wss11_saml_token_with_message_protection_service_policy
sts/wss_saml_token_bearer_over_ssl_service_policy
sts/wss_saml_token_over_ssl_service_policy

sts/wss10_saml20_token_service_policy
sts/wss10_saml20_token_with_message_protection_service_policy
sts/wss11_saml20_token_with_message_protection_service_policy
sts/wss_saml20_token_bearer_over_ssl_service_policy
sts/wss_saml20_token_over_ssl_service_policy

sts/wss11_kerberos_token_service_policy
sts/wss11_kerberos_token_with_message_protection_basic128_service_policy
sts/wss11_kerberos_token_with_message_protection_service_policy

sts/wss11_sts_issued_saml_hok_with_message_protection_service_policy
sts/wss_sts_issued_saml_bearer_token_over_ssl_service_policy

sts/wss10_message_protection_service_policy
sts/wss11_message_protection_service_policy

sts/wss_http_token_over_ssl_service_policy
sts/wss_http_token_service_policy
```

See Also: "Using and Managing WSS Policies for Oracle WSM Agents"

Task overview: Using and modifying WS-S policies

1. From the Oracle Access Manager Console, System Configuration tab, open the Security Token Services section.
2. From the Endpoints node, proceed as described in "Managing Oracle Security Token Service Endpoints" on page 19-29 to locate or create the endpoint to be protected.
3. From the Policy URI list, choose a specific WS Security policy to protect the endpoint, as described in:
   - Managing WSS Policies for Oracle Security Token Service: Classpath
   - Managing WSS Policies for Oracle Security Token Service: Oracle WSM Policy Manager

17.3 Enabling and Disabling Oracle Security Token Service

This topic includes the following topics:

- About Oracle Security Token Service and the Oracle Access Manager Console
- About Enabling Services for Oracle Security Token Service
- Enabling and Disabling Services for Oracle Security Token Service

17.3.1 About Oracle Security Token Service and the Oracle Access Manager Console

Elements in the Oracle Access Manager Console enable administrators to easily configure the Token Service to exchange WS Trust tokens with partners. Token Service elements provide for creation, viewing, modification, and removal of partners, endpoints, validation templates, issuance templates, and data store connections.

All Oracle Security Token Service system configuration is done using the Oracle Access Manager Console. This includes the following common tasks covered in Part II of this book:

- Registering and managing common OAM Servers and proxy information
- Registering and managing the common Default User Identity Store
- Configuring the OAM Keystore, which differs from the OWSM Keystore used for WSS processing
- Certificate Validation and Revocation

The Oracle Access Manager Console enables administrators to perform the following Oracle Security Token Service-specific tasks:

- Manage validation token templates: The validation templates include configuration properties to validate a Web Services Security/WSTrust token, and map it to a Requester Partner or a User record in the Default User Identity Store.
- Manage issuance templates: The issuance templates contain rules on how a token will be created
- Manage Partner Data: A partner represents a partner trusted by OSTS. OSTS defines three types of partners: Requester, Relying Party and Issuing Authority. Each partner entry is associated to a partner profile. The partner entry contains signing and encryption certificates and identifiers used to uniquely identify a partner

- Manage Partner Profile: A partner profile contains configuration properties that are common to a set of partners:
  - Claim Mapping
  - Token Types definition
  - Issuance and Validation templates defined for the token Types
  - Override Validation Template rules for Issuing Authorities(Other STS)
- Manage Oracle Security Token Service Endpoints
- Manage Token Issuance Policies (authorization policies that will be evaluated to determine if a Requester Partner can request a token based on the Relying Party referenced in the request)
- Oracle Security Token Service Global Settings
- Custom tokens

17.3.1.1 About Oracle Security Token Service Administrators

Users with administrative access to the Oracle Access Manager Console, have access to Oracle Security Token Services.

Initially, administrative users must log in to the Oracle Access Manager Console using the WebLogic Administrator credentials set during initial configuration. However, your enterprise might require independent sets of administrators: one set of users responsible for Oracle Access Manager and another for Oracle Security Token Service.

17.3.1.2 About Logging In To, and Signing Out Of, Oracle Security Token Service

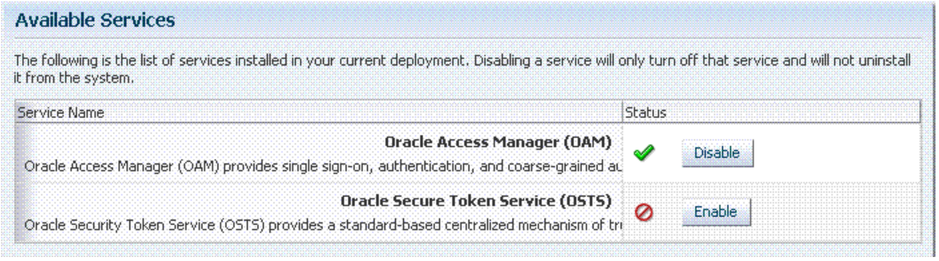
When using Oracle Security Token Service with Oracle Access Manager, logging in to, and signing out of the Oracle Access Manager Console is the same.

> **See Also:** Chapter 3 for the following topics:
>
> - Logging In to the Oracle Access Manager Console
> - Signing Out of Oracle Access Manager Console

17.3.2 About Enabling Services for Oracle Security Token Service

To use Oracle Security Token Service, both it and Oracle Access Manager must be enabled, as shown in Figure 17–3. By default Oracle Security Token Service is disabled and needs to be enabled.

Figure 17–3 Oracle Access Manager with Oracle Security Token Service Enabled



Available Services

The following is the list of services installed in your current deployment. Disabling a service will only turn off that service and will not uninstall it from the system.

Service Name	Status
Oracle Access Manager (OAM) Oracle Access Manager (OAM) provides single sign-on, authentication, and coarse-grained au	✓ Disable
Oracle Secure Token Service (OSTS) Oracle Security Token Service (OSTS) provides a standard-based centralized mechanism of tr	⊘ Enable

A green check mark in the Status field beside the service name indicates the service is enabled. A red circle with a line through it indicates that the corresponding service is disabled.

17.3.3 Enabling and Disabling Services for Oracle Security Token Service

Prerequisites

Oracle Access Manager service must be enabled.

To enable or disable Oracle Security Token Service

1. Log in to the Oracle Access Manager Console, as usual

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

2. From the System Configuration tab, Common Configuration section, click Available Services.
3. **Enable OSTS**: Beside Oracle Security Token Service, click Enable (or confirm that the Status check mark is green) and confirm that the Oracle Access Manager Service is also enabled.
4. **Disable OSTS**: Beside Oracle Security Token Service, click Disable (or confirm that the Status check mark is red).

17.4 Defining Security Token Service Settings Using Oracle Access Manager Console

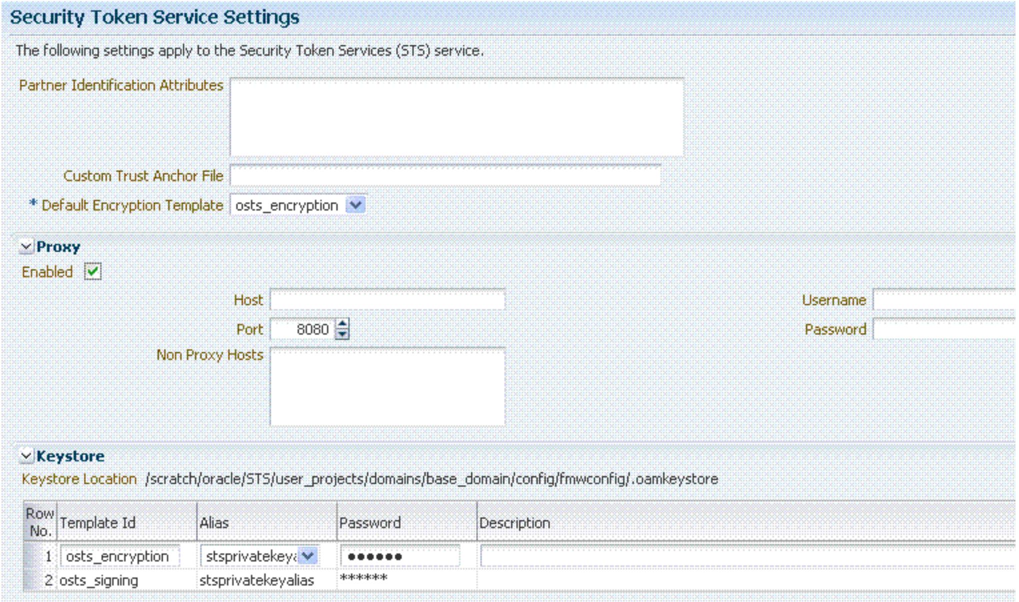
This section provides the following information:

- About Security Token Service Settings
- Managing Security Token Service Settings

17.4.1 About Security Token Service Settings

Security Token Service Settings can be viewed or altered from the Security Token Services section of the System Configuration tab. These settings are show in Figure 17–4.

Figure 17–4 Security Token Service Settings Page



Security Token Service Settings

The following settings apply to the Security Token Services (STS) service.

Partner Identification Attributes

Custom Trust Anchor File

* Default Encryption Template: osts_encryption

Proxy

Enabled ☑

Host

Port: 8080

Non Proxy Hosts

Username

Password

Keystore

Keystore Location /scratch/oracle/STS/user_projects/domains/base_domain/config/fmwconfig/.oamkeystore

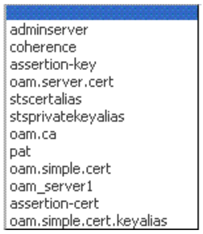
Row No.	Template Id	Alias	Password	Description
1	osts_encryption	stsprivatekeyː	●●●●●●	
2	osts_signing	stsprivatekeyalias	******	

Table 17–2 describes the elements on the Security Token Service Settings page.

Table 17–2 Security Token Service Settings

Element	Description
Partner Identification Attributes	A field where you list attributes, other than the standard ones available by default, that should be available in "Identity Attributes" Table in the Partner page. These attributes can be used to identify a partner by matching their values against those in the incoming request. When a Requester sends a WS-Trust request to Oracle Security Token Service, the server might map the incoming token containing the requester's identity to a partner entry in the Oracle Security Token Service partner store. To do so, Oracle Security Token Service will use the mapping settings configured in a validation template and will attempt to map the token data to a partner entry by performing a lookup by matching the token data to a Partner Identification Attribute. By default, each requester partner contains three identification attributes that can be set: username, HTTP Basic Username, SSL Client Certificate DN. It is possible to define additional Identification Attributes that could be set for each requester partner entry. This section allows new attributes to be set. After defining a new attribute, it becomes available in the Requester Partner entry section, and it can be used in mapping rules in the WSS Validation Templates.
Custom Trust Anchor File	By default, Oracle Access Manager and Oracle Security Token Service use the default `$DOMAIN_HOME/config/fmwconfig/amtruststore` keystore containing the trust anchors used for certificate validation by Oracle Security Token Service, when verifying X.509 Tokens, or when verifying certificates used in SAML Assertion signatures. It is possible to configure Oracle Security Token Service to use a specific trust anchor file if necessary, that will contain trust anchors only used for Oracle Security Token Service operations and validations. In this case, this field should contain the location of the JKS keystore to use. Note the following: ■ When using a custom trust anchor keystore, it will not be replicated automatically across the cluster. You must manage replication. ■ In most cases, the default Oracle Access Manager and Oracle Security Token Service trust anchor should be enough. See Also: Chapter 18, "Managing Oracle Security Token Service Certificates and Keys"
Default Encryption Template	A list from which you choose the default template for Oracle Security Token Service encryption: ■ osts_encryption ■ osts_signing See Also: Setting the Default Encryption Key on page 18-6.

Table 17–2 (Cont.) Security Token Service Settings

Element	Description
Proxy	Outbound Connection Properties, HTTP Proxy Settings Use this section to configure Oracle Security Token Service to use a proxy for outgoing HTTP connections when optionally retrieving the WS-Sec Policy of Relying Parties at runtime: ■ Enabled: When this box is checked the Proxy function is enabled and will be used when retrieving the WS-Security Policy of Relying Parties. When the box is not checked, the Proxy function is disabled and related fields are inaccessible for editing. ■ Host: The proxy hostname ■ Port: The proxy port number. Default is 8080 ■ Non Proxy Hosts: A list of hosts for which the proxy should not be used. Use ';' to separate multiple hosts. ■ Username: The username to use when connecting to the proxy. ■ Password: The password to use when connecting to the proxy.
Keystore	Location: Path of the active keystore that was set up during Oracle Security Token Service installation. The Keystore table includes the following information for each of the templates in the table, which are available for use as the Default Encryption Template: ■ Template ID: The name of the template that can access the keystore. ■ Alias: Identifies the alias for the template. When adding a template, you can choose from the Aliases listed. For example:  adminserver coherence assertion-key oam.server.cert stscertalias stsprivatekeyalias oam.ca pat oam.simple.cert oam_server1 assertion-cert oam.simple.cert.keyalias ■ Password: The password for the selected Alias. ■ Description: Optional. The keystore section defines key entries that exist in the `OAM/OSTS Keystore ($DOMAIN_HOME/config/fmwconfig/.oamkeystore` and of type JCEKS) After an entry is defined an entry, it can be used in other Oracle Security Token Service templates (like SAML Issuance Templates).

17.4.2 Managing Security Token Service Settings

Users with valid administrator credentials can use this procedure to confirm or alter Security Token Service Settings.

Prerequisites

Both the Oracle Access Manager Service and the Oracle Security Token Service must be enabled.

To view or edit Security Token Service Settings

1. Log in to the Oracle Access Manager Console, as usual.

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

2. From the System Configuration tab, open the Security Token Service Settings section.
3. Double-click the Security Token Service Settings
4. On the Security Token Service Settings page view (or modify) the following information (see Table 17–2):
   - Partner Identification Attributes
   - Custom Trust Anchor File
   - Proxy details
5. Keystore Table: View, add, or remove new encryption templates
6. Click Apply to submit changes (or Revert to cancel changes).
7. Close the page when finished.

17.5 Using and Managing WSS Policies for Oracle WSM Agents

You can use existing Web Service Security policies to protect Oracle Security Token Service Web Service endpoints. For instance:

- classpath mode: Existing Web Service Security policies defined in `$ORACLE_IDM_HOME/oam/server/policy/sts-policies.jar` are used in this mode
- SOA deployment: Policies defined in the Oracle WSM Policy Manager available from a SOA deployment are used

This section describes how to manage Web Service Security Policies for Oracle Security Token Service in the following topics:

- Using and Modifying Web Service Security Policies
- Managing WSS Policies for Oracle Security Token Service: Classpath
- Managing WSS Policies for Oracle Security Token Service: Oracle WSM Policy Manager

17.5.1 Using and Modifying Web Service Security Policies

This section introduces WS-Security Policies used to protect Oracle Security Token Service WS Endpoint and how to modify these policies. TheWS-Security Policies that are provided by Oracle should cover most use cases.

See Also:

- "About Oracle Security Token Service End Points and Policies" on page 17-5
- Attaching Policies to Web Services in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*

17.5.2 Managing WSS Policies for Oracle Security Token Service: Classpath

Predefined Oracle Web Services Manager policies are constructed using assertions based on predefined assertion templates. For WSS Policy classpath mode, the OWSM Agent retrieves policies from sts-policies.jar located on the classpath.

If SOA is not deployed in the WebLogic Server domain, the Oracle Security Token Service installer configures the WebLogic Server domain for WSS Policy classpath mode. The JAR file containing the WSS Policies used when the WLS Domain is configured for classpath is located at:

```
$IDM_ORACLE_HOME/oam/server/policy/sts-policies.jar
```

When your environment is in classpath mode, perform the following tasks to Administrators confirm sts-policies.jar is located on the classpath.

See Also:

- "About Oracle Security Token Service End Points and Policies" on page 17-5
- Oracle WSM Predefined Policies and Assertion Templates in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*

Task overview: Managing WSS Policies for Oracle Security Token Service: Classpath

1. Define an OWSM Assertion Template.
2. Proceed as follows, depending on your need:
   - Modify an OWSM Policy
   - Define a Policy using the OWSM Assertion Template
3. Bundle the Assertion Template and policy in the sts-policies.jar file:

   ```
   META-INF/assertiontemplates/oracle of the $IDM_ORACLE_HOME/oam/server/policy/
   sts-policies.jar
   ```

4. Confirm that `sts-policies.jar` is located in the following path to enable the policy URI to be available the Policy URI drop down list.

   ```
   $IDM_ORACLE_HOME/oam/server/policy/sts-policies.jar
   ```

5. Restart the Managed Servers running Oracle Security Token Service.
6. Proceed to the OAM Admin Console to configure the Oracle Security Token Service Endpoints.

17.5.3 Managing WSS Policies for Oracle Security Token Service: Oracle WSM Policy Manager

The Oracle WSM Policy Manager is the security linchpin for Oracle Fusion Middleware Web services and SOA applications. For more information about how the Oracle WSM Policy Manager manages the policy framework, see "Understanding Oracle WSM Policy Framework" in Oracle Fusion Middleware Security and Administrator's Guide for Web Services.

At design time, you attach Oracle WSM and WebLogic Web service policies to applications programmatically using your favorite IDE, such as Oracle JDeveloper. Alternatively, at deployment time you attach policies to SOA composites, ADF, and WebCenter applications using the Oracle Enterprise Manager Fusion Middleware Control, and to WebLogic Web services (Java EE) using the WebLogic Server Administration Console.

System administrators can leverage the Oracle WSM through the Oracle Enterprise Manager Fusion Middleware Control to:

- Centrally define policies using the Oracle WSM Policy Manager.
- Enforce Oracle WSM security and management polices locally at run time.

When your environment is integrated with the OWSM Policy Manager, perform the following tasks to add or modify WSS policies for Security Token Service Settings using Oracle Web Services Manager.

> **Note:** All of Oracle WSM's functionality is accessible to administrators from Oracle Enterprise Manager Fusion Middleware Control.

> **See Also:** Oracle Fusion Middleware Security and Administrator's Guide for Web Services
>
> - Part II, "Basic Administration"
> - Part III, "Advanced Administration"

Task overview: Managing WSS Policies for Oracle Security Token Service: OWSM Policy Manager

1. From the OWSM Policy Manager, locate and open the desired policy.
2. Refer to the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* and make any required changes to the policy.
3. Restart all Managed Servers running Oracle Security Token Service.
4. Proceed to "Configuring OWSM for WSS Protocol Communication".

17.6 Configuring OWSM for WSS Protocol Communication

This section describes how to configure communication between WS-Sec Clients and the Oracle WSM Agent embedded with Oracle Security Token Service.

The Oracle WSM Agent protects the Web Service endpoints of Oracle Security Token Service, and provides support for WSS protocol exchanges. To ensure a client is communicating successfully with the Oracle WSM Agent:

- The client might need to be aware of the signing and encryption certificates used by the Oracle WSM Agent (this will require extracting and distributing the signing and encryption certificates used by the OWSM Agent embedded with Oracle Security Token Service).
- The Oracle WSM Agent might need to be aware, depending on the policies, of the signing certificate used by the client (this will require adding the client's certificate as a trusted certificate for the Oracle WSM Agent)

Task overview: Configuring communication with Oracle WSM agents

1. See "About Oracle WSM Agent WS-Security Policies for Oracle Security Token Service"
2. Retrieving the Oracle WSM Keystore Password
3. Extracting the Oracle STS/Oracle WSM Signing and Encryption Certificate
4. Adding Trusted Certificates to the Oracle WSM Keystore
5. Validating Trusted Certificates in the Oracle WSM Keystore

> **See Also:** Chapter 18, "Managing Oracle Security Token Service Certificates and Keys"

17.6.1 About Oracle WSM Agent WS-Security Policies for Oracle Security Token Service

The Oracle WSM Agent requires a repository to retrieve the Web Services Security (WSS) policies it needs. Oracle Access Manager supports two types of repositories for Oracle Security Token Service:

- JAR file with WSS Policies: Used when the WLS Domain is configured for classpath. The required JAR file is located in $IDM_ORACLE_HOME/oam/server/policy/ sts-policies.jar.
- Oracle WSM Policy Manager available from the SOA deployment

During Oracle Security Token Service installation, the installer detects if the Oracle Web Services Manager Policy Manager is present and deployed in the WebLogic Security domain.

- If not deployed in the WebLogic Security domain, the installer configures the WebLogic Security domain for the Web Services Security Policy classpath mode, where the WSM Agent will retrieve the policies from a JAR file.
- If present, the installer connects to the Oracle Web Services Manager Policy Manager and uploads the policies that are used to protect Oracle Security Token Service endpoints.

> **See Also:** "About the Policy and Session Database Store" on page 5-4 for details about the OAM 11g required database for OAM policy data and (optionally) OAM user session data

17.6.2 Retrieving the Oracle WSM Keystore Password

Administrators need to retrieve the keystore password and key entry password from CSF for certain activities. Otherwise, keystore or key entry cannot be changed. Having access to the keystore is sometimes required to:

- Extract the signing/encryption certificate to distribute to clients if necessary
- Update or replace the signing/encryption key entry
- Add trusted certificates

The following procedure displays the password used to protect the Oracle WSM keystore as well as the key entry.

To retrieve the Oracle WSM keystore password

1. Enter the WSLT scripting environment.
2. Connect to the WebLogic Server AdminServer, using the `connect()` command.
3. Execute the following command by providing the connection information to the AdminServer: `listCred(map="OAM_STORE", key="jks")`.
4. Note the password.
5. Proceed to "Extracting the Oracle STS/Oracle WSM Signing and Encryption Certificate".

17.6.3 Extracting the Oracle STS/Oracle WSM Signing and Encryption Certificate

During SOAP interactions, the WS-Security protocol might require the client to trust the signing/encryption certificate used for WSS operations by the OWSM Agent protecting the OSTS endpoint. In those cases, the OSTS administrator should extract the OSTS OWSM signing/encryption certificate used for WSS operations and provide it to the WS Client.

The administrator must export the signing and encryption certificate used by Oracle Security Token Service for WSS cryptographic operations. The following procedure guides as you do this by:

- Replacing $DOMAIN_HOME with the path to the Domain directory
- CERT_FILE with the location of the file where the certificate will be saved

If you are prompted to enter a password, simply press the Enter key.

Prerequisites

Retrieving the Oracle WSM Keystore Password

To export the signing and encryption certificate

1. Locate keytool.
2. Execute the following command.

   ```
   keytool -exportcert -keystore $DOMAIN_HOME/config/fmwconfig/default-keystore.j
   ks -storetype JKS -alias orakey -file $CERT_FILE
   ```

3. Enter the keystore password retrieved in the previous section if prompted.
4. Proceed to "Adding Trusted Certificates to the Oracle WSM Keystore".

17.6.4 Adding Trusted Certificates to the Oracle WSM Keystore

To add a trusted certificate to the OWSM keystore for WSS cryptographic operations:

- perform the command in the following procedure
- replace the $DOMAIN_HOME with the path to the Domain directory

- replace the $TRUSTED_CERT_FILE with the location of the file containing the trusted certificate
- replace the TRUSTED_CERT_ALIAS with the alias under which the trusted certificate will be stored

When prompted to enter a password, enter the password of the OWSM keystore that you retrieved earlier.

Prerequisites

Retrieving the Oracle WSM Keystore Password

The administrator must have the certificate to import.

To add trusted certificates to the Oracle WSM keystore

1. Locate keytool.
2. Execute the following command.

```
keytool -importcert -trustcacerts -keystore $DOMAIN_HOME/config/fmwconfig/de
fault-keystore.jks -storetype JKS -alias $TRUSTED_CERT_ALIAS -file $TRUSTED_
CERT_ALIAS
```

3. Observe messages on the screen, enter a password if requested.
4. Proceed to "Validating Trusted Certificates in the Oracle WSM Keystore".

17.6.5 Validating Trusted Certificates in the Oracle WSM Keystore

When the Oracle WSM Agent performs a certificate validation, it uses the keystore configured for Oracle WSM tasks, and will validate the certificate against the trusted certificate entries contained in the keystore. For those operations, it might be required to add trusted certificate entries (the certificate itself or the issuer's certificate) in the OWSM keystore.

When receiving a SOAP requester, the Oracle WSM Agent processes the request for message protection. Part of the steps might include a certificate validation operation if the incoming message:

- is of type WSS 1.0, and includes a digital signature created with a private key, without the certificate being present. In this case:

  Remedy: The Oracle WSM keystore must contain the signing certificate.

- is of type WSS 1.0, and includes a digital signature created with a private key, with the certificate being present.

  Remedy: The Oracle WSM keystore must contain either the signing certificate or the issuer's certificate of the signing certificate.

- is of type WSS 1.1, and includes a digital signature created with a private key, without the certificate being present.

  Remedy: The Oracle WSM keystore must contain the signing certificate.

- is of type WSS 1.1, and includes a digital signature created with a private key, with the certificate being present. In this case, the OWSM keystore will need to contain either the signing certificate or the issuer's certificate of the signing certificate

  Remedy: The Oracle WSM keystore must contain either the signing certificate or the issuer's certificate of the signing certificate

See Also: Chapter 18, "Managing Oracle Security Token Service Certificates and Keys"

17.7 Managing and Migrating Oracle Security Token Service Policies

This section provides the following topics:

- About Managing and Migrating Oracle Security Token Service Policies
- Managing Oracle Security Token Service Policies
- Migrating Oracle Security Token Service Policies

17.7.1 About Managing and Migrating Oracle Security Token Service Policies

Oracle Security Token Service packages policies for endpoints in sts-policies.jar file. This jar is copied to following location under WLS_HOME ($WL_HOME/Oracle_IDM1, for example):

```
$DW_HOME/oam/server/policy
```

The sts-policies.jar contains the stspolicies.prop file at the following location in the JAR:

```
META-INF/policies/sts/
```

This file lists all the policies packaged in the directory as file names to allow the server to read the JAR entries programatically when migrating policies to destination repository.

Note: Be sure to update policies and stspolicies.prop as needed before migration.

17.7.2 Managing Oracle Security Token Service Policies

The following procedure outlines the various scenarios for policy updates.

Task overview: Updating policies and stspolicies.prop

1. Add a Policy to sts-policies jar: Before creating the new jar, you must also update the stspolicies.prop file at META-INF/policies/sts/ to include this new policy file name.
2. Delete a Policy from sts-policies jar: You must also delete the entry from file META-INF/policies/sts/stspolicies.prop.
3. Update Existing Policy File Name: When re-naming a policy file at META-INF/policies/sts/, you must also update the corresponding entry in the file META-INF/policies/sts/stspolicies.prop file.
4. Update Existing Policy Content: When updating the content of a policy file, without touching the file name, there is no need to do anything else.

17.7.3 Migrating Oracle Security Token Service Policies

During installation a check is performed to establish whether SOA is deployed within the domain where Oracle Security Token Service is being installed:

- If SOA is not installed, the Oracle WSM protocol is set to classpath and policies are read from the JAR on the class path.

See Also: "Using and Managing WSS Policies for Oracle WSM Agents" on page 17-13.

- If SOA is present within the domain, Oracle Security Token Service reads the policies from sts-policies.jar and migrates them to the Oracle WSM PM repository by calling Oracle WSM Mbeans.
- If SOA is installed after Oracle Security Token Service within the same domain, ensure smooth operations between SOA and Oracle Security Token Service as follows:
  - The Oracle WSM protocol must be set to 'remote'.
  - Oracle Security Token Service policies from sts-policies jar must be migrated to Oracle WSM PM repository using Oracle WSM provided tools.

17.8 Introduction to Logging Oracle Security Token Service Messages

Logging is the mechanism by which components write messages to a file. Administrators can use the logging mechanism to capture critical 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.

Configuring logging and locating log files are the focus of this section. Diagnosing problems using the information in log files is outside the scope of this manual.

Log messages are used for problem diagnosis. The logging infrastructure records messages from Oracle Access Manager components. The administrator controls the amount of information that is logged in a message by specifying log levels for each Oracle Access Manager 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 Oracle Access Manager 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 with Oracle Security Token Service 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

17.9 Introduction to Auditing for Oracle Security Token Service

In Oracle Access Manager with Oracle Security Token Service, 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 section introduces the Oracle Security Token Service administrative and run-time events that can be audited. Configuring common auditing settings for Oracle Security Token Service and validating your auditing configuration is a subject of this chapter. However, analyzing and using audit data is outside the scope of this book.

Oracle Access Manager with Oracle Security Token Service uses the Oracle Fusion Middleware Common Audit Framework to support auditing for a number of user 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).

Auditing for Oracle Access Manager with Oracle Security Token Service is based on configuration parameters set in the Oracle Access Manager Console that enable data capture for a user or set of users.

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 with Oracle Security Token Service integrates with Oracle Business Intelligence Publisher, which provides a pre-defined set of compliance reports.

17.9.1 About Oracle Security Token Service Audit Record Storage

Oracle Access Manager with Oracle Security Token Service can be configured to write audit records to a variety of targets supported by the Common Audit Framework:

- Local flat files: By default, Oracle Access Manager with Oracle Security Token Service records audit data to a file.
- Central database: 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.
- Platform-specific log (Linux Syslog and Windows Event Log)
- Audit Vault

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

See Also:

- Oracle Fusion Middleware Application Security Guide
- "Setting Up the Audit Database Store" on page 24-11
- "Adding, Viewing, or Editing Common Audit Settings within Oracle Access Manager" on page 24-14

17.9.2 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.

See Also: "About Audit Reports and Oracle Business Intelligence Publisher" on page 24-4

17.9.3 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

17.10 Auditing Oracle Security Token Service Administrative and Run-time Events

Oracle Security Token Service provides an independent audit configuration file, component_events.xml, that defines specific event types and events to audit.

This section provides the following topics:

- About Audit Record Content Common to All Events
- Oracle Security Token Service Administrative Events You Can Audit
- Oracle Security Token Service Run-time Events You Can Audit

17.10.1 About Audit Record Content Common to All Events

The following data is part of each audit record, regardless of the event or event type that is audited:

- Date and time of event
- IP address of the client initiating event
- Client identity
- Processing time for the event

17.10.2 Oracle Security Token Service Administrative Events You Can Audit

Oracle Security Token Service administrative events fall into several configuration management operations defined in component_events.xml. See details in Table 17–3.

> **See Also:** "Setting Up Auditing for Oracle Access Manager with Oracle Security Token Service" on page 24-11

Table 17–3 Oracle Security Token Service Configuration Management Operations

OSTS Configuration Management Operations	Description
Common Attributes	■ OldSettings: The string representing the previous settings before the change was applied. ■ NewSettings: The string representing the new settings. ■ TemplateID: The ID of the Validation or Issuance Template being created or updated or deleted. ■ ProfileID: The ID of the Partner Profile being created or updated or deleted. ■ PartnerID: The ID of the Partner being created or updated or deleted. ■ SettingsID: The ID of the generic settings being created or updated or deleted.
Create Validation Template	Audit event recorded for the creation of a Validation Template referenced by CreateValidationTemplate. Attributes: ■ TemplateID ■ NewSettings
Update Validation Template	Audit event recorded for the update of a Validation Template referenced by UpdateValidationTemplate. Attributes: ■ TemplateID ■ OldSettings ■ NewSettings
Delete Validation Template	Audit event recorded for the delete event of a Validation Template referenced by DeleteValidationTemplate. Attributes: ■ TemplateID ■ OldSettings
Create Issuance Template	Audit event recorded for the creation of an Issuance Template referenced by CreateIssuanceTemplate. Attributes: ■ TemplateID ■ NewSettings

Table 17–3 (Cont.) Oracle Security Token Service Configuration Management

OSTS Configuration Management Operations	Description
Update Issuance Template	Audit event recorded for the update of an Issuance Template referenced by UpdateIssuanceTemplate. Attributes: ▪ TemplateID ▪ OldSettings ▪ NewSettings
Delete Issuance Template	Audit event recorded for the delete event of an Issuance Template referenced by DeleteIssuanceTemplate. Attributes: ▪ TemplateID ▪ OldSettings
Create Partner Profile	Audit event recorded for the creation of Partner Profile referenced by CreatePartnerProfile. Attributes: ▪ ProfileID ▪ NewSettings
Update Partner Profile	Audit event recorded for the update of a Partner Profile referenced by UpdatePartnerProfile. Attributes: ▪ ProfileID ▪ OldSettings ▪ NewSettings
Delete Partner Profile	Audit event recorded for the delete event of Partner Profile referenced by DeletePartnerProfile. Attributes: ▪ ProfileID ▪ OldSettings
Create Partner	Audit event recorded for the creation of Partner Profile referenced by CreatePartner. Attributes: ▪ PartnerID ▪ NewSettings
Update Partner	Audit event recorded for the update of a Partner Profile referenced by UpdatePartner. Attributes: ▪ PartnerID ▪ OldSettings ▪ NewSettings
Delete Partner	Audit event recorded for the delete event of Partner Profile referenced by DeletePartner. Attributes: ▪ PartnerID ▪ OldSettings

Table 17–3 (Cont.) Oracle Security Token Service Configuration Management

OSTS Configuration Management Operations	Description
Generic Admin Creation	Audit event recorded for the generic create administrative operation referenced by GenericAdminCreation. Attributes: ■ SettingsID ■ NewSettings
Generic Admin Update	Audit event recorded for the update of a generic update administrative operation referenced by GenericAdminUpdate. Attributes: ■ SettingsID ■ OldSettings ■ NewSettings
Generic Admin Removal	Audit event recorded for generic delete administrative operation referenced by GenericAdminDeletion. Attributes: ■ SettingsID ■ OldSettings

17.10.3 Oracle Security Token Service Run-time Events You Can Audit

Oracle Security Token Service-specific run-time events for token operations are defined in component_events.xml. See details in Table 17–4.

Table 17–4 Oracle Security Token Service-specific Run-time Events

Token Operations	Description
Common Attributes	■ Requester: Who made the request by sending the RST ■ RelyingParty: The one for whom the token is created ■ UserID: End user identity ■ TokenType: Either SAML11, SAML20, Username, X.509, Kerberos, OAM or Custom ■ Token: The XML value of the token ■ TokenContext: The Context data passed for token operations ■ Message: The XML representation of the incoming or outgoing message
Incoming Message	Incoming RSTR message received by Oracle Security Token Service referenced by OutgoingMessage. Attributes populated for this event, if available: ■ Requester ■ RelyingParty ■ Message

Table 17–4 (Cont.) Oracle Security Token Service-specific Run-time Events

Token Operations	Description
Outgoing Message	Outgoing RSTR message received by Oracle Security Token Service referenced by IncomingMessage. Attributes populated for this event, if available: ■ Requester ■ RelyingParty ■ Message
Token Validation	Audit event for token validation in Oracle Security Token Service referenced by TokenValidation. The status attribute indicates whether or not the validation operation was successful. Attributes populated for this event, if available: ■ Requester ■ RelyingParty ■ Token ■ TokenType ■ TokenContext ■ Status
Token Generation	Audit event for token generation in Oracle Security Token Service referenced by TokenGeneration. Attributes populated for this event, if available: ■ Requester ■ RelyingParty ■ Token ■ TokenType ■ TokenContext ■ UserID
LDAP User Authentication	Audit event for local user authentication with the LDAP Directory referenced by LDAPUserAuthentication. Attributes populated for this event, if available: ■ UserID ■ Status
Generic Runtime Operation	Audit event for a generic operation performed by Oracle Security Token Service referenced by GenericRuntimeOperation Attributes populated for this event, if available: ■ OperationType: type of operation ■ OperationData: string representing context of the operation

18

Managing Oracle Security Token Service Certificates and Keys

This chapter provides the following sections:

- Prerequisites
- Introduction to Certificates and Keys for Oracle Security Token Service
- Managing Oracle Security Token Service Encryption/Signing Keys
- Managing Partner Keys for WS-Trust Communications
- Managing Certificate Validation

18.1 Prerequisites

Both the Oracle Access Manager and Oracle Security Token Service services must be running, as described in "Enabling and Disabling Oracle Security Token Service" on page 17-8.

18.2 Introduction to Certificates and Keys for Oracle Security Token Service

This section describes certificates and keys used for Oracle Security Token Service.

Depending on the public key infrastructure, the digital certificate establishes credentials for Web-based transactions, as described in "About Certificates, Authorities, and Encryption Keys" on page E-3.

Public Keys at Run Time: There are distinct cases where public key infrastructure materials are used at run time. For instance, during Web Services Security (WSS) protocol communication between Requesters and Oracle Security Token Service (with OWSM Agent). See also Table 18–1.

Table 18–1 OSTS Public Keys Used at Run Time

When Oracle Security Token Service ...	Description
Issues SAML Assertions	■ Oracle Security Token Service Signing Assertions using a key defined in the STS Global settings ■ Oracle Security Token Service using the Requester's signing certificate as a proof key for Holder-of-Key of type Public Key confirmation method ■ Oracle Security Token Service using the Relying Party's encryption certificate to encrypt the secret proof key for Holder-of-Key of type Secret Key confirmation method ■ Oracle Security Token Service using the Requester's encryption certificate to encrypt a secret proof/entry in the RSTR for Holder-of-Key of type Secret Key confirmation method
Issues tokens	■ Oracle Security Token Service uses the Relying Party's encryption certificate to encrypt the outgoing token
Validates SAML Assertions	■ Oracle Security Token Service uses the Issuing Authority's signing certificate to verify the signature of the incoming SAML Assertion
Uses Web Services Security (WSS) protocol communication	Between Requesters and Oracle Security Token Service (with OWSM Agent)

18.2.1 About Keystores and Oracle Security Token Service

Following is a brief summary of several types of keystores for Oracle Access Manager with Oracle Security Token Service:

Table 18–2 describes these in greater detail. Additional details follow the table.

Table 18–2 Keystores for Oracle Access Manager with Oracle Security Token Service

Keystore	Description
System Keystore	The container for keys and certificates associated with OAM Server instances (OAM secret keys and Oracle Security Token Service private keys for signing and encryption). Only one System Keystore of type JCEKS may be present: .oamkeystore. `$DOMAIN_HOME/config/fmwconfig/.oamkeystore` The certificate alias and password can be configured using the Oracle Access Manager Console, as described in Table 17–2, " Security Token Service Settings" on page 17-11.
Trust Keystore	The Trust Keystore is used to validate certificates presented by clients. `$DOMAIN_HOME/config/fmwconfig/amtruststore` amtruststore is created during installation, and must include at least one trusted anchor. The Trust Keystore is managed by using the JRE's keytool application. Oracle Security Token Service can use a custom trust keystore.

Table 18–2 (Cont.) Keystores for Oracle Access Manager with Oracle Security Token

Keystore	Description
Certificate Revocation Lists (CRL)	Certificate revocation information lists are stored in a ZIP archive on the filesystem. These are used by the Oracle Access Manager/Oracle Security Token Service server instances when performing CRL-based certificate revocation checking. amcrl.jar contains CRL files in the DER format: $DOMAIN_HOME/config/fmwconfig/amcrl.jar The OAM Server defines a notification listener for the Keystores and the CRL Zip file. Any changes to these files causes Oracle Access Manager with Oracle Security Token Service to reload the keystore/crl-zip at runtime, without requiring any restarts. amcrl.jar is created by installation and can be modified using the Oracle Access Manager Console, as described in "Managing Certificate Revocation Lists (CLRs)" on page 4-7.

The files in Table 18–3, are distributed across all Managed Servers in the domain by the JMX framework. The $DOMAIN_HOME/config/fmwconfig /mbeans directory defines a registration mbeans.xml for each file that indicates the MBean to manage the file and also identify that the file should be propagated across the domain.

Table 18–3 Keystore Mbeans

Keystore	Mbean and Description
System Keystore: .oamkeystore	Configuration of the .oamkeystore is done using the JRE's keytool application.
Trust Keystore: .amtruststore	Configuration of the amtruststore is done using the JRE's keytool application.
CRL: amcrl.jar	CRL MBean: Can be used to manage CRLs.

The token security key pair is populated to the common keystore shared by Oracle Access Manager and Oracle Security Token Service. This eliminates the need for Oracle Web Services Manager agents to interact with the common keystore.

You can use a WLST command to retrieve the password for keystores and for the amtruststore, as described in "Retrieving the System Keystore (.oamkeystore) Password" on page 18-5.

18.2.2 About the Oracle Web Services Manager Keystore (default-keystore.jks)

This topic describes the keystore of type JKS required by the Oracle WSM Agent to contain System and Partner keys and certificates.

Oracle WSM Agent functionality is available to Oracle Security Token Service to publish WS Policies and enforce message protection on inbound and outbound WS messages. Oracle WSM requires a separate keystore to contain System and Partner keys and certificates.

The Oracle WSM Agent uses a keystore for various cryptographic operations. For these tasks, the Oracle Web Services Manager Agent uses the keystore configured for Oracle Web Services Manager tasks (containing OWSM private keys and OWSM trusted certificates). The OPSS modules publish a keystore service used by Oracle Web Services Manager for certificate validation operations, and the $DOMAIN_HOME/config/fmwconfig/jps-config.xml will contain the settings for the keystore service. The default name is default-keystore.jks, which is specified in jps-config.xml.

Oracle strongly recommends that the Oracle WSM Agent keystore and the Oracle Access Manager and the Oracle Security Token Service keystore always be different. Otherwise, keys could be available to any modules authorized by OPSS to access the keystore and Oracle Access Manager keys might be accessed.

> **Note:** Oracle strongly recommends that the Oracle WSM Agent keystore and the OAM/OSTS keystore always be different.

During installation, if the Oracle WSM keystore service has not been configured, the installer:

- creates a new keystore in the $DOMAIN_HOME/config/fmwconfig folder (default name is default-keystore.jks)
- creates a key entry with the corresponding certificate that will be used by OWSM for signature and encryption operations. This key entry will be stored in the OWSM Keystore under the `orakey` alias
- stores the passwords of the key entry and of the keystore in CSF

Having access to the keystore is sometimes required, to:

- extract the signing/encryption certificate to distribute to clients if necessary
- update or replace the signing/encryption key entry
- add trusted certificates

See Also:

- "Configuring OWSM for WSS Protocol Communication" on page 17-15

18.2.3 About Using the OPSS Keystore for Requester Certificates

For the special cases where clients use referencing schemes such as SKI etc (as opposed to certificate token being received as part of the web service request), the requester's certificates need to be populated in the OPSS Keystore. This is an uncommon scenario that requires manually provisioning keys to the OPSS keystore

For more information on this, see "About Agents and Oracle Security Token Service" on page 17-4.

18.3 Managing Oracle Security Token Service Encryption/Signing Keys

Oracle Security Token Service uses keys to:

- Sign outgoing Assertions
- Decrypt any incoming XML encrypted data contained inside the RST message (tokens, entropies...), which is not handled by the WSS Protocol

Oracle Security Token Service uses the following keystore for storing Encryption and Signing Certificates.

```
DOMAIN_HOME/config/fmwconfig/.oamkeystore
```

Task overview: Managing Oracle Security Token Service Keys

1. Retrieving the System Keystore (.oamkeystore) Password
2. Adding a New Key Entry to the System Keystore (.oamkeystore)

3. Extracting an Oracle Security Token Service Certificate

   See Also: "Configuring OWSM for WSS Protocol Communication" on page 17-15

18.3.1 Retrieving the System Keystore (.oamkeystore) Password

The following procedure can be used to display the password that protects the keystore as well as the key entry.

If the keystore was created and configured by the IM/OAM/OSTS installer, then the keystore password and the key entry password will need to be retrieved from CSF. Otherwise, the administrator cannot change the keystore or key entry.

To retrieve the system keystore (.oamkeystore) password

1. Enter the WSLT scripting environment.
2. Connect to the WebLogic Server AdminServer, using the `connect()` command.
3. Execute the following command by providing the connection information to the AdminServer: `listCred(map="OAM_STORE", key="jks").`
4. Note the password.

18.3.2 Adding a New Key Entry to the System Keystore (.oamkeystore)

An administrator can use the following procedure to add a new key entry into the System keystore(.oamkeystore) using the keytool command to create and add the new key entry. Once the entry has been added, it must be defined in the Oracle Security Token Service configuration screen so that it can be used to sign assertions and decrypt incoming messages.

This topic provides the following procedures to add a new entry to sign SAML Assertions or decrypt XML-Encrypted data not covered by WSS:

- Adding a New Entry
- Configuring a SAML Issuance Template to use a Signing Key
- Setting the Default Encryption Key

18.3.2.1 Adding a New Entry

Prerequisites

Retrieving the System Keystore (.oamkeystore) Password

To configure a new entry

1. Locate keytool.
2. Execute the following command.

```
keytool -importcert -trustcacerts -keystore $DOMAIN_HOME/config/fmwconfig/de
fault-keystore.jks -storetype JKS -alias $TRUSTED_CERT_ALIAS -file $TRUSTED_
CERT_ALIAS
```

3. Observe messages on the screen.
4. Proceed as needed:
   - "Configuring a SAML Issuance Template to use a Signing Key", if needed

- "Setting the Default Encryption Key", if needed

18.3.2.2 Configuring a SAML Issuance Template to use a Signing Key

Users with valid Administrator credentials can use this procedure as a guide when editing an existing template to use a signing key.

See Also:

- About Managing Token Issuance Templates
- Searching for an Existing Template

To configure a SAML Issuance Template to use a signing key

1. Display the list of existing Token Issuance Templates.

   Oracle Access Manager Console
   System Configuration
   Security Token Services
   Token Issuance Templates

2. Find and open the SAML issuance template that will use the new key. For example: `saml11-issuance-template`.
3. On the SAML Issuance Template page, click the Security tab.
4. On the Security tab, Signing And Encryption section, click Sign Assertion.
5. From the Signing Keystore Access Template Id list, choose the KeyID as the Signing Keystore Entry.
6. Click Apply at the top of the page to save this information.
7. Proceed to "Setting the Default Encryption Key", if needed.

18.3.2.3 Setting the Default Encryption Key

Users with valid Administrator credentials can use this procedure as a guide when editing an existing template to use a signing key.

See Also: "About Security Token Service Settings" on page 17-10

To set the default encryption key

1. Go to the Security Token Service Settings page.

   Oracle Access Manager Console
   System Configuration
   Security Token Service
   Security Token Service Settings

2. From the Default Encryption Template list, select the new key entry.
3. Click Apply at the top of the page to save this information.
4. Proceed to "Setting the Default Encryption Key".

18.3.3 Extracting an Oracle Security Token Service Certificate

In some cases, it is required to distribute the Oracle Security Token Service keys used for SAML Signature operations or XML encryption operations:

- When a Relying Party needs to have access to the Oracle Security Token Service signing key, in order to validate the SAML Assertion issued by Oracle Security Token Service
- When a token needs to be encrypted for Oracle Security Token Service Server

To distribute the certificate of a key entry used by Oracle Security Token Service for SAML Signature operations or XML encryption operations, use the Certificate Retrieval Service by specifying the KeyID (listed in System Configuration > Security Token Services > Security Token Service Settings section) and the preferred encoding (der vs pem). For more information, see "Using the Certificate Retrieval Service".

18.3.3.1 Using the Certificate Retrieval Service

To use the Certificate Retrieval service

1. Retrieve the KeyID of the entry for which the certificate should be retrieved (listed in Oracle Access Manager Console System Configuration tab, Security Token Services section, Security Token Service Settings).
2. Create a URL. For example: http(s)://*osts-hostname:osts-port*/sts/servlet/samlcert?id=<KEYID>&encoding=<ENCODING>, with:
   - id holding the KeyID of the entry
   - encoding representing the format with which the certificate will be returned. Possible values are pem (PEM format) or der (DER format). (optional, default value is pem)
3. Review the certificate returned in the browser.

18.4 Managing Partner Keys for WS-Trust Communications

This topic provides the following information:

- About Partner Certificates
- About Downloading the Relying Party's Certificate at Run Time
- Setting the Partner's Signing or Encryption Certificate

18.4.1 About Partner Certificates

During the processing of the WS-Trust messages, Oracle Security Token Service might need to use a partner's certificate. The certificate needed depends on the situation, as described in Table 18–4.

Table 18–4 Partner Keys for WS-Trust Communications

If the OSTS Server Must Issue a ...	The Server ...
Issue a SAML Assertion encrypted for the Relying Party	Uses the Relying Party's encryption certificate to encrypt the outgoing token
Issue a SAML Assertion with the Subject Confirmation being of type Holder of Key / Asymmetric	Uses the Requester Partner's signing certificate as the proof key to be included in the Assertion Note: if the WS-Trust RST contains a UseKey element referencing an X.509 Binary Security Token in the SOAP header that was used in a signature, then Oracle Security Token Service will be able to use this certificate as the proof key.
Issue a SAML Assertion with the Subject Confirmation being of type Holder of Key / Symmetric	Uses the Relying Party's encryption certificate to encrypt the secret proof key to be included in the Assertion.
Issue a SAML Assertion with the Subject Confirmation being of type Holder of Key / Symmetric	Can encrypt in the RSTR for the Requester, the secret or the server entropy. In this case, the server: ■ uses the Requester's encryption certificate to encrypt the secret (if the secret was generated using only server entropy) ■ or uses the server entropy to encrypt the secret in the RSTR (if the secret was derived from client and server entropy). Note: if the WS-Trust RST contains a ProofEncryption element referencing an X.509 Binary Security Token in the SOAP header that was used in a signature, then Oracle Security Token Service will be able to use this certificate to encrypt the secret or entropy returned to the client.
Validate an incoming SAML Assertion	Uses the Issuing Authority's signing certificate to verify the XML digital signature present on the Assertion.

18.4.2 About Downloading the Relying Party's Certificate at Run Time

At runtime, Oracle Security Token Service is capable of downloading the Relying Party WSS Policy of the service listed in the AppliesTo field of the RST. If Oracle Security Token Service is configured to download the Relying Party's WS-Sec policy, then ensure that the Proxy settings are correctly entered, if needed, so that Oracle Security Token Service can connect to the Relying Party.

If the Relying Party Partner Profile is configured to do so, it instructs Oracle Security Token Service to download the WS-Sec Policy from the service. Oracle Security Token Service then extracts the certificate located in the policy and uses it for cryptographic operations, if necessary. Also:

- If Oracle Security Token Service issues a SAML Assertion encrypted for the Relying Party, the server uses the certificate downloaded from the Relying Party's WS-Sec Policy to encrypt the outgoing token.
- If Oracle Security Token Service issues a SAML Assertion with the Subject Confirmation of type Holder of Key / Symmetric, Oracle Security Token Service uses the certificate downloaded from the Relying Party's WS-Sec Policy to encrypt the secret proof key to be included in the Assertion.

To configure the Relying Party Partner Profile to download the certificate at run time, see "Setting the Partner's Signing or Encryption Certificate".

18.4.3 Setting the Partner's Signing or Encryption Certificate

To set the signing or encryption certificate of a partner, perform the following operations.

Alternatively: Use the WLST Partner commands to set the signing or encryption certificate of a specific partner.

Prerequisites

Review Table 18–4, " Partner Keys for WS-Trust Communications"

To set the certificate of a partner

1. From the Oracle Access Manager Console System Configuration, tab, Security Token Services section, and expand the Partners node.
2. Within the Partners node, expand Requester (or Relying Party or IssuingAuthority (see Table 18–4)).
3. Search for and open (or Create) the Partner for which the certificate must be set.
4. Edit Partner settings as needed (see "Managing Token Service Partners" on page 20-2) and click Save.
5. **Encryption Certificate**: Click the Browse button to locate and choose the Encryption certificate.
6. **Signing Certificate**: Click the Browse button to locate and choose the Signing certificate.
7. Save the information and close the page.
8. Proceed with "Managing Certificate Validation".

18.5 Managing Certificate Validation

This section describes managing certificate validation. Conditions for certificate validation are described in Table 18–5.

Table 18–5 Conditions for Oracle Security Token Service Certificate Validation

OSTS Validates a Certificate When ...
The security token to be validated is one of the following types: ■ X.509 ■ X.509v3 ■ PKCS#7
A SAML Assertion must be validated
OSTS is configured to validate the signing certificate of a SAML Issuing Authority

Successful validation requirements are listed in Table 18–6.

Table 18–6 Successful Certificate Validation Requirements

Certificates Must ...	How ...
Be linked to a trusted anchor:	■ by being a trusted anchor ■ or by having its issuer being a trusted anchor

Table 18–6 (Cont.) Successful Certificate Validation Requirements

Certificates Must ...	How ...
Not be revoked: - by being a trusted anchor - or by having its issuer being a trusted anchor	The revocation status of a certificate can be decided by checking: - Against a list of CRLs that were uploaded by the administrator - Against an OCSP server - CRL Distribution Points

Certificate validation requires the Trust Anchors Store (.amtruststore). Procedures for managing this store and validation are described in following topics:

- Retrieving the Trust Anchors Store (amtruststore) Password
- Managing the Trust Anchors Store (amtruststore)
- Managing Certificate Revocation Lists
- Using a Custom Trust Anchor Store for Oracle Security Token Service

18.5.1 Retrieving the Trust Anchors Store (amtruststore) Password

The Trust Anchors keystore password must be retrieved from CSF. Otherwise administrators cannot change the keystore. The store is located in:

$DOMAIN_HOME/config/fmwconfig/amtruststore

To retrieve the password of the, perform the following operations, which will display the password used to protect the Trust Anchors Keystore.

- Enter the WSLT scripting environment
- Execute the following command, by providing the connection information to the WLS Admin Server: `listCred(map="OAM_STORE", key="jks")`

To retrieve the Trust Anchors store password

1. Enter the WSLT scripting environment.
2. Connect to the WebLogic Server AdminServer, using the `connect()` command.
3. Execute the following command by providing the connection information to the AdminServer: `listCred(map="OAM_STORE", key="jks").`
4. Observe messages on the screen and note the password.
5. Proceed to "Managing the Trust Anchors Store (amtruststore)".

18.5.2 Managing the Trust Anchors Store (amtruststore)

The Trust Anchors keystore is managed using the keytool command. Certificates added to the keystore are detected by the Certificate Validation module.

> **Note:** Notification is performed using the JMX Notification Framework and might take some time, depending on the notification refreshing time (60 seconds by default).

Prerequisites

Retrieving the Trust Anchors Store (amtruststore) Password

To manage the Trust Anchors store (amtruststore)

1. Locate keytool.
2. Execute the following command.

   ```
   keytool -keystore $DOMAIN_HOME/config/fmwconfig/amtruststore
   -storetype JKS -alias orakey -file $CERT_FILE
   ```

3. Observe messages on the screen and enter a password if requested.
4. Proceed to "Managing Certificate Revocation Lists".

18.5.3 Managing Certificate Revocation Lists

Oracle Security Token Service and Oracle Access Manager use the common infrastructure certification validation module. Trusted Certificates and Certificate Revocation Lists (CRLs) used during certificate validation are stored in Trust Keystore and CRL ZIP file. The Oracle Security Token Service configuration stores the OCSP/CDP settings.

This section outlines how to add or remove certificate revocation lists (CLRs) to check the revocation status of a certificate, perform the following operations.

> **See Also:** "Managing Global Certificate Validation and Revocation" on page 4-6

Prerequisites

Have your Certificate Revocation List ready to import.

Task overview: Manage Certificate Validation and Revocation Lists

1. From the Oracle Access Manager Console System Configuration tab, Common Configuration section, select Certificate Validation.
2. See "Managing Certificate Revocation Lists (CLRs)" on page 4-7.
3. See "Managing Certificate Validation" on page 4-8:
4. See "Configuring CDP" on page 4-8.

18.5.4 Using a Custom Trust Anchor Store for Oracle Security Token Service

Optionally, if a particular deployment requires a set of trust anchors separate from that of Oracle Access Manager, another keystore can be configured as the trusted certificate store for Oracle Security Token Service. This can be done by having the administrator perform the following tasks.

> **Note:** Using a Custom Trust Anchor Store is an optional feature that most customers will not need.

Task overview: Deploying a custom keystore for trusted certificates

1. Create the JKS keystore in the $*DOMAIN_HOME*/config/fmwconfig directory.
2. In the Oracle Access Manager Console, Security Token Service Settings page, enter the full path name of the new trust store and Apply your changes.
3. In the domain where Oracle Access Manager with Oracle Security Token Service is deployed, the Custom Trust Anchor Keystore must be propagated manually by the administrator across all the servers.

19

Managing Templates, Endpoints, and Policies

This chapter provides the following information for Oracle Security Token Service.

- Prerequisites
- Introduction
- Searching for an Existing Template
- Managing Token Issuance Templates
- Managing Token Validation Templates
- Managing Oracle Security Token Service Endpoints
- Managing Token Issuance Policies and Constraints with Oracle Access Manager
- Managing TokenServiceRP Type Resources

19.1 Prerequisites

Both the Oracle Access Manager and Oracle Security Token Service services must be running, as described in "Enabling and Disabling Oracle Security Token Service" on page 17-8.

19.2 Introduction

Oracle Security Token Service supports control of who can access WSPs by defining Application Domains that provide access to resources based on policies. Application domains identify Web Services, along with authorization rules that determine who can request a security token based on the WSPs.

The following functionality is established by Trust Issuance Policies, which can be managed through Application Domain under the Policy Configuration tab of the Oracle Access Manager Console.

- Resource of type TokenServiceRP representing Relying Parties or Web Service Providers.
- Token Issuance Policy defining a policy for a set of resources of type TokenServiceRP.
- Constraint defining the identities of the clients that are allowed or denied issuance of tokens for the resources listed in the policy. The clients can either be Requester Partners or User from the Default Identity Store.

Oracle Security Token Service supports the creation of Relying Party Partner, representing a remote Web Service Provider that will be the consumer of a security token issued by Oracle Security Token Service.

For each Relying Party Partner, it is possible to define URLs that will be mapped to the partner, so that WS-Addressing endpoint specified in a WS-Trust Request can be mapped to an Oracle Security Token Service Relying Party Partner.

At runtime, when a client requests a token to be issued, Oracle Security Token Service will evaluate the Trust Issuance Policies to determine whether or not the token can be issued:

- The client will be identified either as a Requester Partner or as an end user
- If an AppliesTo element was present in the WS-Trust Request and was mapped to a Relying Party Partner, then the TokenServiceRP resource for the Trust Issuance Policy evaluation will be the Partner ID of that Oracle Security Token Service Relying Partner.
- If an AppliesTo element was present in the WS-Trust Request and could not be mapped to a Relying Party Partner, then the TokenServiceRP resource for the Trust Issuance Policy evaluation will be the UnknownRP defined in the OAM Suite Application Domain.
- If an AppliesTo element was missing in the WS-Trust Request, then the TokenServiceRP resource for the Trust Issuance Policy evaluation will be the MissingRP defined in the OAM Suite Application Domain.

Oracle Security Token Service requires the following items (at a minimum) to process a request and issue a token based on an incoming request (RST):

- EndPoints
- One Issuance Template
- One Validation Template
- One Requester Partner Profile that contains the token
- One Relying Party Partner Profile

Note: Partners might need to be provisioned.

An LDAP server is required for the Oracle Security Token Service to map the Username token that references the user to an LDAP User record, and then use that record to populate the outgoing token. Partners might need to be provisioned before they are available.

19.3 Searching for an Existing Template

All defined template names appear in the Search Results Table when you open either the Token Validation Template or Token Issuance Template node. To quickly find a specific template or set of templates, you can use the search controls.

This section explains the controls you can use to refine your search, which are similar whether you are searching for a Token Validation Template or a Token Issuance Template. It includes the following topics:

- About Template Search Controls

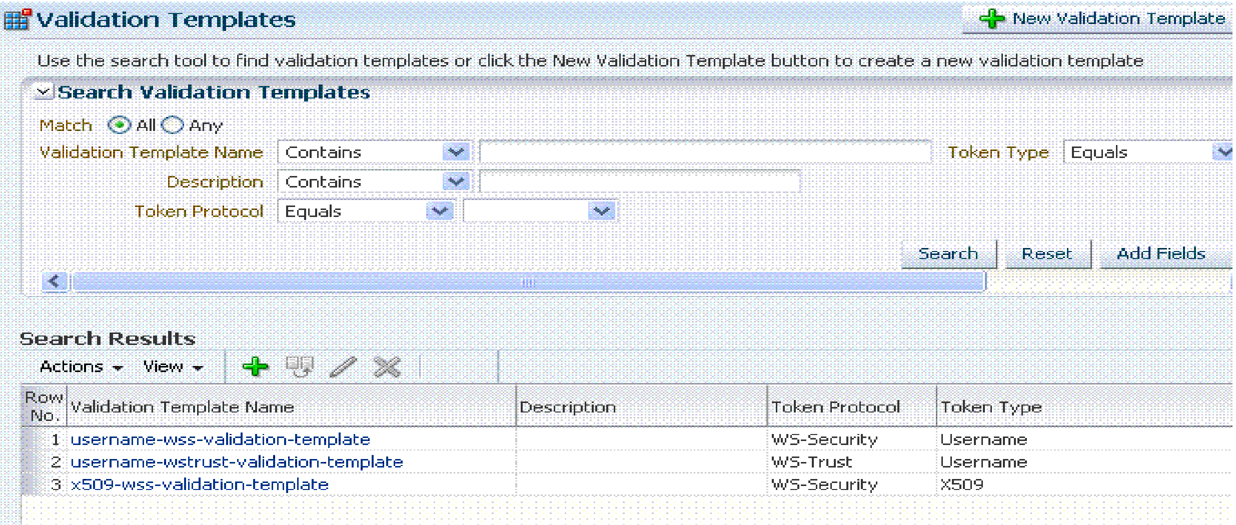
- Searching for a Template

19.3.1 About Template Search Controls

The following figures show the search pages where you will see many similarities:

- Figure 19–1, "Validation Templates Search Controls"
- Figure 19–2, "Issuance Template Search Controls"

Figure 19–1 Validation Templates Search Controls



Validation Templates

Use the search tool to find validation templates or click the New Validation Template button to create a new validation template

Search Validation Templates

Match ● All ○ Any

Validation Template Name: Contains

Token Type: Equals

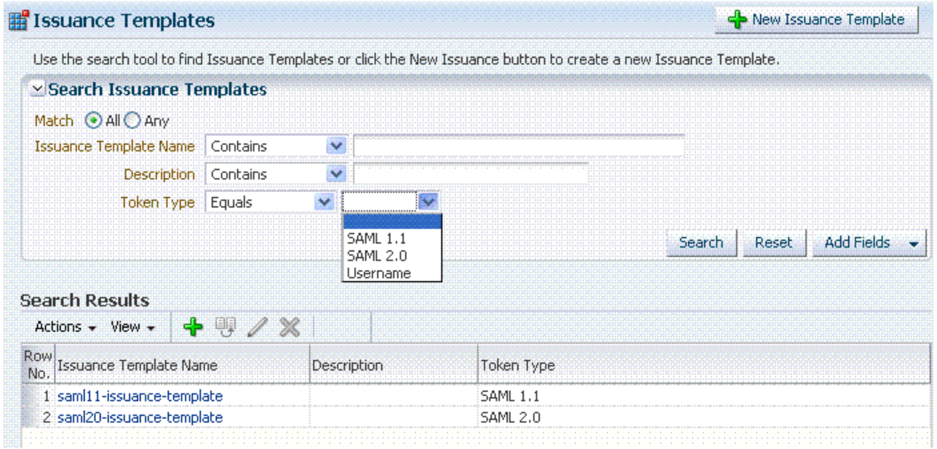
Description: Contains

Token Protocol: Equals

Search Results

Row No.	Validation Template Name	Description	Token Protocol	Token Type
1	username-wss-validation-template		WS-Security	Username
2	username-wstrust-validation-template		WS-Trust	Username
3	x509-wss-validation-template		WS-Security	X509

Figure 19–2 Issuance Template Search Controls



Issuance Templates

Use the search tool to find Issuance Templates or click the New Issuance button to create a new Issuance Template.

Search Issuance Templates

Match ● All ○ Any

Issuance Template Name: Contains

Description: Contains

Token Type: Equals

SAML 1.1
SAML 2.0
Username

Search Results

Row No.	Issuance Template Name	Description	Token Type
1	saml11-issuance-template		SAML 1.1
2	saml20-issuance-template		SAML 2.0

Table 19–1 describes the controls available to refine a template search. Unless explicitly stated, all elements are available for both Validation and Issuance Template searches.

Table 19–1 Template Search Controls

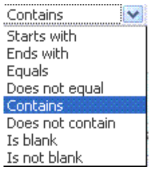
Element	Description
Match	Choose All to search for a template that matches all your specifications. Choose Any to search for a template that matches at least one of your specifications.
Search Operations List	A list of operations from which you choose one to help refine your search.  Contains Starts with Ends with Equals Does not equal Contains Does not contain Is blank Is not blank
... Template Name	Choose an operation from the list and enter information in the field to help refine your search.
Description	Refine your search using the optional description field.
Token Protocol	Choose the token protocol from those listed:
Validation Template only	■ WS-Trust ■ WS-Security
Token type	Choose the token type. Both standard and custom token types are included. ■ Username: Consumption and Creation ■ X.509: Consumption ■ SAML: Consumption & Creation ■ OAM 11g: Consumption using the OBO (on behalf of) field ■ Kerberos: Consumption ■ Custom: Consumption the OBO (on behalf of) field and Creation
Search	Initiates the Search function using criteria in the form.
Reset	Resets the Search form with defaults only.

Table 19–1 (Cont.) Template Search Controls

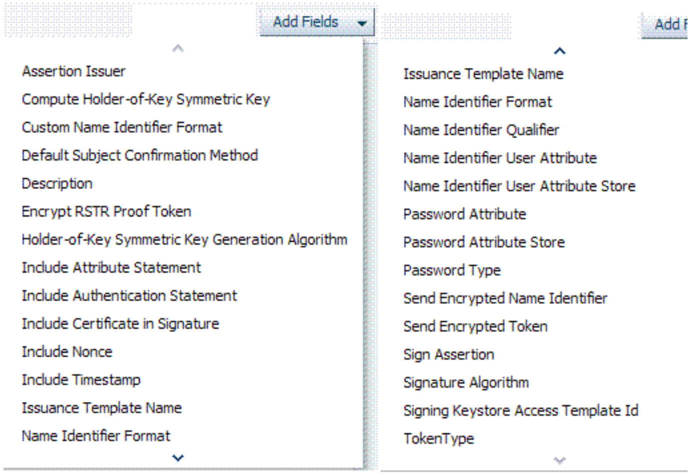
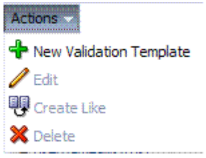
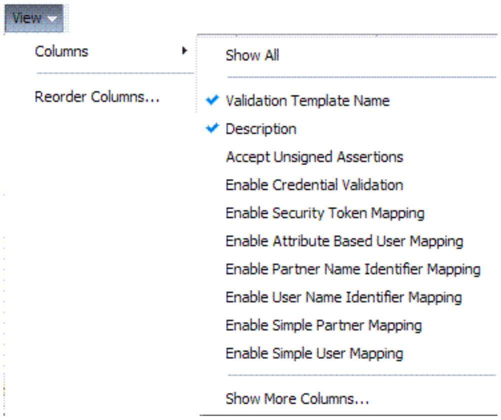
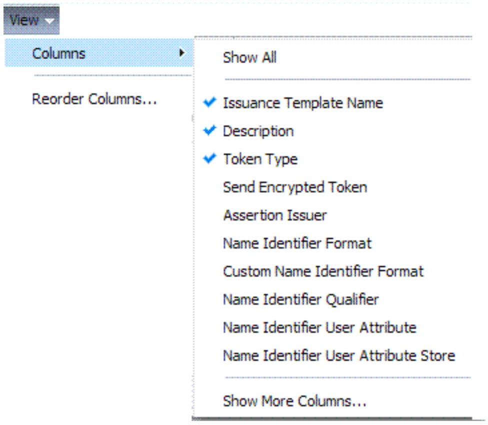
Element	Description
Add Fields	A list of additional items you can add as search criteria.  Add Fields Assertion Issuer Compute Holder-of-Key Symmetric Key Custom Name Identifier Format Default Subject Confirmation Method Description Encrypt RSTR Proof Token Holder-of-Key Symmetric Key Generation Algorithm Include Attribute Statement Include Authentication Statement Include Certificate in Signature Include Nonce Include Timestamp Issuance Template Name Name Identifier Format Add F Issuance Template Name Name Identifier Format Name Identifier Qualifier Name Identifier User Attribute Name Identifier User Attribute Store Password Attribute Password Attribute Store Password Type Send Encrypted Name Identifier Send Encrypted Token Sign Assertion Signature Algorithm Signing Keystore Access Template Id TokenType
Search Results Table	Itemizes the results of your search based on choices in the View menu, described later in this table.
Actions menu *Shown: Actions for Validation Template*	Provides the following functions that can be performed on a selection in the results table:  Actions New Validation Template Edit Create Like Delete Note: Actions menu functions mirror command buttons above the results table. For example: ■ New ... Template: Click the New ... Template button at the top of the Search page, or select New ... Template from the menu, or click the **+** button above the table. ■ Edit: Click a name in the Results Table, or select Edit from the Actions menu, or click the Edit (pencil) command button above the Results Table. ■ Create Like: Select the desired row in the table and either select Create Like from the Actions menu, or click the Create Like command button above the table ■ Remove: Select the desired row in the Results Table and either select Delete from the Actions menu, or click the Delete (**X**) command button above the table.

Table 19–1 (Cont.) Template Search Controls

Element	Description
View menu Validation Template only	A list from which you can identify which information to display in the results table.  View Columns Reorder Columns... Show All ✓ Validation Template Name ✓ Description Accept Unsigned Assertions Enable Credential Validation Enable Security Token Mapping Enable Attribute Based User Mapping Enable Partner Name Identifier Mapping Enable User Name Identifier Mapping Enable Simple Partner Mapping Enable Simple User Mapping Show More Columns...
View menu Issuance Template only	A list from which you can identify which information to display in the results table.  View Columns Reorder Columns... Show All ✓ Issuance Template Name ✓ Description ✓ Token Type Send Encrypted Token Assertion Issuer Name Identifier Format Custom Name Identifier Format Name Identifier Qualifier Name Identifier User Attribute Name Identifier User Attribute Store Show More Columns...
△▽	Controls you can choose to define the order of items listed in the results table: ▪ Ascending ▪ Descending

19.3.2 Searching for a Template

Users with valid Administrator credentials can use the following procedure to use search controls to locate a specific template or set of templates. For example, to locate all templates of a certain token type you can simply choose the type of token. To refine the search further to all templates of a specific token type and name.

When performing these steps, fill in as much or as little as you want. Skip any steps that do not apply to you.

See Also: "About Template Search Controls"

To search for a template

1. From the System Configuration tab, expand the Secure Token Services Settings section.
2. In the navigation tree, double-click the desired Template node. For example: Token Validation Templates.
3. Edit Search Criteria (Table 19–1). For example:
   - Match: All
   - Name: contains em
   - Token Type: equals Username
4. Click Search and review results.

19.4 Managing Token Issuance Templates

An issuance template contains rules on how a token will be created and is specific to a token type. Each issuance template indicates Signing and Encryption and also contains Attribute Name, Value Mapping, and Filtering settings to be sent as part of the token.

This section provides the following information:

- About Managing Token Issuance Templates
- Managing a Token Issuance Template

19.4.1 About Managing Token Issuance Templates

Each Token Issuance Template indicates how to construct a token. In other words, which signing or encryption to use when constructing a token. Each Token Issuance Template also defines the attributes mapping and filtering rules to be applied to the attributes that will be included in the outgoing token. However, Issuance Templates do not list the attributes that will be sent in the outgoing token: these are defined in the Relying Party Partner Profile.

Token Issuance Template details which will differ depending on your chosen token type. Table 19–2 describes where to find more information.

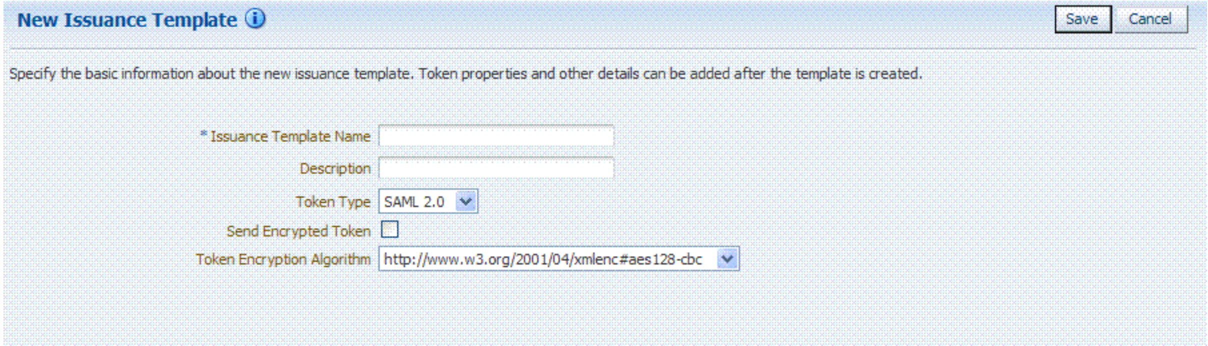
Table 19–2 Issuance Template Requirements

Topic	Figures and Tables
General Details	Figure 19–3, Table 19–3
Issuance Properties: Username Tokens	Figure 19–4, Table 19–4
Issuance Properties: SAML Tokens	Figure 19–5, Table 19–6
Security: SAML Tokens	Figure 19–6, Table 19–6
Attribute Mapping: SAML Tokens	Figure 19–9, Table 19–7

General Details

Figure 19–3 shows the New Issuance Template page with defaults showing. Unless explicitly stated, General information is the same regardless of the Token Type you choose. For more information, see Table 19–3. After you fill in General information and click Save, you cannot return and edit the template name or token type.

Figure 19–3 Issuance Template: General Details and Defaults



Specify the basic information about the new issuance template. Token properties and other details can be added after the template is created.

* Issuance Template Name	
Description	
Token Type	SAML 2.0
Send Encrypted Token	□
Token Encryption Algorithm	http://www.w3.org/2001/04/xmlenc#aes128-cbc

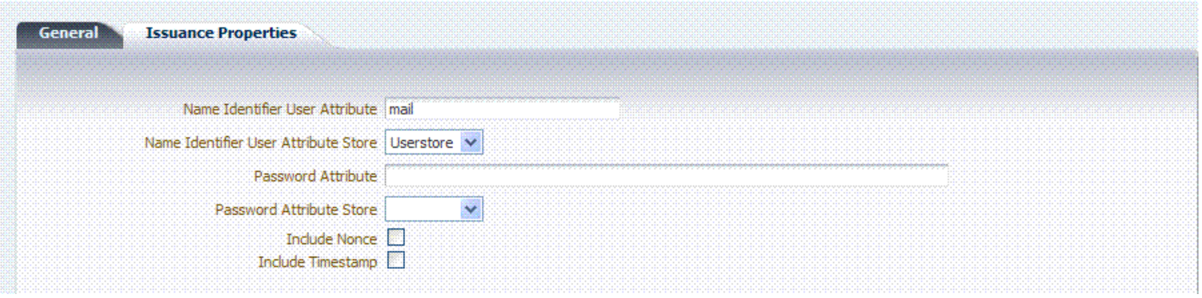
Table 19–3 Issuance Template: General Details

Elements	Description
Issuance Template Name	Enter a unique name for this template.
Description	Optional.
Token Type	Choose a standard (or custom, if any) token type from those listed.
SAML, Username, and Custom Token Types	
Send Encrypted Token	Click to enable token encryption.
Token Encryption Algorithm	When token encryption is enabled, choose a Token Encryption Algorithm from those listed.

Issuance Properties: Username Token Type

If the token type is Username, the Issuance Properties shown in Figure 19–4 are needed for a Username token type template.

Figure 19–4 Issuance Properties: Username Token Type



Name Identifier User Attribute	mail
Name Identifier User Attribute Store	Userstore
Password Attribute	
Password Attribute Store	
Include Nonce	□
Include Timestamp	□

Table 19–4 describes the Issuance Properties for the Username token type.

Table 19–4 Issuance Properties: Username Token Type

Element	Description
Name Identifier User Attribute	Attribute to be used to populate the Username element in the Username Token.

Table 19–4 (Cont.) Issuance Properties: Username Token Type

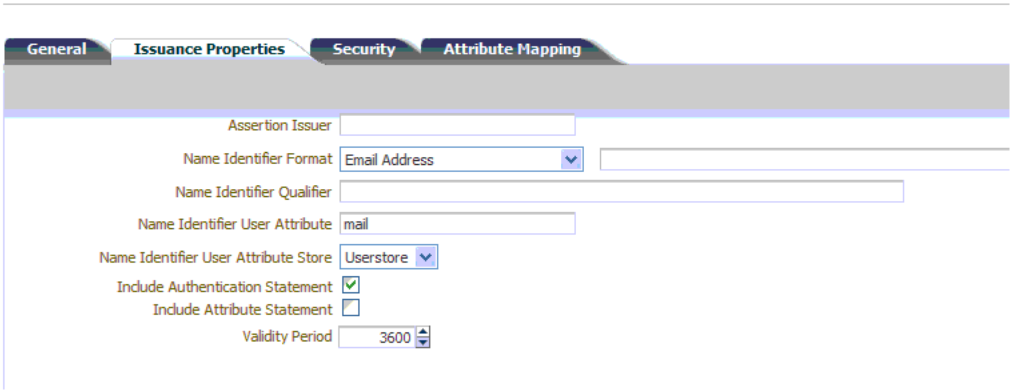
Element	Description
Name Identifier User Attribute Store	Choose the user attribute store type: ■ Userstore ■ Context Note: If the Attribute Store is the Userstore, LDAP is used to retrieve the attribute from the user record. If the Attribute Store is context, data from the incoming token is used as the attribute source.
Password Attribute	Attribute to be used to populate the Password element in the Username Token.
Password Attribute Store	Choose the password attribute store type: ■ Userstore ■ Context Note: If the Attribute Store is the Userstore, LDAP is used to retrieve the attribute from the user record. If the Attribute Store is context, data from the incoming token is used as the attribute source.
Include Nonce	Indicates whether or not a Nonce made of random data should be included in the Username token Default: Disabled
Include Timestamp	Indicates whether or not a the Created element should be included in the Username token Default: Disabled

Issuance Properties: SAML Token Types

SAML 1.1 and 2.0 token types require the issuance properties illustrated in Figure 19–5.

> **Note:** These issuance properties differ from those for Username token type.

Figure 19–5 Issuance Properties: SAML Token Types



Assertion Issuer

Name Identifier Format: Email Address

Name Identifier Qualifier

Name Identifier User Attribute: mail

Name Identifier User Attribute Store: Userstore

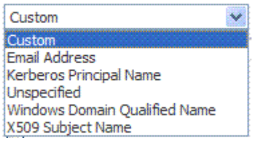
Include Authentication Statement ☑

Include Attribute Statement ☐

Validity Period: 3600

Table 19–5 describes all Issuance Properties by token type. Only SAML token types require issuance properties.

Table 19–5 Issuance Properties: SAML Token Types

Element	Description
Assertion Issuer	Specifies the identifier representing the issuer of the assertion. This string is used to represent this Oracle Security Token Service as the issuer of the assertion.
Name Identifier Format	Choose a format from the list and then enter the details in the text field.  Custom Custom Email Address Kerberos Principal Name Unspecified Windows Domain Qualified Name X509 Subject Name
Name Identifier Qualifier	Contains the string that will be set as the Name Identifier Qualifier.
Name Identifier User Attribute	References the attribute that will be used to populate the value of the Name Identifier.
Name Identifier User Attribute Store	■ Userstore ■ Context Note: If the Attribute Store is the Userstore, LDAP is used to retrieve the attribute from the user record. If the Attribute Store is context, data from the incoming token is used as the attribute source.
Include Authentication Statement	Indicates whether or not a SAML Authentication Statement should be included in the Assertion. Default: Disabled Note: An authentication operation is required for a statement of this type to be included. An authentication statement will be included if the incoming token contained some authentication data and that those were validated (for example, the incoming SAML Assertion contains an authentication statement, or a Username Token contains credentials that were validated).
Include Attribute Statement	Indicates whether or not a SAML Attribute Statement will be included in the outgoing Assertion. A statement of this type will be included only if this flag is set to true and if at least one attribute is included in the outgoing Assertion. Default: Enabled Note: the RP PP will determine which attributes need to be included in an outgoing token.
Validity Period	Specify the length of time (in seconds) that the token will be valid. Default: 3600 (seconds)

Security Details: SAML Tokens

Only SAML token types require Security Details, as shown in Figure 19–6 and described in Table 19–6.

Figure 19–6 Security Details: SAML Tokens

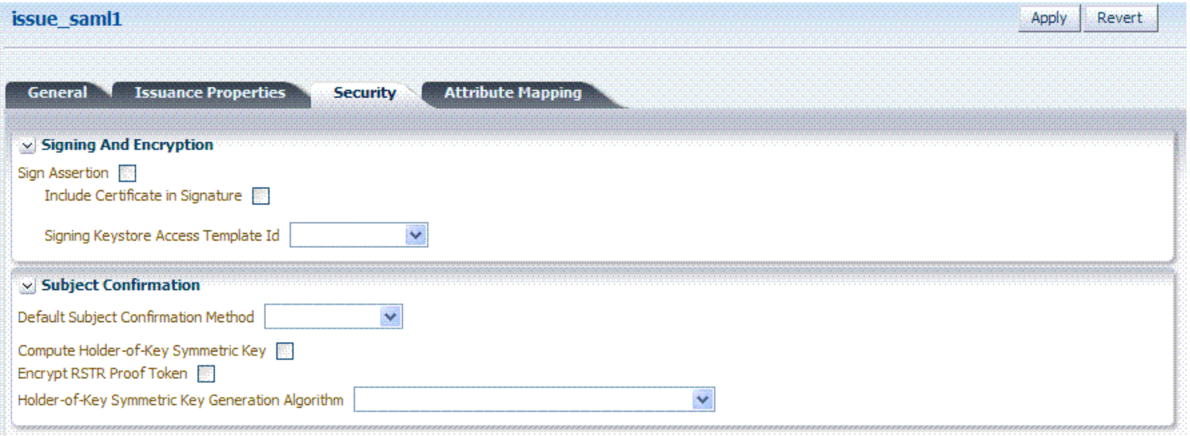
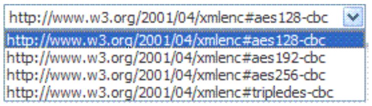

issue_saml1
Apply
Revert
General
Issuance Properties
Security
Attribute Mapping
Signing And Encryption
Sign Assertion □
Include Certificate in Signature □
Signing Keystore Access Template Id
Subject Confirmation
Default Subject Confirmation Method
Compute Holder-of-Key Symmetric Key □
Encrypt RSTR Proof Token □
Holder-of-Key Symmetric Key Generation Algorithm

Table 19–6 Security Details: SAML Tokens

Elements	Description
Signing And Encryption	Indicates whether or not the Assertion will be signed using the Key referenced by the Signing Keystore Access Template ID field.
Sign Assertion	Indicates whether or not the signing certificate will be included in the Assertion. Default: Enabled
Include Certificate in Signature	Default: Enabled
Signing Keystore Access Template Id	References the key to be used to sign assertions created with this issuance template. The key templates are defined in the Security Token Service Settings section.
Subject Confirmation	
Default Subject Confirmation Method	Indicates which Subject Confirmation Method will be used by default, if the requester did not specify a method in the WS-Trust request. Possible values are: ■ Bearer ■ Holder of Key with Public Key ■ Holder of Key with Symmetric Key ■ Sender Vouches
Compute Holder-of-Key Symmetric Key	Default: Enabled Indicates whether or not Oracle Security Token Service will generate random data when creating the Secret Key for the Holder of Key Symmetric Key data. ■ If true, the server will generate the secret key if the client did not specify entropy. Otherwise it will derive the key from the client and server entropy ■ If false, the client entropy will be used as the secret key
Encrypt RSTR Proof Token	Indicates whether or not the Proof Token must be encrypted when returning the server entropy or secret key to the requester in the WS-Trust response, when the Subject Confirmation method is Holder of Key with Symmetric Key Default: Disabled

Table 19–6 (Cont.) Security Details: SAML Tokens

Elements	Description
Holder-of-Key Symmetric Key Generation Algorithm	Indicates the symmetric key generation algorithm to use to create the secret key when the Subject Confirmation method is Holder of Key with Symmetric Key:  http://www.w3.org/2001/04/xmlenc#aes128-cbc http://www.w3.org/2001/04/xmlenc#aes128-cbc http://www.w3.org/2001/04/xmlenc#aes192-cbc http://www.w3.org/2001/04/xmlenc#aes256-cbc http://www.w3.org/2001/04/xmlenc#tripledes-cbc

Attribute Mapping: SAML Tokens

When the token type is SAML 1.1 or 2.0, it is possible to define attribute mapping and filter rules that will be applied to the attributes included in the Assertion.

There are three different rules:

- Attribute name mapping where the local name of an attribute can be changed to another value. For example, givenname can be changed to firstname.
- Attribute value mapping where the local value of an attribute can be translated to another value. For example, President to CEO.
- Attribute value filtering where the local value of an attribute can be filtered so it is not included in the outgoing assertion. For example, some sensitive attribute values could be removed while others would be issued.

See Also: Token Mapping attributes in Figure 19–9 and Table 19–11.

Table 19–7 Issuance Template: Attribute Mapping, SAML Token

Element	Description
Attribute Name Mapping	Defines an optional mapping between the local name of an attribute, and the name used to reference this attribute in the assertion. The mapping is optional. If an attribute does not have a mapping defined, then its local name will be used, and the namespace will be set to `urn:oracle:security:fed:attrnamespace` for SAML 1.1 Assertions or the format will be set to `urn:oasis:names:tc:SAML:2.0:attrname-format:basic` for SAML 2.0 Assertions. ■ External Attribute: Contains the externam name of the attribute as it will appear in the Assertion. ■ Local Attribute: Contains the local name of the attribute. ■ Format of Namespace: Contains an optional Format or Namespace. If missing, the namespace will be set to urn:oracle:security:fed:attrnamespace for SAML 1.1 Assertions or the format will be set to urn:oasis:names:tc:SAML:2.0:attrname-format:basic for SAML 2.0 Assertions.

Table 19–7 (Cont.) Issuance Template: Attribute Mapping, SAML Token

Element	Description
Attribute Value Mapping	Defines an optional value mapping for an attribute that will be included in the Assertion. Note: this attribute value mapping applies to an Attribute Name mapping. In order to define an attribute mapping for an attribute, it is required to first define an attribute name mapping for that attribute. ■ External Attribute: Contains the value that should be included in the Assertion, if the local attribute value matches the Local Attribute/Local Null fields. ■ Local Attribute: Contains the local value of the attribute. ■ External Null: Indicates if the value to be included in the Assertion should be null, if the local value of the attribute matches the Local Attribute/Local Null fields. ■ Local Null: Represents a null local value. ■ Ignore Case: Indicates whether or not Oracle Security Token Service should ignore case when comparing the attribute value to the Local Attribute field.
Attribute Value Filters	Defines an optional value filtering for an attribute that will be included in the Assertion. Note: This attribute value filtering applies to an Attribute Name mapping. In order to define an attribute filtering for an attribute, it is required to first define an attribute name mapping for that attribute. ■ Condition: Contains the condition associated with the expression to determine whether or not the attribute value should be filtered. The possible values are described in "Attribute Value Condition Filters". ■ Expression: Contains data that will be used to evaluate the filtering rule. ■ Ignore Case: Indicates whether or not Oracle Security Token Service should ignore case when comparing the attribute value to the expression field.

Attribute Value Condition Filters

This optional value filtering applies to an Attribute Name mapping and will be included in the Assertion. To define an attribute filtering for an attribute, you must first define an attribute name mapping for that attribute. The Condition is associated with the expression to determine whether or not the attribute value should be filtered. The possible Condition values are:

- regexp: the expression will contain a regular expression, and if it evaluates to true, the attribute value will be filtered.
- equals: if the attribute value matches the data contained in the expression field, then it will be filtered.
- not-equals: if the attribute value does not match the data contained in the expression field, then it will be filtered.
- not-equals: if the attribute value does not match the data contained in the expression field, then it will be filtered.
- endswith: if the attribute value ends with the data contained in the expression field, then it will be filtered.

- contains: if the attribute value contains an occurrence of the data contained in the expression field, then it will be filtered.
- not-contains: if the attribute value does not contains any occurrence of the data contained in the expression field, then it will be filtered.
- equals-null: if the attribute value is null, then it will be filtered.
- not-equals-null: if the attribute value is not null, then it will be filtered.

19.4.2 Managing a Token Issuance Template

Users with valid Administrator credentials can use this procedure as a guide when developing a new Token Issuance Template (or editing an existing template) Skip any steps that do not apply to you.

The following procedure describes how to create a new Token Issuance Template for a Security Assertion Markup Language (SAML) token.

Prerequisites

Confirm that the desired LDAP Identity Store is registered with Oracle Access Manager and configured as the default Identity Store.

See Also:

- About Managing Token Issuance Templates
- Searching for an Existing Template

To create a new token Issuance template

1. Display the list of existing Token Issuance Templates.

   Oracle Access Manager Console
     System Configuration
       Security Token Services
         Token Issuance Templates

2. **New Token Issuance Template**:

   a. Click the New Issuance Template button in the upper-right corner (or click the Add (+) command button above the Search Results table).

   b. **General**: Define general information for this template and see:

      Table 19–3, " Issuance Template: General Details"

   c. Click Save and dismiss the confirmation window (or click Cancel without saving).

   d. **Username Token Type**: Define issuance parameters for this template and see:

      Table 19–4, " Issuance Properties: Username Token Type"

   e. **SAML Token Type**: Define parameters for this template and see:

      Table 19–5, " Issuance Properties: SAML Token Types"

      Table 19–6, " Security Details: SAML Tokens"

      Table 19–7, " Issuance Template: Attribute Mapping, SAML Token"

   f. Click Apply (or click Revert without saving it).

   g. Close the definition.

3. **Find an Existing Template**: From the Security Token Service section of the System Configuration tab:

   a. **Find All**: Double-click the Token Issuance Templates node and review the results table.

   b. **Narrow the Search**: Specify your search criteria (Table 19–1), click the Search Button, and review the results table.

   c. **Reset the Search Form**: Click the Reset button.

4. **Edit a Template**: Start with the saved page you just created.

   Alternatively: Use Step 3 to find the desired template and click the name in the Search Results table to display the definition.

   a. Edit details as needed.

   b. Click the Apply button at the top of the page to submit changes (or Revert to undo your changes).

5. **Remove a Template**:

   a. Click the desired name in the Search Results table to select the item to remove.

   b. From the Actions menu, click Delete (or click the Delete (X) command button above the table.

   c. Click the Delete button in the Confirmation window (or click No to cancel the operation).

19.5 Managing Token Validation Templates

A validation template is used to validate an incoming token and, optionally, map the incoming token to either a Requester Partner or a user record:

- For OnBehalfOf use cases, a WS-Trust Validation Template must be present.
- For validating an Assertion, one Issuing Authority Partner Profile must be present.

The Oracle Security Token Service Endpoint is linked to a WSS Validation Template that indicates how to validate the token in the WSS header and how to map the token and binding data to a Requester.

This section provides the following topics.

- About Managing Token Validation Templates
- Managing Token Validation Templates

19.5.1 About Managing Token Validation Templates

An Oracle Security Token Service Endpoint is always mapped with a WS-Security Validation Template that indicates how to map the request to a requester entry or to a user:

- If mapping is required and no match is found, processing will fail.
- If no mapping is required, a default requester partner profile will be used.
- In either case, a requester partner profile is retrieved.
- If a mapping is performed to a user record, a default requester partner profile will be used.

- If a mapping is performed to a requester partner entry, the requester partner profile for this partner will be used.

A validation template determines the token validation rules:

- Whether or not to validate and map the incoming token.
- The mapping rules to be used if mapping is enabled.

A validation template is specific to a token type and specific to a protocol as described in Table 19–8.

Table 19–8 Validation Template Protocols

Protocol	Description
WS-Security	Validates only WS-Security Tokens: ■ Possible Mapping actions: no action, map binding data to partner, map incoming token to partner, map incoming token to user and binding data to partner, map incoming token to user ■ Token Types supported: SAML 1.1, SAML 2.0, Username X.509, Kerberos, None. When you toggle the Token Protocol from WS-Trust to WS-Security, options in the Token Type list do not change. However, the required "Default Partner Profile" list appears from which you must choose one profile for WS-Security.
WS-Trust	Validates only Tokens included in OBO (on behalf of) field of the RST (request): ■ Possible Mapping actions: none, map incoming token to user ■ Token Types supported: SAML 1.1, SAML 2.0, Username, X.509, Kerberos, OAM, Custom.

A validation template mapping rules determines how the incoming data is mapped to a user or a partner, using data from the incoming token:

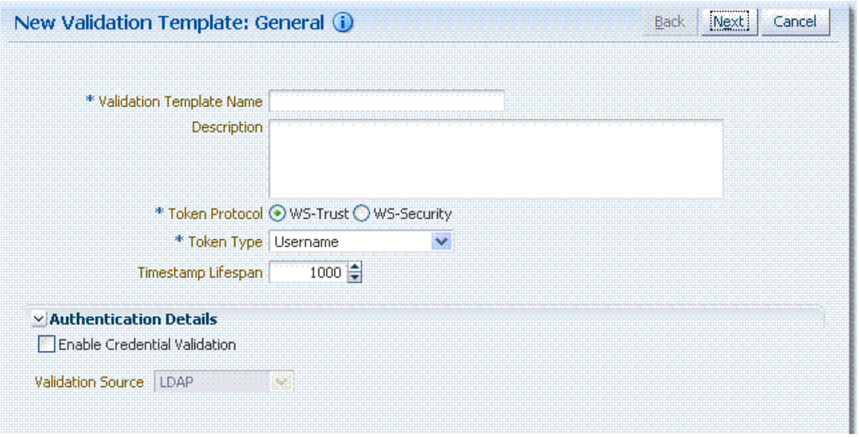
- Username for Username Token
- UserID for Kerberos Token
- NameID and attributes for SAML Token
- DN Components for X.509 Token
- Attributes from a Custom

Mapping is performed as follows:

- Simple mapping: one incoming attribute matched against one user record attributes
- Complex LDAP query: LDAP query with placeholders for incoming data (e.g.: (&(sn=%lastname%)(mail=%email%))
- NameID Mapping table for SAML Token

Figure 19–7 illustrates default General details on the New Validation Template page.

Figure 19–7 New Validation Template page: General Page Defaults



* Validation Template Name

Description

* Token Protocol ● WS-Trust ○ WS-Security

* Token Type Username

Timestamp Lifespan 1000

Authentication Details

□ Enable Credential Validation

Validation Source LDAP

Table 19–9 describes the elements on the New Validation Template, General page.

Table 19–9 New Validation Template: General Details

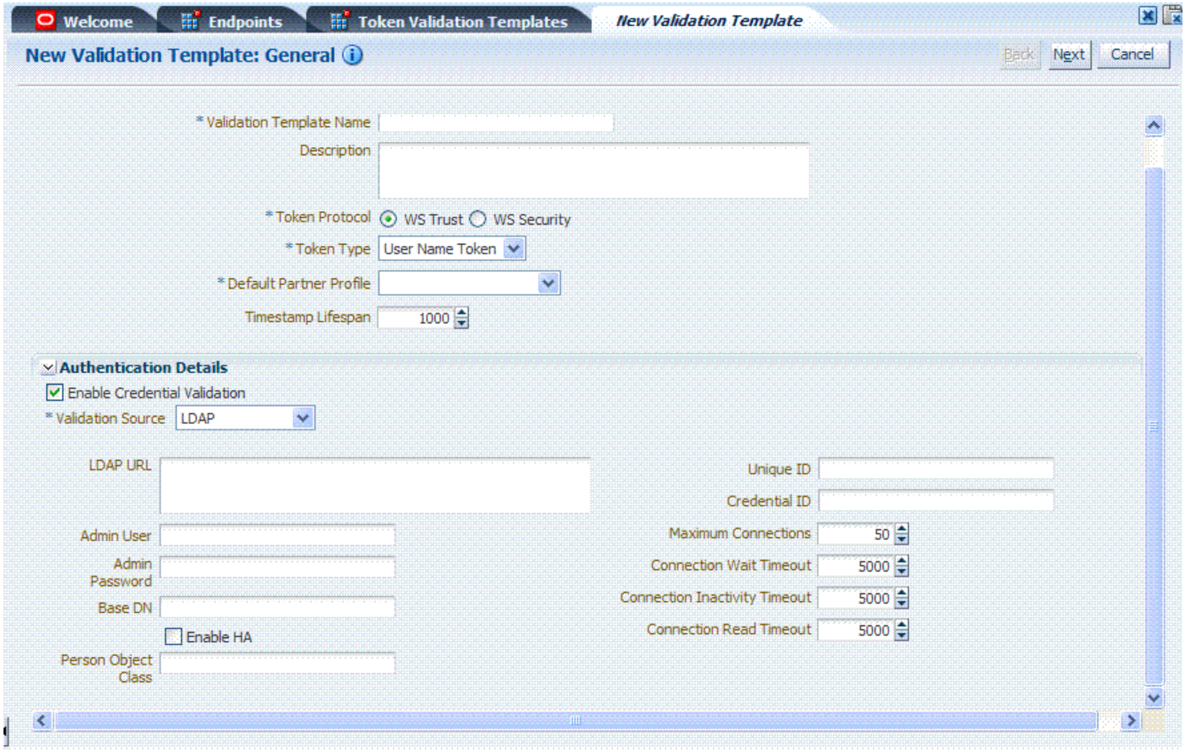
Element	Description
Back	Click this button to return to the previous page.
Next	Click this button to proceed to the next page.
Cancel	Click this button to dismiss the page.
Validation Template Name	The name you choose for this template. For example: `email-wstrust-valid-temp`
Description	Optional.
Token Protocol	The type of Validation Template to be created. Type can be either: ■ WS-Trust: This template will be used to validate and map tokens included in the OnBehalfOf element of the WS-Trust request. ■ WS-Security: This template will be used to validate and map tokens located in the Security SOAP Header of the incoming message
Token Type	A list of in-bound token types from which you choose the one to use for this template. The token type options depends on the protocol type: ■ WS-Trust: SAML 1.1, SAML 2.0, Username, X.509, Kerberos, OAM, Custom ■ WS-Security: SAML 1.1, SAML 2.0, Username, X,509, Kerberos, None
Default Partner Profile	Only applies to WS-Security Validation Template References the default requester partner profile to use, in case the incoming request is not mapped to a requester partner. For example, if the request is mapped to a user instead. A requester partner profiles contains settings that are used during the request processing. If the incoming request was mapped to a requester partner, then the partner profile for that requester will be retrieved and used as the requester partner profile

Table 19–9 (Cont.) New Validation Template: General Details

Element	Description
Timestamp Lifespan	Applies only to Username and SAML Validation Templates. It determines the validity time of a Token (for Username Token, only if it contains a Created element indicating the instant it was created). Default: 1000 (seconds)
Authentication Details	Specific to username token validation template.
Enable Credential Validation	Check this box to enable validation using credentials contained in the username token. When enabled, Oracle Security Token Service will validate the username and the password elements contained in the username token, using the specified validation source. Note: password digest as defined in the Username Token WS-Security Profile is not supported in this release. See Also: Table 19–10, " New Validation Template: Authentication Details"

Figure 19–8 illustrates the General details page when Enable Credential Validation is checked and, as a result, the Authentication Details section of the page is visible with its default values. This is specific to username token validation.

Figure 19–8 New Validation Template: General Authentication Details



* Validation Template Name	
Description	
* Token Protocol	● WS Trust ○ WS Security
* Token Type	User Name Token
* Default Partner Profile	
Timestamp Lifespan	1000

Authentication Details

☑ Enable Credential Validation

* Validation Source	LDAP		
LDAP URL		Unique ID	
		Credential ID	
Admin User		Maximum Connections	50
Admin Password		Connection Wait Timeout	5000
Base DN		Connection Inactivity Timeout	5000
□ Enable HA		Connection Read Timeout	5000
Person Object Class			

Table 19–10 describes Authentication related details that are available when you choose Enable Credential Validation.

Table 19–10 New Validation Template: Authentication Details

Element	Description
Validation Source	A list from which you can choose a credential validation sources There are four types of validation sources when validating the credentials contained in a username token: ■ LDAP: a standalone LDAP server will be used to validate the credentials. The connection information will need to be entered ■ Embedded LDAP: the LDAP server embedded in the WebLogic server will be used to validate the credentials. No information is required. ■ Userstore: the default User Identity Store configured in the Common Configuration -> Data Sources will be used to validate the credentials. No information is required in this validation template screen ■ Partner: the credentials will be verified against the username/password information entered in the Requester Partner entries. Note: When selected, the Token Mapping configuration section is disabled, because the token will have been mapped to a requester partner after the credentials validation operation.
LDAP URL	The URL of the LDAP server.
Admin User	The username of an account used to perform lookups in the LDAP server.
Admin Password	The password of an account used to perform lookups in the LDAP server.
Base DN	The Base search DN used when looking up user records.
Enable HA	Indicates whether or not the LDAP server is in HA mode, fronted by a load balancer.
Person Object Class	The person object class associated with the user records.
Unique Id	The attribute of the user record containing the user unique identifier data. In most cases, is identical to the Credential ID field.
Credential Id	The attribute of the user record containing the username data. This field will be used to lookup user records, based on the username.
Maximum Connections	The maximum number of concurrent opened LDAP connections Default: 40
Connection Wait Timeout	Maximum amount of time to wait when opening a new connection. Default: 5 (seconds)
Connection Inactivity Timeout	Maximum amount of inactivity time for an LDAP connection, before closing it. Default: 300 (seconds)
Connection Read Timeout	Maximum number of concurrent opened LDAP connections. Default: 10 (seconds)

Token Mapping

The Token Mapping section indicates the following:

- If an incoming token needs to be mapped.
- If the incoming token needs to be mapped, what kind of mapping is done. For example, mapping token to user, mapping token to partner, and so on.
- How the mapping is done. For example, by mapping a token attribute to a partner/user attribute, or by using an LDAP query involving several token attributes.

Mapping rules determine how the incoming data is mapped to a user or a partner. The following data of the incoming token is used:

- Username for UNT
- UserID for Kerberos
- NameID and attributes for SAML
- DN Components for X.509
- Attributes from custom

Mapping is performed using the following:

- Simple mapping: One incoming attribute matched against one user record attributes.
- Complex LDAP query: An LDAP query with placeholders for incoming data. For example, `(&(sn=%lastname%)(mail=%email%))`
- A NameID Mapping table for SAML

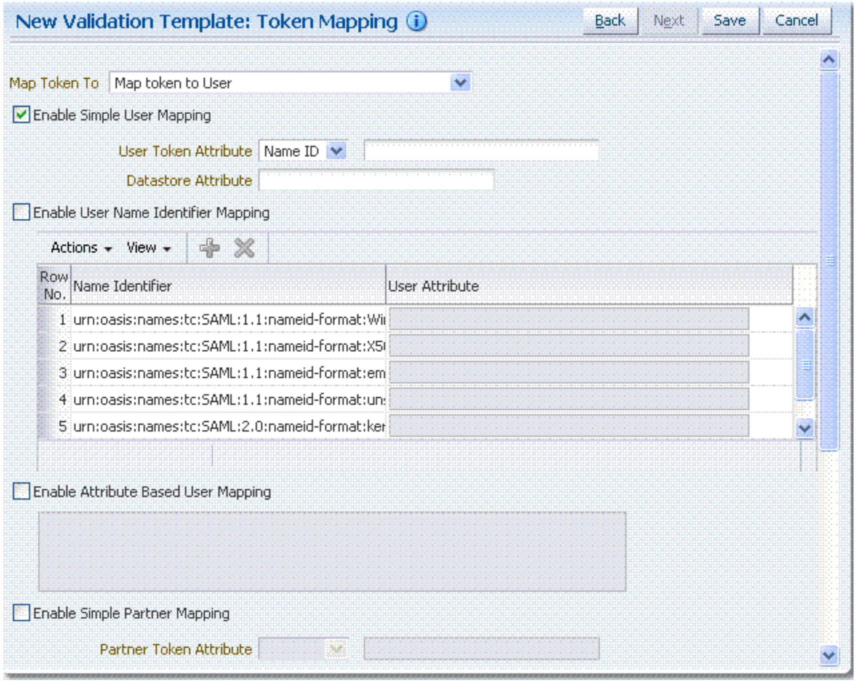
Following are several Token Mapping Examples for a new Validation Template:

- Figure 19–9, "Token Mapping: SAML2 WS-Security Validation Template"
- Figure 19–10, "Token Mapping, username-wstrust-validation-template"
- Figure 19–11, "Token Mapping: x509-wss-validation-template"

Figure 19–9 shows the mapping configuration settings required for Oracle Security Token Service to map the token to a user record, by matching the NameID value to user records that have a matching attribute, based on the NameID format:

- Enable Map Token to User
- Enable Simple User Mapping
- Disable Attribute Based User Mapping

Figure 19–9 Token Mapping: SAML2 WS-Security Validation Template



New Validation Template: Token Mapping

Map Token To: Map token to User

☑ Enable Simple User Mapping

User Token Attribute: Name ID

Datastore Attribute:

☐ Enable User Name Identifier Mapping

Row No.	Name Identifier	User Attribute
1	urn:oasis:names:tc:SAML:1.1:nameid-format:Wi	
2	urn:oasis:names:tc:SAML:1.1:nameid-format:X5	
3	urn:oasis:names:tc:SAML:1.1:nameid-format:em	
4	urn:oasis:names:tc:SAML:1.1:nameid-format:un:	
5	urn:oasis:names:tc:SAML:2.0:nameid-format:ker	

☐ Enable Attribute Based User Mapping

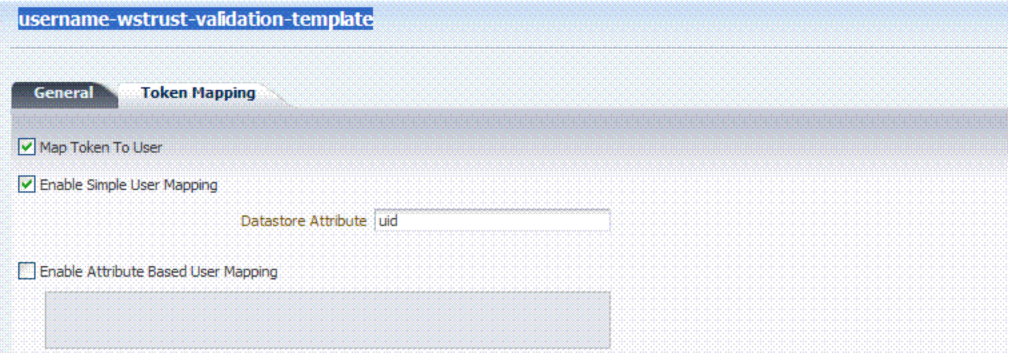
☐ Enable Simple Partner Mapping

Partner Token Attribute:

Figure 19–10 shows the mapping configuration settings required for Oracle Security Token Service to map the token to a user record by matching the username element of the Username token to a user record that has a matching uid. The required settings are:

- Enable Map Token to User
- Enable Simple User Mapping
- Datastore Attribute set to uid
- Disable Attribute Based User Mapping

Figure 19–10 Token Mapping, username-wstrust-validation-template



username-wstrust-validation-template

☑ Map Token To User

☑ Enable Simple User Mapping

Datastore Attribute: uid

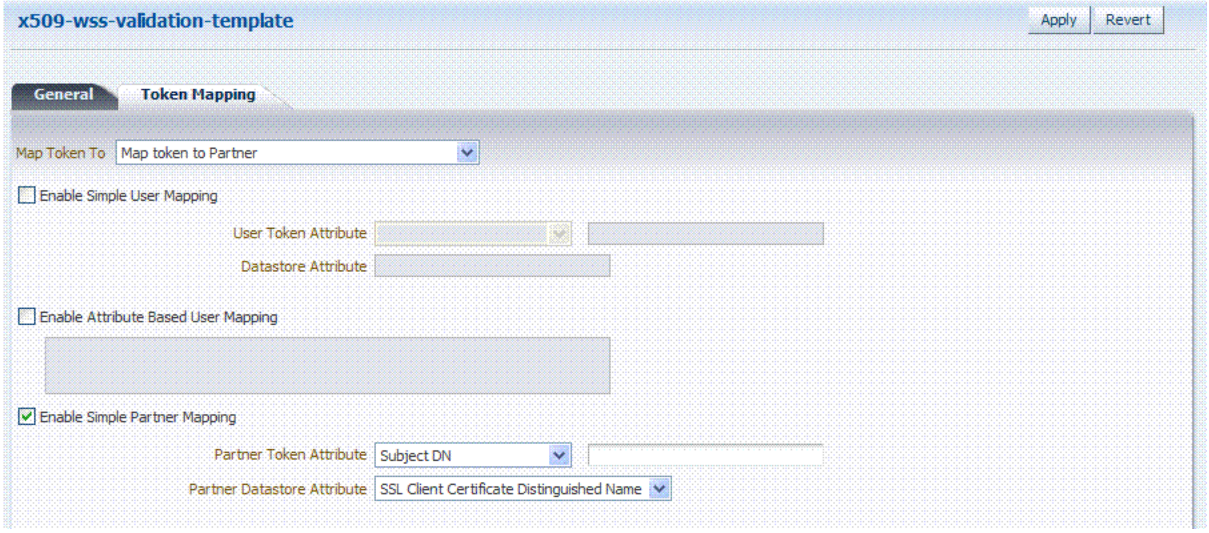
☐ Enable Attribute Based User Mapping

Figure 19–11 shows the mapping configuration settings required for Oracle Security Token Service to map the token to a requester partner entry by matching the Subject DN of the certificate to a Requester Partner that has a match on SSL Client Cert DN Identification attribute. The required settings are:

- Map Token to Partner
- Disable Simple User Mapping

- Disable Attribute Based User Mapping
- Enable Simple Partner Mapping

Figure 19–11 Token Mapping: x509-wss-validation-template



x509-wss-validation-template

Map Token To: Map token to Partner

☐ Enable Simple User Mapping

User Token Attribute

Datastore Attribute

☐ Enable Attribute Based User Mapping

☑ Enable Simple Partner Mapping

Partner Token Attribute: Subject DN

Partner Datastore Attribute: SSL Client Certificate Distinguished Name

Not all elements apply to all token types and token protocols. The elements that you must define will vary.

Table 19–11 describes the token mapping elements for validation templates.

Table 19–11 New Validation Template: Token Mapping

Element	Description
Map Token to	WS-Security Validation Template: Map Token to list ■ <empty>: no token mapping operation will occur ■ Map token to Partner: The token will be mapped to a requester partner ■ Map Token to User and map binding data to Partner: The token will be mapped to a user, and binding data (such as SSL Client Cert DN or HTTP Basic Auth Username) will be used to map the HTTP request to a requester partner ■ Map token to User: The token will be mapped to a user - - - - - - - - - - WS-Trust Validation Template: Map Token to User Check the box to enable (or clear the checkbox to disable).

Table 19–11 (Cont.) New Validation Template: Token Mapping

Element	Description
Enable Simple User Mapping	Simple user mapping consists of mapping the incoming token to a user record by using a single token attribute and matching it against a single user record attribute. WS-Security Validation Template: Only Username, SAML Assertion, Kerberos. and X.509. WS-Trust Validation Template: Username, SAML Assertion, Kerberos, X.509, OAM and custom token. The layout is different, depending on the token type of this validation template: Username Token: ■ Datastore attribute references the user record attribute that will be matched against the username element of the username token. SAML Assertion: ■ User Token attribute references an attribute from the incoming token that will be matched against the Datastore attribute (defined below) of a user record. The values can be STS_SUBJECT_ID for the NameID Value, or the name of an Attribute contained in the Assertion's AttributeStatement. ■ Datastore attribute references the user record attribute that will be matched against the User token attribute referenced above. Kerberos: ■ User Token attribute references an attribute from the incoming token that will be matched against the Datastore attribute (defined below) of a user record. The User Token Attribute can be specified by selecting one of the pre-populated attribute (Kerberos Principal, Kerberos Principal Primary or Kerberos Principal No Domain) or by entering a specific value. ■ Datastore attribute references the user record attribute that will be matched against the User token attribute referenced above. X.509: ■ User Token attribute references an attribute from the incoming token that will be matched against the Datastore attribute (defined below) of a user record. The User Token Attribute can be specified by selecting one of the pre-populated attribute (Subject DN, Common Name, Country Name, State or Province Name, Locality Name, Organizational Name, Organizational Unit Name or Domain Component) or by entering a specific value (which can be set to STS_X509_### by replacing ### with the upper case X.500 component name, for example STS_X509_CN to reference the common name component of the certificate subject). ■ Datastore attribute references the user record attribute that will be matched against the User token attribute referenced above. OAM: ■ Datastore attribute references the user record attribute that will be matched against the username element of the username token. Should be the user ID attribute defined in the Default User Identity Store. Custom: ■ User Token attribute references an attribute from the incoming token that will be matched against the Datastore attribute (defined below) of a user record. The possible values are the names of the attribute returned by the custom token validation module. ■ Datastore attribute references the user record attribute that will be matched against the User token attribute referenced above.

Table 19–11 (Cont.) New Validation Template: Token Mapping

Element	Description
Enable User Name Identifier Mapping	When enabled, define the following: WSS and WS-Trust Validation Templates will contain the same section for the Name Identifier mapping settings. A NameID user mapping operation consists of mapping the incoming SAML Assertion to a user record by mapping the NameID Value to a single user record attribute, based on the NameID format When enabled, OSTS will evaluate the NameID format, and based on the Name Identifier mapping table which user record attribute should be matched against the Name ID value contained in the Assertion. The Name Identifier mapping table holds the user record attributes to be used for the mapping operation. It contains standard NameID formats, but it can be customized to define custom Name ID formats. To add custom NameID format, click the add button on the Name Identifier mapping table, and enter the custom URI. To set an attribute for a specific NameID format to be used for mapping operation, set the user record attribute on the line for that format.

Table 19–11 (Cont.) New Validation Template: Token Mapping

Element	Description
Enable Attribute Based User Mapping	WSS Validation Template: only Username, SAML Assertion, Kerberos and X.509. WS-Trust Validation Template: only Username, SAML Assertion, Kerberos, X.509 and custom token An Attribute Based User Mapping operation consists of mapping the incoming token to a user record by using an LDAP query and token attributes. The format of the LDAP query defines the mapping rule and specifies the token attributes to be used by their names, surrounded by the percent (%) character. For example, an LDAP query that will map a token based on two token attributes (firstname and lastname) would be (&(sn=%lastname)(givenname=%firstname%)). The possible token attributes depend on the token type. Username Token ■ STS_SUBJECT_ID is the only available token attribute containing the username element of the Username token. SAML Assertion ■ STS_SUBJECT_ID contains the NameID Value. ■ STS_NAMEID_FORMAT contains the NameID Format ■ STS_NAMEID_QUALIFIER contains the NameID Qualifier ■ STS_SAML_ASSERTION_ISSUER contains the Issuer of the Assertion ■ Attributes present in the Assertion's AttributeStatement Kerberos ■ STS_KERBEROS_PRINCIPAL_SHORT contains the Kerberos Principal attribute. ■ STS_KERBEROS_PRINCIPAL_FULL contains the Kerberos Principal Primary attribute ■ STS_KERBEROS_PRINCIPAL_NODOMAIN contains the Kerberos Principal No Domain attribute X.509 ■ STS_SUBJECT_ID contains the Subject DN. ■ STS_X509_CN contains the Common Name ■ STS_X509_C contains the Country Name ■ STS_X509_ST contains the State or Province Name ■ STS_X509_L contains the Locality Name ■ STS_X509_O contains the Organizational Name ■ STS_X509_OU contains the Organizational Unit Name ■ STS_X509_DC contains the Domain Component Custom Token ■ The possible values are the names of the attribute returned by the custom token validation module.

Table 19–11 (Cont.) New Validation Template: Token Mapping

Element	Description
Enable Simple Partner Mapping	Only for WSS Validation Template and for the following token types: Username, SAML Assertion, Kerberos, and X.509. A simple partner mapping operation consists of mapping the incoming token to a partner requester by using a single token attribute and matching it against a partner identification attributes. The layout is different, depending on the token type of this validation template Username Token ■ Partner Datastore attribute references the partner identification attribute that will be matched against the username element of the username token. SAML Assertion ■ Partner Token attribute references an attribute from the incoming token that will be matched against the Partner Datastore attribute (defined below) of a Requester Partner. The values can be STS_SUBJECT_ID for the NameID Value, or the name of an Attribute contained in the Assertion's AttributeStatement. ■ Partner Datastore attribute references the partner identification attribute that will be matched against the Partner token attribute referenced above Kerberos ■ Partner Token attribute references an attribute from the incoming token that will be matched against the Partner Datastore attribute (defined below) of a requester partner. The Partner Token Attribute can be specified by selecting one of the pre-populated attribute (Kerberos Principal, Kerberos Principal Primary or Kerberos Principal No Domain) or by entering a specific value. ■ Partner Datastore attribute references the partner identification attribute that will be matched against the Partner token attribute referenced above X.509 ■ Partner Token attribute references an attribute from the incoming token that will be matched against the Partner Datastore attribute (defined below) of a requester partner. The Partner Token Attribute can be specified by selecting one of the pre-populated attribute (Subject DN, Common Name, Country Name, State or Province Name, Locality Name, Organizational Name, Organizational Unit Name or Domain Component) or by entering a specific value (which can be set to STS_X509_### by replacing ### with the upper case X.500 component name, for example STS_X509_CN to reference the common name component of the certificate subject). ■ Partner Datastore attribute references the partner identification attribute that will be matched against the Partner token attribute referenced above.

Table 19–11 (Cont.) New Validation Template: Token Mapping

Element	Description
Enable Partner Name Identifier Mapping	When enabled, defines the following only for WSS Validation Template and for SAML token types: A NameID user mapping operation consists of mapping the incoming SAML Assertion to a user record by mapping the NameID Value to a single requester partner identification attribute, based on the NameID format. When enabled, Oracle Security Token Service will evaluate the NameID format, and based on the Name Identifier mapping table which partner identification attribute should be matched against the Name ID value contained in the Assertion. The Name Identifier mapping table holds the requester partner identification attributes to be used for the mapping operation. It contains standard NameID formats, but it can be customized to define custom Name ID formats. To add custom NameID format, click the Add button on the Name Identifier mapping table, and enter the custom URI. To set an attribute for a specific NameID format to be used for mapping operation, set the requester partner identification attribute on the line for that format.

19.5.2 Managing Token Validation Templates

This is a server side configuration. A default Token Validation Template exists. Users with valid Administrator credentials can use can use the procedure in this section to add, find, edit, or delete token validation templates. Skip any steps that you do not need.

The Oracle Security Token Service Endpoint must be linked to a WS Security Validation Template that indicates:

- how to validate the token in the Webservice Security header
- how to map the token and binding data to a Requester

The information here can be applied when you want to validate the following:

- WS-Security tokens present in the SOAP Header, of type: Username, SAML 1.1, SAML 2.0, X.509 and Kerberos.
- WS-Trust tokens present in the OnBehalfOf element or in the ValidateTarget element of the WS-Trust request, of type: Username, SAML 1.1, SAML 2.0, X.509, Kerberos, OAM Session Propagation Token and custom tokens.

The following procedure includes several examples of input following specific parameters. Also, a brief translation appears within parentheses (). For instance: Name (username-token): `email-wstrust-valid-temp`. Values in your environment will be different.

Prerequisites

See Also:

- "About Managing Token Validation Templates"
- "Searching for an Existing Template"

To manage token validation templates

1. Display the list of existing Token Validation Templates.

Oracle Access Manager Console
  System Configuration
    Security Token Services
      Token Validation Templates

2. **New Token Validation Template**:

   a. Click the New Validation Template button in the upper-right corner (or click the Add (+) command button above the Search Results table).

   b. **General**: Define parameters for this template (Table 19–9). For example:

      Name (username-token): `email-wstrust-valid-temp`
      Token Protocol (WS-Security for token protocol): Webservice
      Token Type (username): `email`
      Default Partner Profile: `requester-profile`

   c. **Authentication**: Enable Credential Validation for this template, if needed, and provide details (Table 19–10). If the token type is username, enable credential validation if needed for this template and provide the details.

   d. **Token Mapping**: Specify preferences for this template based on your token type (Table 19–11).

   e. Click Save and dismiss the confirmation window (or click Cancel without saving it).

   f. Close the definition (or edit it as described in Step 4).

3. **Find an Existing Template**: From the Security Token Service section of the System Configuration tab:

   a. **Find All**: Double-click the Token Validation Templates node and review the results table.

   b. **Refine Search Results**: Specify your search criteria (Table 19–1), click the Search Button, and review the results table.

   c. **Reset the Search Form**: Click the Reset button.

4. **Edit a Template**: Start with the saved page you just created.

   Alternatively: Use Step 3 to find the desired template and click the name in the Search Results table to display the definition.

   a. Edit the template definition as needed.

   b. Click the Apply button at the top of the page to submit changes (or click Revert to undo your changes).

5. **Remove a Token Validation Template**:

   a. Click the desired name in the Search Results table to select the item to remove.

   b. From the Actions menu, click Delete (or click the Delete (X) command button above the table.

   c. Click the Delete button in the Confirmation window (or click No to cancel the operation).

19.6 Managing Oracle Security Token Service Endpoints

An endpoint is a Web Service published by Oracle Security Token Service where clients can send WS-Trust requests over SOAP. An endpoint is:

- Protected by a WS Security Policy.
- Bound to WSS Validation Template that will indicate how to validate the security token and how to map it.
- Specific to a token type, namely, the one specified in the WSS Validation Template.

> **Note:** The WS-Security policy protecting the endpoint must be compatible with the WSS Validation Template bound to the endpoint.

An endpoint is a Web Service endpoint published by Oracle Security Token Service and protected by OWSM Agent. An endpoint is bound to:

- A WS-Security policy that will determine the WSS requirements in terms of message protection and security tokens
- A WSS Validation template that will indicate how the request will be processed, how the security token will be validated.

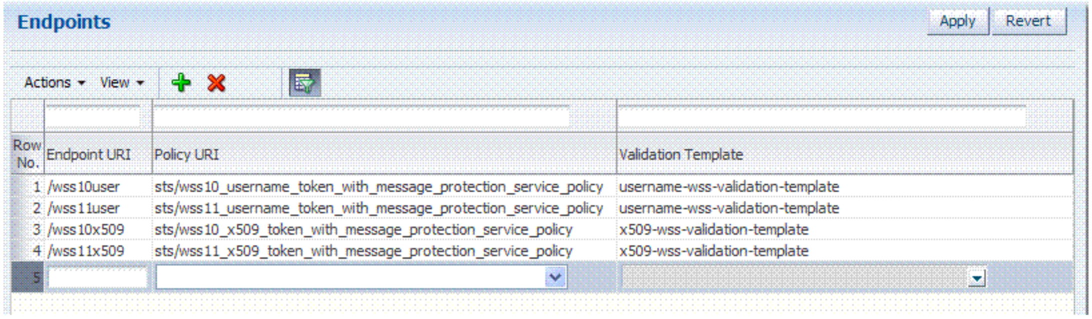
This section provides the following information:

- About Managing Endpoints
- Managing EndPoints

19.6.1 About Managing Endpoints

Oracle Security Token Service Endpoint definitions consist of three categories, as shown in Figure 19–12.

Figure 19–12 Endpoints Page



Row No.	Endpoint URI	Policy URI	Validation Template
1	/wss10user	sts/wss10_username_token_with_message_protection_service_policy	username-wss-validation-template
2	/wss11user	sts/wss11_username_token_with_message_protection_service_policy	username-wss-validation-template
3	/wss10x509	sts/wss10_x509_token_with_message_protection_service_policy	x509-wss-validation-template
4	/wss11x509	sts/wss11_x509_token_with_message_protection_service_policy	x509-wss-validation-template
5			

Table 19–12 describes the required Endpoints categories.

Table 19–12 Endpoints Page

Elements	Description
Endpoint URI	The path to the Endpoint, relative to the Oracle Security Token Service base URL The Oracle Security Token Service base URL is /sts.

Table 19–12 (Cont.) Endpoints Page

Elements	Description

Policy URI	Choose from a listing of Oracle WSM policies the one used to protect this Endpoint.
	OAM Administrator can add a new custom policy to the available listing.To show this newly created Policy URI in the the endpoints table list, use the following wlst command to update the owsmpolicies map:
	`putStringProperty("/stsglobal/owsmpolicies/<index>", "<newcustom_policypath>)`
	For example:
	`putStringProperty("/stsglobal/owsmpolicies/31", "sts/newcustom_policy")`

Validation Template ID	Choose from a listing of Validation Template names to identify one for use with this Endpoint.

Once an Endpoint is created, you can remove it but you cannot edit the definition.

19.6.2 Managing EndPoints

Users with valid OAM Administrator credentials can perform the following task to add, edit, or remove an Endpoint.

Prerequisites

Creating a Token Validation Template to reference

To create or delete an endpoint

1. From the Oracle Access Manager Console System Configuration tab, open the Security Token Services section.
2. Double-click the Endpoints node to display a list of existing Endpoints.
3. **New Endpoint**: see Table 19–12 and
   a. Click the Add (+) button above the table (or choose New Endpoint from the Actions menu).
   b. Enter the new Endpoint URI.
   c. Choose one of the Oracle WSM policies to protect this Endpoint.
   d. Choose the Validation Template to use with this Endpoint.
   e. Click Apply to submit the definition and dismiss the confirmation window (or click Revert to dismiss the page without submitting it).
   f. Close the page.
4. **Remove Endpoint**:
   a. Highlight a row in the Endpoints table and click the Delete (X) button (or choose Delete Selected from the Actions menu).
   b. Confirm removal (or cancel the removal).

19.7 Managing Token Issuance Policies and Constraints with Oracle Access Manager

This section provides the following topics:

- About Token Issuance Policies
- About Managing Token Issuance Policies and Constraints
- Managing Token Issuance Policies and Constraints

19.7.1 About Token Issuance Policies

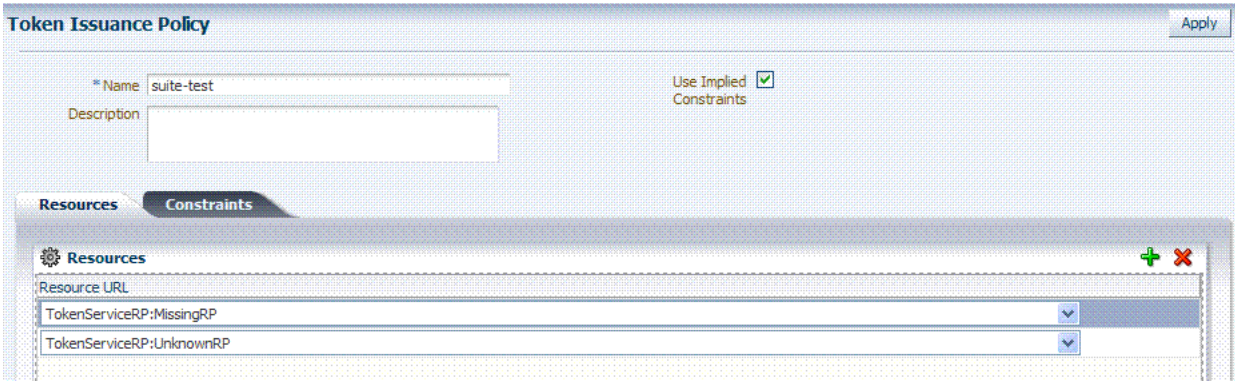
A Token Issuance Policy defines the rules under which a token can be issued for a resource (Relying Party Partner) based on the client's identity, with the client either being a Requester Partner or an end user. If a Requester is NOT present, it is assumed that the User (represented by the OBO token or WSS Token) is trying to access the RelyingParty.

When issuing a token, Oracle Security Token Service will determine for which Relying Party that token is created, and it will then evaluate if the client is authorized to request the token for that Relying Party. In order to issue a token, a Token Issuance Policy must be created with the resource involved in the operation, and with possibly a constraint. At runtime if the policy evaluation is successful, the token will be issued.

IAM Suite Token Issuance Policy

Figure 19–13 presents IAM Suite Token Issuance Policy in the IAM Suite application domain. Use Implied Constraints is checked by default, which allows access in the absence of any authorization constraints of a particular class. By default, there are no explicit constraints defined.

Figure 19–13 IAM Suite Token Issuance Policy and Resource URLs



Token Issuance Policy

* Name suite-test

Description

Use Implied Constraints ☑

Resource URL

TokenServiceRP:MissingRP

TokenServiceRP:UnknownRP

You can add constraints to this Token Issuance Policy.

19.7.2 About Managing Token Issuance Policies and Constraints

The Token Issuance Policy allows the Administrator to define "Allow" and "Deny" constraints on the policy. Each Token Issuance Policy can contain one or more constraints that determine whether access to the requested resource should be granted or denied:

- An Allow type condition specifies who is authorized to access a protected resource.

Only partners and users listed in the Constraint are granted access; everyone else is denied access to the resource.

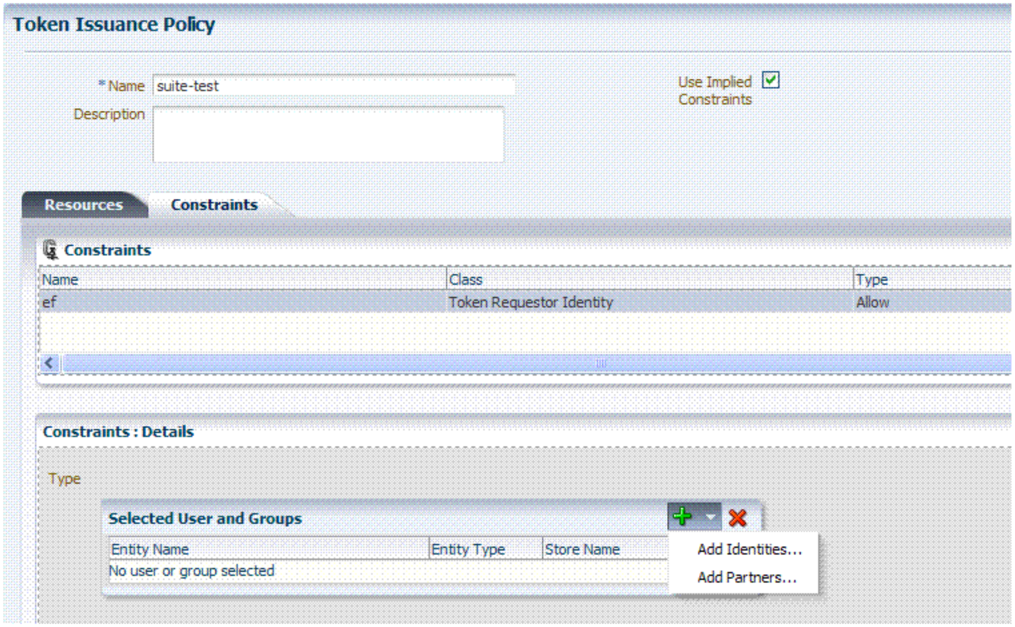
- A Deny type condition specifies explicitly who is denied access to the protected resource.

  Only partners and users listed in the Constraint are denied access; everyone else is granted access to the resource.

> **Note:** When adding User constraints, the identity store from which the users are to be chosen can be selected from a list. Ensure that you choose the Default User Identity Store, which is the only one used by Oracle Security Token Service.

Managing Token Issuance Policies and Constraints is similar to managing Authorization Policies and Constraints. Figure 19–4 shows the Constraints tab of a Token Issuance Policy.

Figure 19–14 Token Issuance Policies and Constraints



* Name: suite-test

Description:

Use Implied Constraints ☑

Name	Class	Type
ef	Token Requestor Identity	Allow

Entity Name	Entity Type	Store Name
No user or group selected		

Table 19–13 describes the Token Issuance Policy and Constraint requirements.

Table 19–13 Constraints Tab: Token Issuance Policy

Element	Description
Name	A unique name for this Token Issuance Policy.
Description	Optional.
Use Implied Constraints	Enabled by default, which allows access in the absence of any authorization constraints of a particular class.
Constraints Tab	
Name	A unique name for this Constraint, which you assign when you add a new constraint. A dialog box opens where you enter the name.

Table 19–13 (Cont.) Constraints Tab: Token Issuance Policy

Element	Description
Class	Only Token Requester Identity is allowed for Token Issuance Policy constraints. You choose this in the Add Constraint dialog box.
Type	Choose a Condition: ■ Allow: Specifies who is authorized to access a protected resource. ■ Deny: Specifies explicitly who is denied access to the protected resource.
Selected Users and Groups	
Add	Choose from the following populations: ■ Add Identities: This choice opens a Search window where you can set the Store Name, Choose an Entity Type (All, User, or Group), and Provide an Entity Name. You then choose one or more results from those listed and click Add Selected to populate the constraint. ■ Add Partners: This choice opens a Search window where you can locate specific partners to populate the constraint. Enter your search criteria (or click the arrow key beside the field to find all partners), then choose one or more results and click Add Selected to populate the constraint.
Entity Name	The name of the User or Group, as defined in the selected User Identity Store.
Entity Type	The type of entity you want to locate during a search to add identifies to the constraint: User, or Group.
Store Name	Choose the name of the User Identity Store to search for users or groups to populate the constraint. Remember, Oracle Security Token Service uses only the Default Identity Store.

19.7.3 Managing Token Issuance Policies and Constraints

Users with valid Administrator credentials can use the following procedure to add a Token Issuance Policy and Constraints to an Application Domain. When adding resources to this policy, you might want to add the UnknownRP and MissingRP resources.

Prerequisites

The application domain must already exist.

> **Note:** You can add Token Issuance Policies to the IAM Suite Application Domain.

To manage Token Issuance Policies and constraints

1. Open the Application Domain, as follows:

   Oracle Access Manager Console
   Policy Configuration
   Application Domains
   Desired Domain

Token Issuance Policy

2. **Add a Token Issuance Policy**:
   - a. In the Application Domain tree, click the Token Issuance Policies node and then click the Add (+) button to open a fresh page.
   - b. On the Token Issuance Policy page, enter a unique name and optional description.
   - c. On the Resources tab, click the Add (+) button and choose the desired resource from those listed. For example: TokenServiceRP:*URL*
   - d. Select the Token Issuance Policy to assign to this new resource
   - e. Click the Apply button at the top of the page to submit changes (or click Revert to undo your changes).
   - f. See Also: "Managing Token Issuance Policies and Constraints".
3. **Add Constraints to a Policy**:
   - a. Click the Constraints tab, then click the Add button on the Constraints tab to display the Add Constraint window.
   - b. Enter a unique name for this constraint in the dialog box.
   - c. Choose Token Requester Identity from the Class list.
   - d. Choose the Allow or Deny condition.
   - e. Click Add Selected.
4. **Add Constraints Details**:
   - a. Click the Constraint name to display Constraints: Details.
   - b. Click the Add button and choose either Add Identifies or Add Partners.
   - c. **Add Partners**: Enter your search criteria (or click the arrow key beside the field to find all partners), then choose one or more results and click Add Selected to populate the constraint.
   - d. **Add Identities**: In the Search window set the Store Name, Choose an Entity Type (All, User, or Group), and provide an Entity Name choose one or more results and click Add Selected to populate the constraint.
   - e. Click the Apply button at the top of the page to submit the policy and constraints.
5. **Find (or Add) TokenServiceRP Resources in the Application Domain**: See "Managing TokenServiceRP Type Resources".

19.8 Managing TokenServiceRP Type Resources

A Token Issuance Policy defines the rules under which a token can be issued for a resource (Relying Party Partner) based on the client's identity, with the client either being a Requester Partner or an end user.

When issuing a token, Oracle Security Token Service will determine for which Relying Party that token is created, and it will then evaluate if the client is authorized to request the token for that Relying Party.

Note: In order to issue a token, a Token Issuance Policy must be created with the resource involved in the operation and, possibly, with a constraint. At run time if the policy evaluation is successful, the token will be issued.

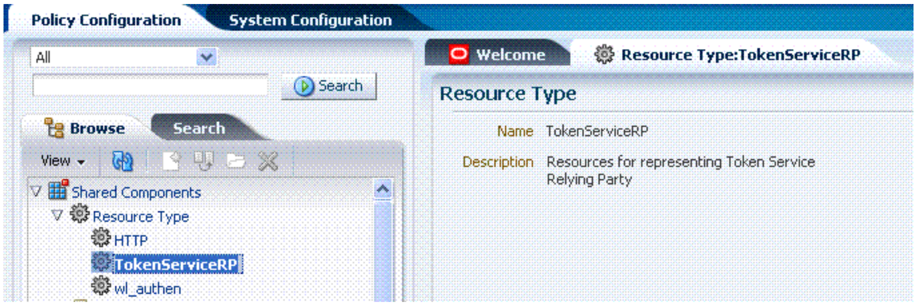
The resource(s) in a policy can be:

- A TokenServiceRP type resource based on the Relying Party Partner ID.
- The pre-existing UnknownRP resource which is needed when Oracle Security Token Service is not able to map the Service URL referenced in the `AppliesTo` element of the WS-Trust request to an Oracle Security Token Service Relying Party Partner entry.
- The pre-existing MissingRP resource which is needed when the `AppliesTo` element of the WS-Trust request is missing.

Note: Both the MissingRP and UnknownRP are defined in the IAM Suite Application Domain.

A resource of type TokenServiceRP, Figure 19–15, represents an Oracle Security Token Service Relying Party Partner defined in the Oracle Security Token Service Partner Store.

Figure 19–15 Pre-defined Resource Type: TokenServiceRP



Resource Type

Name TokenServiceRP

Description Resources for representing Token Service Relying Party

Resources of type TokenServiceRP are used in Token Issuance Policies, which are evaluated when Oracle Security Token Service issues tokens at run time.

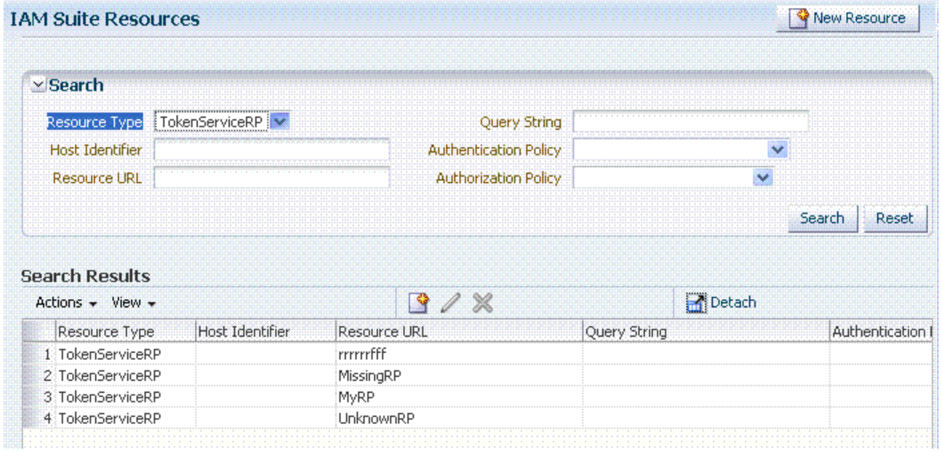
For more information, see:

- About Managing TokenServiceRP Type Resources in Oracle Access Manager
- Managing TokenServiceRP Type Resources in Application Domains

19.8.1 About Managing TokenServiceRP Type Resources in Oracle Access Manager

Use the Search controls for the application domain to locate resources of a specific type within the domain. Figure 19–16 shows the search controls for the IAM Suite resources. Resource Type TokenServiceRP is the search criteria. The Search Results table lists all resources of this type within the application domain.

Figure 19–16 Search: Resource Type TokenServiceRP in Application Domain



Resource Type: TokenServiceRP

	Resource Type	Host Identifier	Resource URL	Query String	Authentication I
1	TokenServiceRP		rrrrrrfff		
2	TokenServiceRP		MissingRP		
3	TokenServiceRP		MyRP		
4	TokenServiceRP		UnknownRP		

The TokenServiceRP resources in this domain include those provided out of the box, and described earlier:

- UnknownRP resource
- MissingRP resource

19.8.2 Managing TokenServiceRP Type Resources in Application Domains

Users with valid Administrator credentials can use the following procedure to add TokenServiceRP resources to an application domain.

Note:

- If `AppliesTo` is present in the RST but the requester could not be mapped, use the `TokenServiceRP:UnknownRP` resource.
- If `AppliesTo` is not present, use `TokenServiceRP:MissingRP`, otherwise select the appropriate resource.

See Also: "About Managing TokenServiceRP Type Resources in Oracle Access Manager"

To manage TokenServiceRP Resources

1. Open the Application Domain, as follows:

   Oracle Access Manager Console
   Policy Configuration
   Application Domains
   Desired Domain

2. **Add TokenServiceRP Resource to the Application Domain**:

   a. Click the New Resource button on the Application Domain Search page.

   b. Specify the Resource Type as TokenServiceRP.

   c. Enter a Resource URL that is the Relying Party ID for whom the token issuance policy will be defined.

   d. Click the Apply button at the top of the page to submit this and dismiss the confirmation window.
   e. See Also:"Adding Resource Definitions to an Application Domain" on page 13-18.

3. **Find TokenServiceRP Resources**:
   a. Open the Resources node to display the Search controls.
   b. From the Resource Type list, choose TokenServiceRP, and click Search.
   c. Review the Search Results table and click a name to open the Resource Definition.

20

Managing Token Service Partners and Partner Profiles

This chapter provides the following topics describing management of Token Service Partners and Partner Profiles:

- Prerequisites
- Introduction Token Service Partners and Partner Profiles
- Managing Token Service Partners
- Managing Token Service Partner Profiles

20.1 Prerequisites

"Introduction to Oracle Security Token Service" on page 1-10

Chapter 16, "Oracle Security Token Service Implementation Scenarios"

Any task you can perform using the Oracle Access Manager Console can also be performed using the

> **See Also:** *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

20.2 Introduction Token Service Partners and Partner Profiles

This section provides the following topics:

- About Token Service Partners
- About Partner Profiles
- About Partner and Profile Data

20.2.1 About Token Service Partners

A Token Service partner represents a partner trusted by the STS Server. There are three types of partners:

- Requester, which represents a Web Service Client interacting directly with Oracle Security Token Service in order to issue or validate tokens
- Relying Party, which references a Web Service Provider that will be the recipient of tokens issued by the Oracle Security Token Service server.

- Issuing Authority, which represents an Assertion issuer. When validating an Assertion, its issuer must be a known Issuing Authority Partner entry in Oracle Security Token Service.

The Security Token Service is capable of interacting with two types of clients:

- Web Service Client modules, defined as requester partners in Oracle Security Token Service (typically SOAP clients)
- End users, not defined as requester partners, but possibly present in the User Identity Store.

20.2.2 About Partner Profiles

A Partner Profile contains configuration properties that are common to a set of partners, and each partner entry is associated to a Partner Profile. Similar to the partners, there are three types of partner profiles: Requester, Relying Party and Issuing Authority Partner Profiles.

20.2.3 About Partner and Profile Data

A partner entry will contain the following information:

- Signing and Encryption Certificates
- Partner name, unique identifier and description
- Reference to a Partner Profile
- For Requester, also contains Username Token credentials, and Identification strings used to map incoming data to a requester.

A partner profile entry will contain the following information, depending on the type of profile:

- Requester
  - Claims Mappings
  - WS-Trust Validation Templates used to validate tokens present in the OnBehalfOf element
- Relying Party
  - Attributes to be sent to RP
  - Issuance Templates to be used
- Issuing Authority
  - Attribute Name/Value Mapping settings
  - Specific Mapping Actions Rules used to map an incoming token to a partner/user

20.3 Managing Token Service Partners

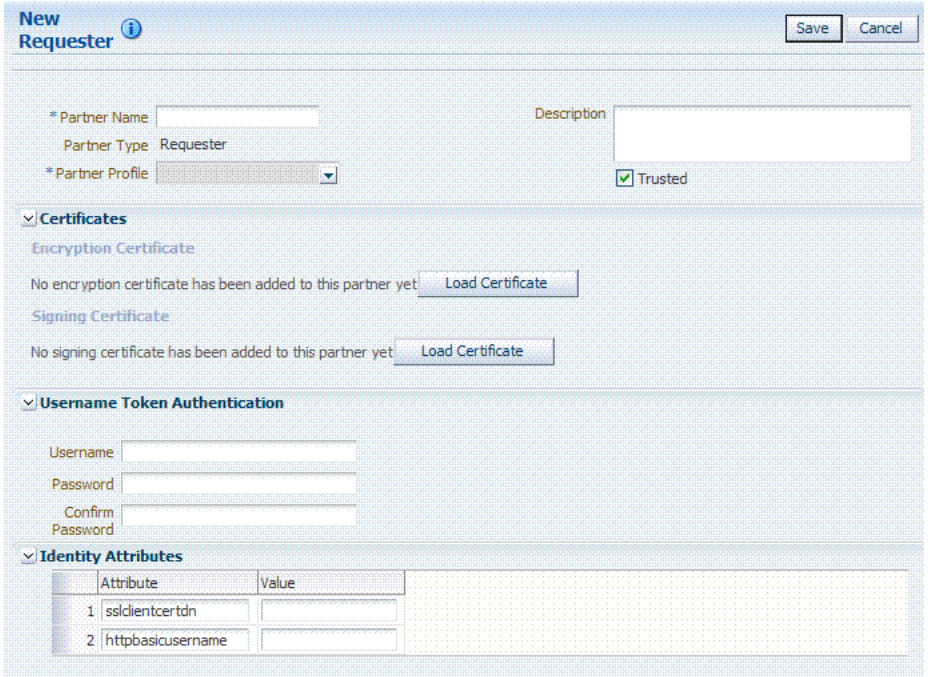
This section provides the following topics.

- About Managing Token Service Partners
- Managing a Token Service Partner
- Refining Partner Searches

20.3.1 About Managing Token Service Partners

When you choose to create a new partner, a fresh page appears for the specific Partner Type you selected. Figure 20–1 shows the New Requester partner page in the Oracle Access Manager Console, which includes all Partner elements.

Figure 20–1 New Requester Partner Page



* Partner Name

Partner Type Requester

* Partner Profile

Description

☑ Trusted

Certificates

Encryption Certificate

No encryption certificate has been added to this partner yet

Signing Certificate

No signing certificate has been added to this partner yet

Username Token Authentication

Username

Password

Confirm Password

Identity Attributes

	Attribute	Value
1	sslclientcertdn	
2	httpbasicusername	

While most elements are common to all partners (name, description, and whether this partner is trusted), certain elements depend upon the specific partner type. For instance:

- Requester partners: Can specify an encryption certificate and a signing certificate.
- Relying Party partners: Can specify only an encryption certificate
- Issuing Authority partners: Can specify only a signing certificate

Table 20–1 describes elements for Oracle Security Token Service partners. Unless explicitly stated otherwise, all elements apply to every partner type.

Table 20–1 Elements for Oracle Security Token Service Partners

Element	Description
Partner Name	Enter a name for this partner.
Partner Type	Uneditable description, depending upon the type of partner you are creating or editing: ■ Requester ■ Relying Party ■ Issuing Authority
Partner Profile	Choose from those listed for your chosen partner type.
Description	Optional.

Table 20–1 (Cont.) Elements for Oracle Security Token Service Partners

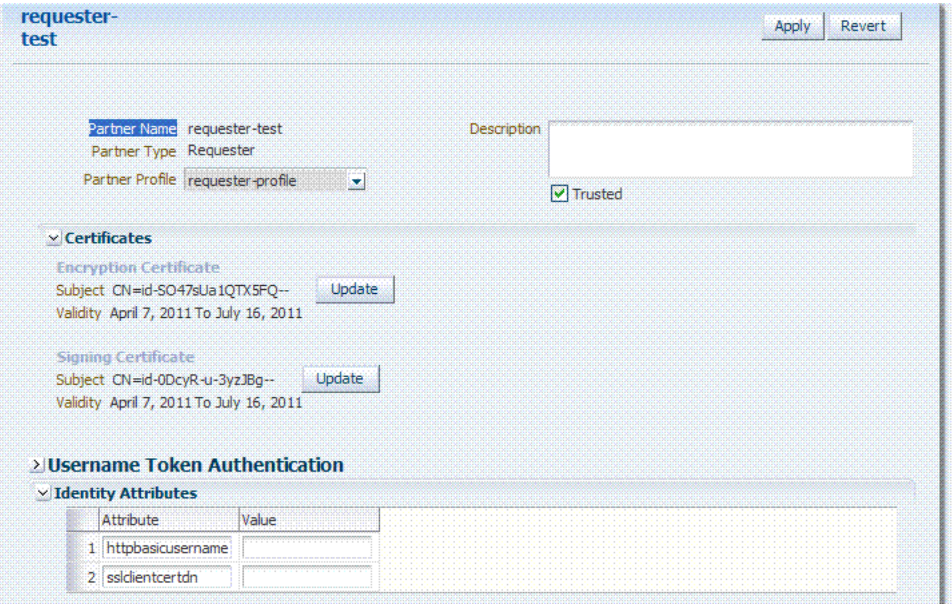
Element	Description
Trusted	Check this box to indicate whether or not the partner is trusted. If not checked, the Oracle Security Token Service server will report an error when a request involves such an entry.
Load Certificate	Browse for and upload the requested certificates, which depend on partner type: ■ Encryption and signing certificates ■ Encryption certificate ■ Signing certificate
Username Token Authentication *Requester only*	Values can be entered for the following for Username Token Authentication: ■ Username ■ Password ■ Confirm Password New Requester Partner Identification Attributes can be defined in the STS Settings section and will appear in the requester partner Identity Attributes table. Note: the username and password data will be used to validate the credentials of a username token. It is also possible to only enter a username and no password, when the data will be used only to map an incoming token to this requester partner using the username.
Identity Attributes *Requester only*	At runtime, Oracle Security Token Service will use the data defined in the section to map an incoming request to a requester partner entry, using: ■ The token data or binding data such as the SSL Client Certificate's Subject DN if present, or HTTP Basic Authentication username. ■ The identity attributes present in each requester partner entry. New mappings can be added in the Relying Party Partner section as follows: http://*relying.party.test.com/testing.service*. At runtime, the Oracle Security Token Service server will use those URLs to map the AppliesTo service location contained in a WS-Trust request to a Relying Party Partner.

Table 20–1 (Cont.) Elements for Oracle Security Token Service Partners

Element	Description
Resource URL *Relying Party only*	Enter the resource URL in the resource pattern column of the table, and enter a description beside it. For instance: Pattern: http://*relying.party.test.com/testing/service* The resource URL listed in the table will be used when mapping the AppliesTo location element from the WS-Trust request to this Relying Party Partner. The AppliesTo location value will be mapped to this Relying Party Partner: ■ A Resource URL matches exactly the AppliesTo location value. For example, the AppliesTo location is http://*relying.party.test.com/testing/service* and the Resource URL is also http://*relying.party.test.com/testing/service*). ■ Or, a Resource URL is the parent of the AppliesTo location value. For example, the AppliesTo location is http://*relying.party.test.com/testing/service* and the Resource URL is http://*relying.party.test.com/testing*, or Resource URL is http://*relying.party.test.com/*

Figure 20–2 shows a Requester Partner page that is filled in.

Figure 20–2 Defined Requester Partner



requester-test

Partner Name requester-test

Partner Type Requester

Partner Profile requester-profile

Description

☑ Trusted

Certificates

Encryption Certificate

Subject CN=id-SO47sUa1QTX5FQ--

Validity April 7, 2011 To July 16, 2011

Signing Certificate

Subject CN=id-0DcyR-u-3yzJBg--

Validity April 7, 2011 To July 16, 2011

Username Token Authentication

Identity Attributes

	Attribute	Value
1	httpbasicusername	
2	sslclientcertdn	

20.3.2 Managing a Token Service Partner

Users with valid Administrator credentials can use the following procedure to create, find, edit, or delete a token service partner using Oracle Access Manager Console.

Prerequisites

A partner profile of the type of partner that will be created needs to be exist.

To manage a token service partner

1. From the Oracle Access Manager Console, open the:

   System Configuration tab
   Security Token Services section
   Partners node

2. Under the Partners node, double-click the desired partner type and proceed with following steps as needed.

   - Requesters
   - Relying Parties
   - Issuing Authorities

3. **New Partner**:

   a. Click the New *PartnerType* button to display a fresh page for your definition.

   b. Enter general information for the chosen partner type (Table 20–1).

   c. **Trusted**: Click to select (or leave blank if this is not a trusted partner).

   d. **Certificates**: Load any necessary certificates.

   e. **Relying Party**: Enter Resource URLs, if needed.

   f. **Requester**: Enter Username Token credentials, if needed.

   g. Click Save to submit (or click Cancel to dismiss the page) and then dismiss the confirmation window.

4. **Refine a Partner Search**: "Refining Partner Searches"

   a. Perform Steps 1 and 2.

   b. Define your query and click the Search button.

   c. In the Search Results table, click the name of partner to view, edit, or remove.

5. **Edit a Partner**:

   a. In the Search Results table, click the name of partner to edit and click the Edit button (or choose Edit from the Actions menu).

   b. Make desired changes to partner information (Table 20–1).

   c. Click Apply to submit the changes (or Revert to cancel changes) and then dismiss the confirmation window.

6. **Remove a Partner**: Use the Search controls to refine and submit your query, as needed.

   a. In the Search Results table, highlight the row containing the partner to remove.

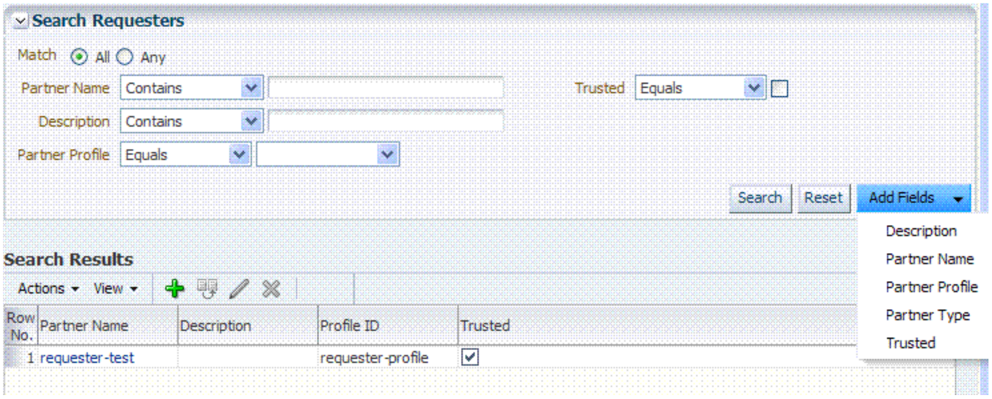
   b. Click the Delete (X) button (or choose Delete Selected from the Actions menu), then dismiss the confirmation window.

20.3.3 Refining Partner Searches

As with other System Configuration components, when you open the Partner node, all Partner type nodes become available. When you choose a specific Partner node, relevant Search controls, and the Search Results table, become available.

Figure 20–3 illustrates a Requester Partner, where only the results differ from that of other Partner Types.

Figure 20–3 Partner Search Controls



Match ● All ○ Any

Partner Name Contains

Description Contains

Partner Profile Equals

Trusted Equals □

Row No.	Partner Name	Description	Profile ID	Trusted
1	requester-test		requester-profile	☑

From the Search page you can simply select a name in the Search Results table, or use the controls to refine your search to locate a specific Partner or Partners with specific characteristics.

20.4 Managing Token Service Partner Profiles

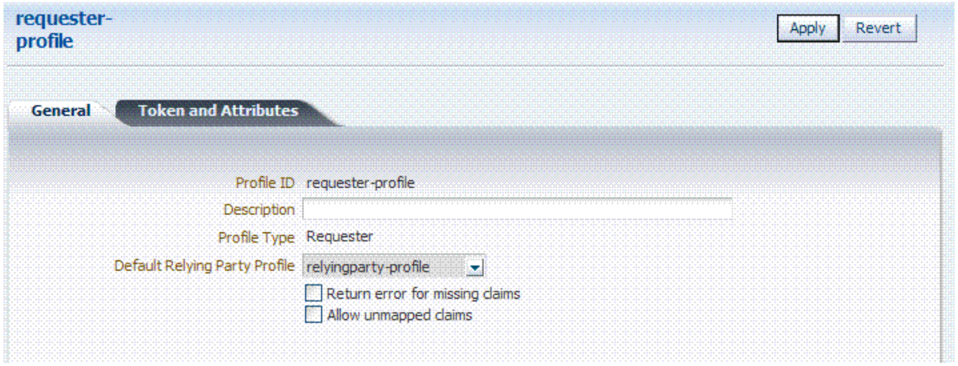
This section provides information about Token Service Partner Profiles.

- About Managing Partner Profiles
- Managing a Token Service Partner Profile
- Refining a Profile Search

20.4.1 About Managing Partner Profiles

Figure 20–4 shows a completed Requester Profile page, with both a General tab and Token and Attributes tab.

Figure 20–4 Requester Profile: General



Profile ID requester-profile

Description

Profile Type Requester

Default Relying Party Profile relyingparty-profile

□ Return error for missing claims

□ Allow unmapped claims

Table 20–2 describes the General elements for all profile types.

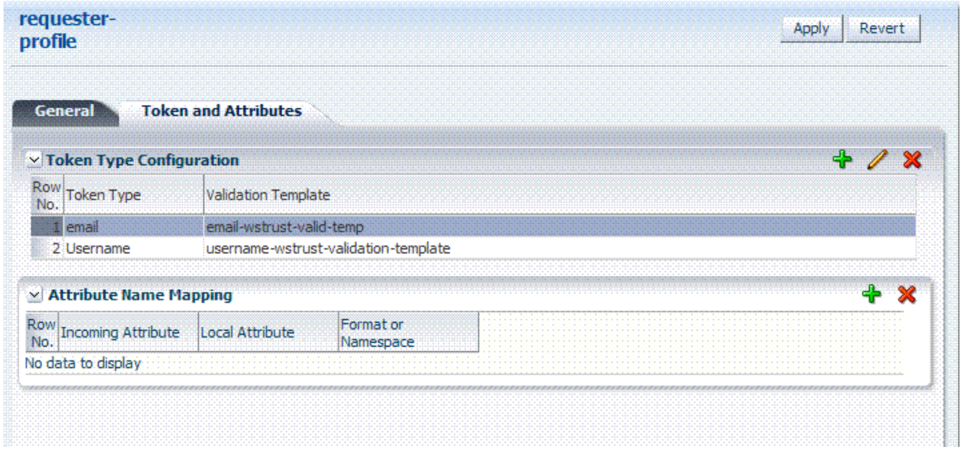
Table 20–2 Profile: General

Element	Description
Profile ID	A unique identifier for this profile
Description	Optional.
Profile Type	Type of profile, which cannot be edited: Requester, Relying Party or Issuing Authority.
Default Relying Party Profile *Requester Partner Profile Only*	References the Relying Partner Profile to use, if the WS-Trust request does not reference the Relying Party (for example, the AppliesTo element is missing), or if the AppliesTo element could not be mapped to a known Relying Party Partner Profile. Choose a Relying Party profile to use as the default and enable or disable the following characteristics as needed: ■ Return error for missing claims. Indicates whether or not Oracle Security Token Service will return an error if the issued token does not contain claims that were requested by the client. Since the Relying Party Partner Profile defines the list of attributes/claims that can be included in the issued token, it is possible that some claims requested by the client cannot be returned. ■ Allow unmapped claims. Claims listed in a WS-Trust request are specified in a dialect that will be translated to map to local attributes using the Token and Attributes section. This flag indicates whether or not claims that cannot be translated should be referenced as is. This allows to control which claims can be requested by the client.
Default Token to Issue *Relying Party Only*	This table indicates which Issuance Template to use to issue a token for Relying Parties linked to this profile. Choose a token type as the default for this profile: ■ SAML 1.1 ■ SAML 2.0 ■ Username ■ Custom Check the box beside Download Policy to associate a policy with the token. When checked, Oracle Security Token Service will download at runtime the WS-Security policy of the Relying Party referenced by the AppliesTo element in the RST. If present, Oracle Security Token Service will use that URL to download the policy, and then determine the type of token to return based on the information located in the policy.

Requester Profile: Token and Attributes

Figure 20–5 illustrates the Token and Attributes tab and accompanying tables for the Requester profile. The Token Type Configuration section indicates which WS-Trust Validation Template to use to validate tokens contained in the OnBelhalfOf element of the WS-Trust request, based on the token type. This section defines mappings between WS-Trust claims requested by the client and local attribute names

Figure 20–5 Requester Profile: Token and Attributes



Token Type Configuration

Row No.	Token Type	Validation Template
1	email	email-wstrust-valid-temp
2	Username	username-wstrust-validation-template

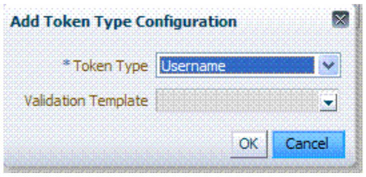
Attribute Name Mapping

Row No.	Incoming Attribute	Local Attribute	Format or Namespace

No data to display

Table 20–3 describes Requester Profile Token and Attributes elements and controls.

Table 20–3 Requester Profile: Token and Attributes

Element	Description
Token Type Configuration	Click the + above the table to display the following dialog box and then make one selection from each list:  Add Token Type Configuration * Token Type: Username Validation Template: OK Cancel Token Type list provides all supported (and custom) token types deployed. Validation Template list contains all currently defined WS-Trust validation templates.
Attribute Name Mapping	This table defines how OSTS will map a claim, represented by its name and optional Format/Namespace, to a local attribute. Oracle Security Token Service supports the Infocard Claims dialect. To translate Infocard claims to local attributes, a mapping will need to be defined where the Incoming Attribute will contain the claim name and the Local Attribute will contain the local name (The Format/Namespace column will be empty). For example, one mapping could be: ■ Incoming Attribute: surname ■ Local Attribute: sn Another mapping could be: ■ Incoming Attribute: givenname ■ Local Attribute: givenname Another mapping could be: ■ Incoming Attribute: emailaddress ■ Local Attribute: mail

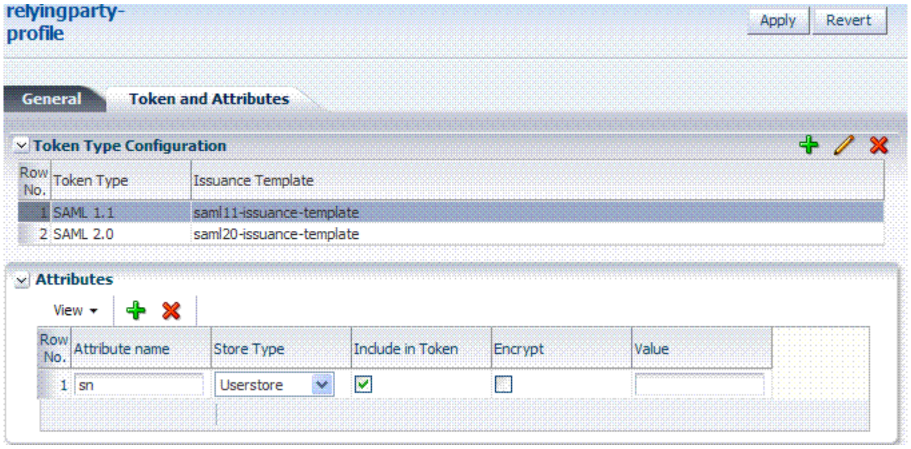
Relying Party Profile: Token and Attributes

Figure 20–6 illustrates the Token and Attributes defined for a Relying Party Profile. This section allows the administrator to define which Issuance Template should be used to issue a token for a Relying Party associated with this profile.

Also, it lists the attributes that might be included in an issued token, by their names, the source of those attributes, and whether or not the attributes should be included in the issued token only if requested by the client or always.

On this page, Relying Party Profiles require an Issuance Template in addition to the token type. Also, the attribute types differ from other profiles.

Figure 20–6 Relying Party Profile Token and Attributes



Row No.	Token Type	Issuance Template
1	SAML 1.1	saml11-issuance-template
2	SAML 2.0	saml20-issuance-template

Row No.	Attribute name	Store Type	Include in Token	Encrypt	Value
1	sn	Userstore	☑	☐	

Table 20–4 describes the elements needed for the Relying Party Profile.

Table 20–4 Relying Party Profile Requirements

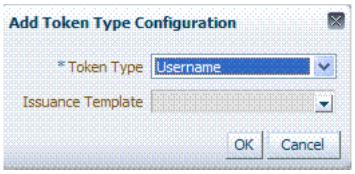
Element	Descrip[tion
Token Type Configuration	Click the + above the table to display the following dialog box and then make one selection from each list:  Add Token Type Configuration * Token Type: Username Issuance Template: OK Cancel Token Type list provides all supported (and custom) token types deployed. Issuance Template list contains all currently defined Issuance Templates.

Table 20–4 (Cont.) Relying Party Profile Requirements

Element	Descrip[tion
Attributes	The attributes that might be included in an issued token: ▪ Attribute Name: Indicates the name of the attribute. ▪ Store Type: Indicates the source of the attribute: Userstore: the default User Identity Store where the LDAP user record will be used to retrieve the attribute value. Incoming Token: the attribute will reference an element of the incoming token. Static: the value will be specified in the Value field. ▪ Include in Token: Indicates whether or not the attribute should always be included in the issued token. If unchecked, the attribute will only be included if the client requested this attribute. ▪ Encrypt: Indicates whether or not the attribute should be encrypted. Note: only supported for SAML 2.0. Also, the Encryption Certificate must be set in the Relying Party Partner entry, or it must be present in the WS-Trust request. ▪ Value: Contains the value to be used if the Store Type is static value. See Also: "Relying Party Profile Attributes".

Relying Party Profile Attributes

When defining an attribute, you can indicate:

- The attribute source: User Store (LDAP), Incoming Token Data or static value.
- Whether or not to include the attribute in the token only if requested by the client or in all tokens.
- Whether or not to encrypt the attribute (only SAML 2.0; requires the Relying Party Encryption Certificate).
- The value of the attribute if this is a static attribute.

Example: To include the mail attribute retrieved from LDAP in all outgoing tokens:

- Attribute Name: mail
- Store Type: User Store
- Include in Token: checked
- Encrypt: unchecked
- Value: empty

Example: To include the username element of an incoming Username Token in all outgoing tokens

- Attribute Name: STS_SUBJECT_ID
- Store Type: Incoming Token
- Include in Token: checked
- Encrypt: unchecked
- Value: empty

Example: To include a static attribute in all outgoing tokens:

- Attribute Name: rp-version
- Store Type: Static
- Include in Token: checked
- Encrypt: unchecked
- Value: 2.0

The following attributes are available from the incoming token data. The SAML attributes referenced by their names are also available as incoming token data:

STS_SUBJECT_ID
Contains the subject identifier (username for Username token, NameID Value for SAML assertions, Subject DN for X.509 certificates)

STS_NAMEID_FORMAT
Contains the SAML NameID Format.

STS_NAMEID_QUALIFIER
Contains the SAML NameID Format.

STS_SPNAME_QUALIFIER
Contains the SAML NameID Qualifier.

STS_SP_PROVIDED_ID
Contains the SAML NameID SP Qualifier

STS_SESSION_INDEX
Contains the session index.

STS_AUTHENTICATION_INSTANT
Contains the authentication instant (current after Username token credentials validation, from the authentication statement for SAML Assertions, current for X.509 validation, current for Kerberos Validation, authentication instant for OAM Session Propagation tokens).

STS_AUTHENTICATION_TIMEOUT
Contains the session expiration time if set (applies to SAML assertions and OAM Session Propagation tokens if present).

STS_X509_CN
Contains the CN component of the X.509 Certificate's Subject DN

STS_X509_OU
Contains the OU component of the X.509 Certificate's Subject DN.

STS_X509_O
Contains the O component of the X.509 Certificate's Subject DN.

STS_X509_L
Contains the L component of the X.509 Certificate's Subject DN.

STS_X509_ST
Contains the ST component of the X.509 Certificate's Subject DN.

STS_X509_C
Contains the C component of the X.509 Certificate's Subject DN.

STS_X509_DC
Contains the DC component of the X.509 Certificate's Subject DN.

STS_X509_*
Contains the component identified by * of the X.509 Certificate's Subject DN.

STS_X509_VERSION
Contains the version attribute of the X.509 Certificate.

STS_X509_ISSUER_X500_PRINCIPAL_NAME
Contains the issuer DN of the X.509 Certificate.

STS_X509_NOT_AFTER
Contains the not after attribute of the X.509 Certificate.

STS_X509_NOT_BEFORE
Contains the not before attribute of the X.509 Certificate.

STS_X509_SUBJECT_X500_PRINCIPAL_NAME
Contains the subject DN of the X.509 Certificate.

STS_X509_SUBJECT_ALTERNATIVE_NAMES
Contains the subject alternative name extension value of the X.509 Certificate.

STS_X509_SERIAL_NUMBER
Contains the serial number of the X.509 Certificate.

STS_OAM_LAST_ACCESS_TIME
Contains the last access time of the OAM Session Propagation Token.

STS_OAM_LAST_UPDATE_TIME
Contains the last update time of the OAM Session Propagation Token.

STS_OAM_CREATION_TIME
Contains the creation time of the OAM Session Propagation Token.

STS_KERBEROS_PRINCIPAL_SHORT
Contains the Principal Short value of the Kerberos Token.

STS_KERBEROS_PRINCIPAL_FULL
Contains the Principal Full value of the Kerberos Token.

STS_KERBEROS_PRINCIPAL_NODOMAIN
Contains the Principal No Domain value of the Kerberos Token.

STS_SAML_ASSERTION_ID
Contains the AssertionID of the SAML Assertion.

STS_SAML_SUBJECT_DNS
Contains the Subject DNS attribute of the SAML Assertion.

STS_SAML_SUBJECT_IP_ADDRESS
Contains the Subject IP Address attribute of the SAML Assertion.

STS_SAML_ASSERTION_ISSUER
Contains the Issuer of the SAML Assertion.

STS_SAML_AUTHN_INSTANT
Contains the authentication instant of the SAML Assertion.

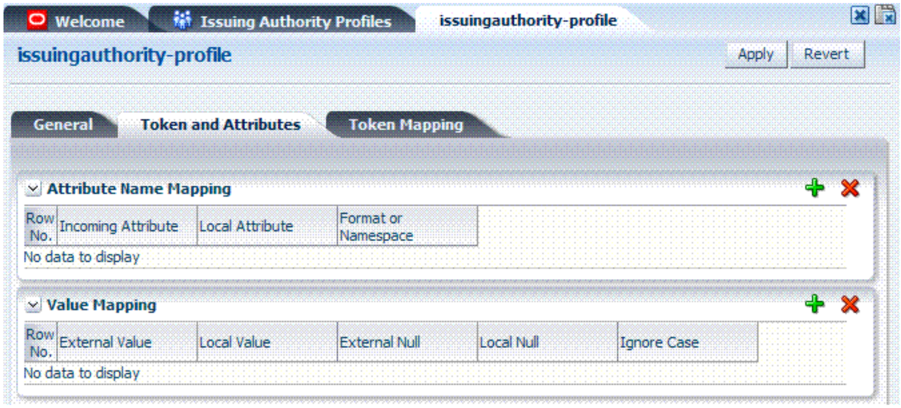
STS_SAML_AUTHN_METHOD
Contains the authentication method of the SAML Assertion.

Issuing Authority Profile: Token and Attributes

The Issuing Authority Partner Profile defines settings that can be common to different Issuing Authority Partners.

The Token and Attributes section, as shown in Figure 20–7, allows the administrator to define mapping rules that will be used to translate the name and value of attributes to local names and values.

Figure 20–7 Token and Attributes: Issuing Authority



Attribute Name Mapping

Row No.	Incoming Attribute	Local Attribute	Format or Namespace

No data to display

Value Mapping

Row No.	External Value	Local Value	External Null	Local Null	Ignore Case

No data to display

Table 20–5 describes the Token and Attributes elements for Issuing Authority. It is possible to define attribute mapping rules that will be applied to the attributes included in the Assertion, when extracting them from the token. There are two different sets of rules:

- Attribute name mapping where the name of a SAML Attribute can be translated to a local name (for example, firstname could be translated to givenname).
- Attribute value mapping where the value of a SAML Attribute can be translated to a local value (for example, President to CEO).

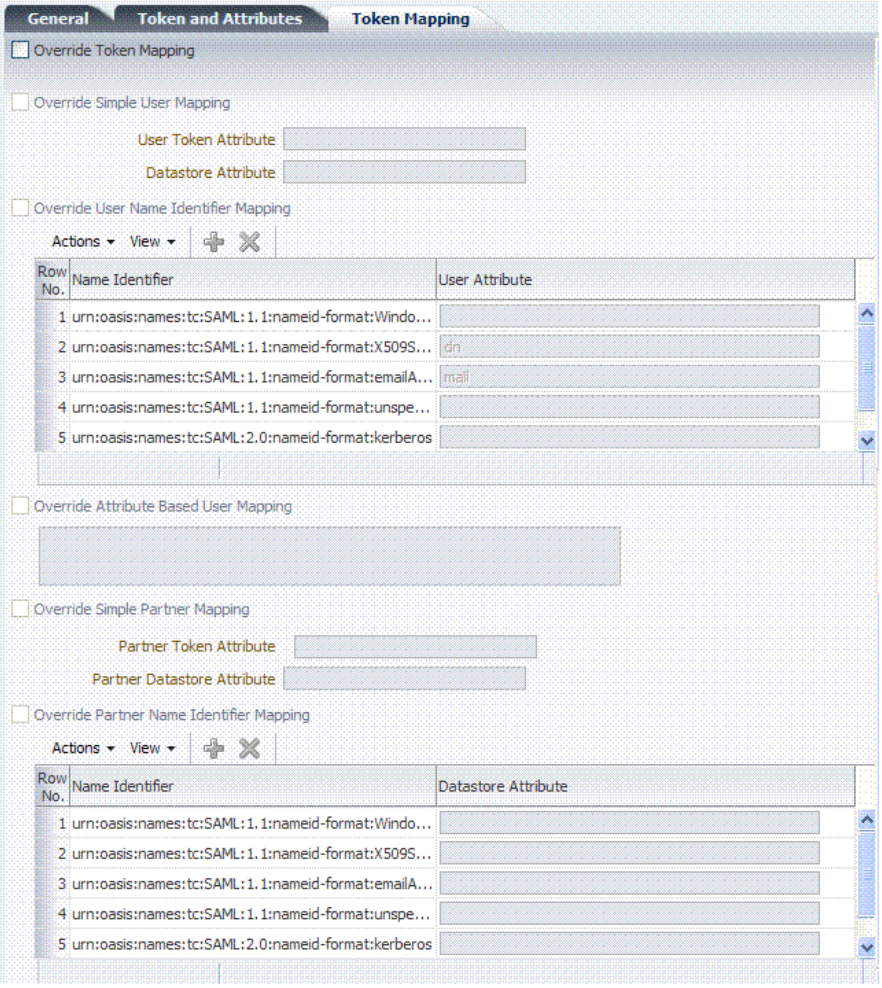
Table 20–5 Token and Attributes Elements: Issuing Authority

Element	Description
Attribute Name Mapping	Define an optional mapping between the name of a SAML Attribute and the local name of an attribute. The mapping is optional. If an attribute does not have a mapping defined, then its SAML attribute name will be used. ▪ Incoming Attribute: Contains the external name of the attribute as it will appear in the Assertion. ▪ Local Attribute: Contains the local name of the attribute. ▪ Format or Namespace: Contains an optional Format or Namespace. If missing, the namespace value for mapping purposes will be assumed to be `urn:oracle:security:fed:attrnamespace` for SAML 1.1 Assertions or the format value for mapping purposes will be assumed to be `urn:oasis:names:tc:SAML:2.0:attrname-format:basic` for SAML 2.0 Assertions
Value Mapping	Define an optional value mapping for a SAML attribute. This will indicate how to translate an attribute value to a local value, if needed. Note: This attribute value mapping applies to an Attribute Name mapping. In order to define an attribute mapping for an attribute, it is required to first define an attribute name mapping for that attribute. ▪ External Value: Contains the value of the SAML Attribute. ▪ Local Value: Contains the local value that will be set, if the SAML attribute value matches the External Attribute/Local Null fields. ▪ External Null: Represents a null SAML attribute value. ▪ Local Null: Indicates if the local value should be null, if the SAML attribute value matches the External Attribute/Local Null fields. ▪ Ignore Case: Indicates whether or not Oracle Security Token Service should ignore case when comparing the attribute value to the Local Attribute field.

Issuing Authority Profile: Token Mapping

Using the Token Mapping tab, shown in Figure 20–8, Administrators can override the Mapping Rules defined in a SAML Validation Template with the ones defined in an Issuing Authority Partner Profile. This way, Oracle Security Token Service can map SAML Assertions based on rules specific to a set of Assertion Issuers.Table 20–5 describes the Token Mapping elements for the Issuing Authority.

Figure 20–8 Issuing Authority Profile: Token Mapping Tab



□ Override Token Mapping

□ Override Simple User Mapping

User Token Attribute

Datastore Attribute

□ Override User Name Identifier Mapping

Row No.	Name Identifier	User Attribute
1	urn:oasis:names:tc:SAML:1.1:nameid-format:Windo…	
2	urn:oasis:names:tc:SAML:1.1:nameid-format:X509S…	dn
3	urn:oasis:names:tc:SAML:1.1:nameid-format:emailA…	mail
4	urn:oasis:names:tc:SAML:1.1:nameid-format:unspe…	
5	urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos	

□ Override Attribute Based User Mapping

□ Override Simple Partner Mapping

Partner Token Attribute

Partner Datastore Attribute

□ Override Partner Name Identifier Mapping

Row No.	Name Identifier	Datastore Attribute
1	urn:oasis:names:tc:SAML:1.1:nameid-format:Windo…	
2	urn:oasis:names:tc:SAML:1.1:nameid-format:X509S…	
3	urn:oasis:names:tc:SAML:1.1:nameid-format:emailA…	
4	urn:oasis:names:tc:SAML:1.1:nameid-format:unspe…	
5	urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos	

Table 20–6 Issuing Authority Token Mapping Elements

Element	Description
Override Token Mapping	Indicates whether or not the Mapping Rules defined in this section should override the ones listed in the SAML Validation Template used to process the assertion. This allows OSTS to use Mapping Rules that are specific to the Assertion Issuer. If true, all the Mapping Rules will be overridden by the settings listed in this section.
Override Simple User Mapping	Simple user mapping consists of mapping the incoming token to a user record by using a single token attribute and matching it against a single user record attribute. ■ User Token attribute references an attribute from the incoming token that will be matched against the Datastore attribute (defined below) of a user record. The values can be STS_SUBJECT_ID for the NameID Value, or the name of an Attribute contained in the Assertion's AttributeStatement ■ Datastore attribute references the user record attribute that will be matched against the User token attribute referenced above

Table 20–6 (Cont.) Issuing Authority Token Mapping Elements

Element	Description
Override User Name Identifier Mapping	When enabled, define a NameID user mapping operation, which consists of mapping the incoming SAML Assertion to a user record by mapping the NameID Value to a single user record attribute, based on the NameID format. When enabled, Oracle Security Token Service will evaluate the NameID format, and based on the Name Identifier mapping table which user record attribute should be matched against the Name ID value contained in the Assertion. The Name Identifier mapping table holds the user record attributes to be used for the mapping operation. It contains standard NameID formats, but it can be customized to define custom Name ID formats. ■ To add custom NameID format, click the add button on the Name Identifier mapping table, and enter the custom URI.. ■ To set an attribute for a specific NameID format to be used for mapping operation, set the user record attribute on the line for that format.
Override Attribute Based User Mapping	An Attribute Based User Mapping operation consists of mapping the incoming token to a user record by using an LDAP query and token attributes. The format of the LDAP query defines the mapping rule and specifies the token attributes to be used by their names, surrounded by % character. For example, an LDAP query that will map a token based on two token attributes (firstname and lastname) would be: `(&(sn=%lastname)(givenname=%firstname%))` STS_SUBJECT_ID contains the NameID Value. STS_NAMEID_FORMAT contains the NameID Format STS_NAMEID_QUALIFIER contains the NameID Qualifier STS_SAML_ASSERTION_ISSUER contains the Issuer of the Assertion Attributes present in the Assertion's AttributeStatement
Override Simple Partner Mapping	A simple partner mapping operation consists of mapping the incoming token to a partner requester by using a single token attribute and matching it against a partner identification attributes. ■ Partner Token attribute references an attribute from the incoming token that will be matched against the Partner Datastore attribute (defined below) of a Requester Partner. The values can be STS_SUBJECT_ID for the NameID Value, or the name of an Attribute contained in the Assertion's AttributeStatement. ■ Partner Datastore attribute references the partner identification attribute that will be matched against the Partner token attribute referenced above.
Override Partner Name Identifier Mapping	When enabled, define the following: A NameID user mapping operation consists of mapping the incoming SAML Assertion to a user record by mapping the NameID Value to a single requester partner identification attribute, based on the NameID format. When enabled, OSTS will evaluate the NameID format, and based on the Name Identifier mapping table which partner identification attribute should be matched against the Name ID value contained in the Assertion. The Name Identifier mapping table holds the requester partner identification attributes to be used for the mapping operation. It contains standard NameID formats, but it can be customized to define custom Name ID formats. ■ To add custom NameID format, click the add button on the Name Identifier mapping table, and enter the custom URI. ■ To set an attribute for a specific NameID format to be used for mapping operation, set the requester partner identification attribute on the line for that format.

20.4.2 Managing a Token Service Partner Profile

Users with valid Administrator credentials can use this procedure to create, locate, view, edit, or remove a token service partner profile.

Prerequisites

The prerequisites for Requester Partner Profiles are:

- A Relying Party Partner Profile must exist, in order to be able to set the default Relying Partner Profile.
- WS-Trust Validation Templates must exist in order to set the templates that will be used to validate tokens located in the OnBehalfOf element.

The prerequisites for Relying Partner Profiles are:

- Issuance Template must exist in order to configure which templates to use for token issuance operations.

There are no prerequisites for Issuing Authority Partner Profiles.

To create, find, edit, or remove a partner profile

1. From the Oracle Access Manager Console, open the:

   System Configuration tab
   Security Token Services section
   Partner Profiles node

2. Under the Partner Profiles node, double-click the desired profile type and proceed with following steps as needed.
   - Requester Profiles
   - Relying Party Profiles
   - Issuing Authority Profiles
3. **New Profile**:
   a. Click the New *ProfileType* button to display a fresh page for your definition.
   b. Enter general information for the chosen profile type (Table 20–2) and click the Next Button.
   c. **Token and Attributes**: Use the appropriate table to provide details for the chosen profile type:

      Requester Profile Table 20–3

      Relying Party Profile Table 20–4

      Issuing Authority Profile Table 20–5

   d. Click Save to submit (or click Cancel to dismiss the page) and then dismiss the confirmation window.
4. **Refine a Profile Search**: "Refining a Profile Search"
   a. Perform Steps 1 and 2.
   b. Define your query and click the Search button.
   c. In the Search Results table, click the name of partner to view, edit, or remove.
5. **Edit a Profile**:

a. In the Search Results table, click the name of profile to edit and click the Edit button (or choose Edit from the Actions menu).

b. Make desired changes to partner information.

   Requester Profile Table 20–3

   Relying Party Profile Table 20–4

   Issuing Authority Profile Table 20–5

c. Click Apply to submit the changes (or Revert to cancel changes) and then dismiss the confirmation window.

6. **Remove a Profile**: To remove a profile, it is required not to be referenced anywhere else.

   To remove a Requester Partner Profile, it is required that:

   - No Requester Partner references the profile.
   - No WS-Security Validation Template references the profile

   To remove a Relying Party Partner Profile, it is required that:

   - No Relying Party Partner references the profile.
   - No Requester Partner Profile references the profile.

   To remove an Issuing Authority Partner Profile, it is required that:

   - No Issuing Authority Partner references the profile

   If these prerequisites are met, proceed as follows:

   a. In the Search Results table, highlight the row containing the profile to remove.

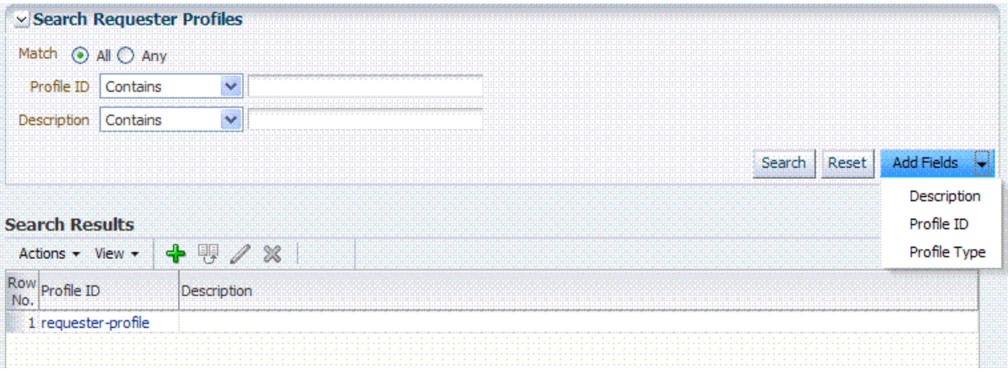
   b. Click the Delete (X) button (or choose Delete Selected from the Actions menu), then dismiss the confirmation window.

20.4.3 Refining a Profile Search

As with Partner definitions, when you open the Partner Profiles node, all Partner Profiles nodes become available. When you choose a specific type of Partner Profile node, relevant Search controls, and the Search Results table, become available.

Figure 20–3 illustrates a typical Search Profiles page. This one is for a Requester Profile. However, all controls are the same; only the results differ for different profile types.

Figure 20–9 Search Profiles Page: Requester



Search Requester Profiles

Match ● All ○ Any

Profile ID: Contains

Description: Contains

Row No.	Profile ID	Description
1	requester-profile	

From the Search page you can simply select a name in the Search Results table to view or edit the Profile, or use the controls to refine your search to locate a specific Profile or a Profile with specific characteristics.

21

Troubleshooting Oracle Security Token Services

This chapter provides troubleshooting tips for Oracle Security Token Service:

- Authorization Issues
- Endpoint Issues
- Mapping Operation Issues

21.1 Authorization Issues

Problem: Authorization Failure during Token Issuance operation

During a WS-Trust request issuance operation, the Oracle Security Token Service returns an error.

Error Message

The following are sample error messages that can be seen in the logs:

```
<Error> <oracle.security.fed.controller.ApplicationController> <STS-12064>
<Exception: {0}
oracle.security.fed.event.EventException:
oracle.security.fed.event.EventException: Authorization Failure for Relying
Party=%RELYING_PARTY_ID%, Requester=%REQUESTER_ID% and User=%USER_ID%
```

When:

- `%RELYING_PARTY_ID%` indicates the Relying Party Partner ID.
  - If the WS-Trust request did not contain an AppliesTo element, then the `%RELYING_PARTY_ID%` is set to MissingRP
  - if the WS-Trust request contained an AppliesTo element but it could not be mapped to a Relying Party Partner, then the `%RELYING_PARTY_ID%` is set to UnknownRP
  - if the WS-Trust request contained an AppliesTo element and it was mapped to a Relying Party Partner, then the `%RELYING_PARTY_ID%` is set to Relying Party Partner ID.
- `%REQUESTER_ID%` is set to the Requester Partner ID, if the incoming request was mapped to a Requester Partner. If `%REQUESTER_ID%` is not null, it will be used when evaluating the Token Issuance Policy, against any present Identity Constraint.

- %USER_ID% is set to the User ID, if the incoming request was mapped to a user record. If %USER_ID% is not null and if %REQUESTER_ID% is null, it will be used when evaluating the Token Issuance Policy, against any present Identity Constraint.

Issue

The Token Issuance Policy evaluation failed due to one of the following reasons:

- No TokenServiceRP resource referencing the %RELYING_PARTY_ID% is defined and assigned to a Token Issuance Policy. In this case, create TokenServiceRP resource referencing the %RELYING_PARTY_ID% and assign it to a Token Issuance Policy.
- A TokenServiceRP resource referencing the %RELYING_PARTY_ID% exists and is assigned to a Token Issuance Policy, but the policy contains constraints that are not met. In this case, review the policy rules: if the policies are correct, then the client is not allowed to request a token; otherwise, update the policies/constraints to include the client's identity.

21.2 Endpoint Issues

Problem: Endpoint not found

When accessing an Oracle Security Token Service endpoint that has been added via the OAM/OSTS console, the server returns an error indicating that the page does not exist when retrieving the WSDL policy or that the endpoint does not exist.

Error Message

The following are possible error messages:

- When retrieving the WSDL policy, a 404 HTTP error code is returned.
- When sending a WS-Trust request, an error is reported:

```
<Error> <oracle.webservices.service> <OWS-04115> <An error occurred for port:
PortableProvider: oracle.j2ee.ws.server.EndpointNotFoundException: /PATH.>
```

Solution

The Oracle Security Token Service application is deployed but not enabled. To enable Oracle Security Token Service, perform the following operations:

1. Go to the OAM Admin console.
2. Navigate to **System Configuration**, select **Common Configuration**, then select **Available Services**.
3. Enable Oracle Security Token Service.

Oracle Security Token Service will detect the change and will publish the endpoints. No restart is required.

21.3 Mapping Operation Issues

Problem: Failure to map the AppliesTo element to a Relying Party Partner

When Oracle Security Token Service processes a WS-Trust request with an AppliesTo element referencing the Web Service Provider, the server will attempt to map the location contained in the AppliesTo element to an Oracle Security Token Service

Relying Party Partner using the Resource URL defined in the Partner entry. If such a mapping fails, the server will log an Info message in the logs indicating that the operation failed and indicating what was the AppliesTo address used.

Error Message

The following is a sample of an error message:

```
[2011-04-22T15:08:12.632-07:00] [oam_server1] [NOTIFICATION] [STS-15542]
[oracle.security.fed.eventhandler.sts.creation.v13.CreateV13TokenEventHandler]
[tid: [ACTIVE].ExecuteThread: '0' for
queue: 'weblogic.kernel.Default (self-tuning)'] [userId: <anonymous>] [ecid:
f00aacae2d3f3ded:125005ed:12f7f412274:-8000-0000000000000016,0] [WEBSERVICE_
PORT.name: wssuser-port] [APP:
oam_server] [J2EE_MODULE.name: sts] [WEBSERVICE.name: wssuser-serviceSoap12]
[J2EE_APP.name: oam_server] The mapping
of the AppliesTo element from the WS-Trust Request to a Relying Party Partner
failed: could not map
http://relying.party.test.com/testing/service
```

Solution

If the AppliesTo location should have been mapped to a Relying Party Partner, then the Partner settings should be verified to ensure that the Resource URLs are correctly defined to:

- be the exact match of the AppliesTo address
- be a parent of the AppliesTo address.

  For example, if the AppliesTo address is http://*relying.party.test.com/testing/service*, a parent could be http://*relying.party.test.com/testing/* or http://*relying.party.test.com/*. In both cases, the AppliesTo location would be mapped to a Relying Party Partner with any of those Resource URLs defined.

> **Note:** this message is recorded at Notification level, thus in order for OSTS to record it, the appropriate logging level must be set to include the Notification:1 level.

In certain cases, failure to correctly map the AppliesTo address to a Relying Party Partner will result in errors due to:

- Authorization evaluation failures
- Oracle Security Token Service not being able to retrieve certificate belonging to the Relying Party Partner.

Part VI

Common Logging, Auditing, Performance Monitoring

Part VI provides information to help you perform logging, auditing, and performance monitoring for Oracle Access Manager and Oracle Security Token Service.

Part VI contains the following chapters:

- Chapter 22, "Logging Component Event Messages"
- Chapter 23, "Logging Webgate Event Messages"
- Chapter 24, "Auditing Administrative and Run-time Events"
- Chapter 25, "Monitoring Performance by Using Oracle Access Manager Console"
- Chapter 26, "Monitoring Performance and Logs with Fusion Middleware Control"

22

Logging Component Event Messages

Logging is the mechanism by which components write messages to a file. Administrators can use the logging mechanism to capture critical component events. Oracle Access Manager and Oracle Security Token Service 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.

Oracle Fusion Middleware Administrator's Guide for Oracle Access Manager with Oracle Security Token Service

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.

> **Note:** Unless explicitly stated, information in this chapter is the same whether you are using Oracle Access Manager alone or with Oracle Security Token Service.

This chapter includes the following topics:

- Prerequisites
- Introduction to Logging Component Event Messages
- Configuring Logging for Oracle Access Manager
- Configuring Logging for Oracle Security Token Service
- Validating Run-time Event Logging Configuration

22.1 Prerequisites

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

Oracle recommends that you review the Chapter 6, "Managing Common OAM Server Registration".

22.2 Introduction to Logging Component Event Messages

The logging infrastructure records messages that can be used for problem diagnosis. Oracle Security Token Service is a J2EE Web application, part of the Oracle Access Manager J2EE Application. Both use OJDL for logging purposes. Oracle Security Token Service captures the interactions between itself and Partners with timestamps.

The administrator controls the amount of information that is logged in a message by specifying log levels for each 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.

Oracle Access Manager and Oracle Security Token Service use the WebLogic container's logging defaults:

- **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
- **Oracle Access Manager Log File**: *DOMAIN_HOME*/servers/*SERVER-NAME*/logs/*SERVER-NAME*-diagnostics.log

> **Note:** By default, Oracle Security Token Service messages are logged into the Managed Servers log files. However, for convenience, you can edit logging.xml to direct Oracle Security Token Service information to a separate log file, as described in "Configuring Logging for Oracle Security Token Service" on page 22-8.

The following events are logged automatically:

- OAM Server events (managed run-time servers)
- Administrative events (generated for configuration changes made using the console)

By default, the log level for all Oracle Access Manager and Oracle Security Token Service 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 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).

For more information, see:

- About Component Loggers
- Sample Logger and Log Handler Definition
- About Logging Levels

> **See Also:**
>
> - Chapter 26 for details about how you can configure and view logs using Fusion Middleware Control
> - Logging information in the Oracle Fusion Middleware Application Security Guide

22.2.1 About Component Loggers

This section introduces component loggers for both Oracle Security Token Service and Oracle Access Manager. There are differences.

Oracle Security Token Service has only a single logger: oracle.security.fed. For more information, see "Configuring Logging for Oracle Security Token Service" on page 22-8.

Each Oracle Access Manager component is associated with its own logger name, as listed in the following tables:

- Table 22–1, " Oracle Access Manager Server-Side Components"
- Table 22–2, " Oracle Access Manager Shared-Service Engine Components"
- Table 22–3, " Oracle Access Manager Foundation APIs Components"

Table 22–1 Oracle Access Manager 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
Oracle Access Manager Console	oracle.oam.admin.console
Admin-Service Config	oracle.oam.admin.service.config
Diagnostics and Monitoring	oracle.oam.diag

Table 22–2 Oracle Access Manager 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 22–3 Oracle Access Manager Foundation APIs Components

Component Name	OAM Logger Name
Session Access	oracle.oam.session.access
Session Access Implementation	oracle.oam.session.accessimpl
Policy Access	oracle.oam.policy.access

22.2.2 Sample Logger and Log Handler Definition

This topic provides a sample for Oracle Access Manager only. Oracle Security Token Service has only one logger and log handler, as described in "Configuring Logging for Oracle Security Token Service" on page 22-8.

Example 22–1 illustrates the configuration of an Oracle Access Manager logger and a log handler in the file `logging.xml`.

Example 22–1 Configuring Oracle Access Manager 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.

22.2.3 About Logging Levels

This topic applies equally to Oracle Access Manager and Oracle Security Token Service.

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 "About Component Loggers"), and *notifLevel* is either an ODL message level or a Java message level.

Table 22–4 shows the correspondence between ODL message levels and Java message levels, in increasing order:

Table 22–4 Mapping of ODL to Java Levels

ODL Message Level	Java Message Level
INCIDENT_ERROR:1	SEVERE.intValue()+100
ERROR:1	SEVERE (logs exceptions)
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)

Table 22–4 (Cont.) Mapping of ODL to Java Levels

ODL Message Level	Java Message Level
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.

> **Note:** if you define a filter to log messages at the finest level for the oracle.security.fed package and sub-package (classes for Oracle Security Token Service), after restarting the server you would see logs for the OAM Server with Oracle Security Token Service. For more information, see "Configuring Logging for Oracle Security Token Service" on page 22-8.

22.3 Configuring Logging for Oracle Access Manager

This section describes tasks for only Oracle Access Manager.

> **See Also:** "Configuring Logging for Oracle Security Token Service"

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 Logger Level for Oracle Access Manager
- Adding an Oracle Access Manager-Specific Logger and Log Handler

22.3.1 Modifying the Logger Level for Oracle Access Manager

Administrators can use custom WLST commands for Oracle Access Manager to change 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 Administrators"*

To modify the OAM logger level

1. Confirm that the OAM Server is running.
2. Acquire the custom WLST script for Oracle Access Manager. 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 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 Oracle Access Manager 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                      | TRACE:32
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 22-10.

22.3.2 Adding an Oracle Access Manager-Specific Logger and Log Handler

Administrators can use the following procedure to specify a log file path and necessary attributes.

In the following procedure, you will identify the target OAM Server, rotation and retention periods, a path to the log file, the handler, and logger. Your deployment and choices will 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 specify 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 Oracle Access Manager 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")
```

5. Add a custom log handler and associate it with the Oracle Access Manager 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 logs in the DOMAIN_HOME/oamlogs directory:

   ```
   DOMAIN_HOME/oamlogs/
   ```

22.4 Configuring Logging for Oracle Security Token Service

By default Oracle Security Token Service messages are logged into the OAM Server's log files. You can view and configure logs in Fusion Middleware Control. However, you can also edit logging.xml and direct Oracle Security Token Service information to a separate log file, as described here.

The files that are involved include:

- **Logging Configuration File**: Provides logger names and other configuration information for logging. This file is stored in: *DOMAIN_HOME*/config/fmwconfig/servers/*SERVER-NAME*/logging.xml
- **Oracle Security Token Service Log File**: *DOMAIN_HOME*/ostslogs/*SERVER-NAME*-diagnostics.log, for example

Oracle Security Token Service does not categorize log handlers as Oracle Access Manager does. Instead, there is only one logger that affects the log levels for Oracle Security Token Service. Table 22–5 provides details for this logger that are required in the WLST command.

Table 22–5 Oracle Security Token Service Logger

Component Name	Logger Name	Log Handler Name	Log Class
Oracle Security Token Service	oracle.security.fed	sts-handler	class='oracle.core.ojdl.logging.ODLHandlerFactory'

For details, see:

- Configuring Logging for Oracle Security Token Service
- Defining the Log Level and Log Details for Oracle Security Token Service

See Also:

- Chapter 26 for details about how you can configure and view logs using Fusion Middleware Control
- Logging information in the Oracle Fusion Middleware Application Security Guide

22.4.1 Configuring Logging for Oracle Security Token Service

Administrators can use the following procedure to separate Oracle Security Token Service log messages from OAM Server message logs.

To configure logging for Oracle Security Token Service

1. Locate and open logging.xml: *DOMAIN_HOME*/config/fmwconfig/servers/*SERVER-NAME*/logging.xml.
2. Add the following to create the independent message log for Oracle Security Token Service:

```
<log_handler name='sts-handler' class='oracle.core.ojdl.logging.ODLHand
lerFactory'>
      <property name='path' value='sts/log'/>
      <property name='maxFileSize' value='10485760'/>
      <property name='maxLogSize' value='104857600'/>
    </log_handler>

<logger name='oracle.security.fed' level='TRACE:32'>
      <handler name='sts-handler'/>
    </logger>
```

3. Save the file.
4. Proceed with "Defining the Log Level and Log Details for Oracle Security Token Service".

22.4.2 Defining the Log Level and Log Details for Oracle Security Token Service

Administrators can use custom WLST commands for Oracle Access Manager to change logger settings for Oracle Security Token Service as described here. This specifies an independent output file for only Oracle Security Token Service log messages.

This sample procedure for Oracle Security Token Service logging is very similar to the one for Oracle Access Manager. However, there are a few differences. Your deployment choices will be different.

> **Note:** Use the WLST command `help("fmw diagnostics")`.

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:** *Appendix F, "Introduction to Custom WLST Commands for Administrators"*

To modify the logger level and log file for Oracle Security Token Service

1. Confirm that the OAM Server is running.
2. Acquire the custom WLST script for Oracle Access Manager:

   ```
   <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. Modify the log level of `oracle.security.fed` based on your requirements. For example, this sequence changes the log level to WARNING with no persistence:

   ```
   wls:/base_domain/serverConfig> domainRuntime()
   wls:/base_domain/domainRuntime> setLogLevel(logger="oracle.security.fed",
   level="WARNING", persist="0", target="oam_server1")
   ```

5. Specify the target OAM Server, as well as rotation and retention periods, path to the log file, the handler, and logger. For example:

```
 wls:/base_domain/domainRuntime> configureLogHandler(name="osts-log-handler",
target="oam_server1", rotationFrequency="daily", retentionPeriod="week",
path="${domain.home}/ostslogs", maxFileSize ="10485760", maxLogSize
="104857600", addHandler="true",handlerType="oracle.core.ojdl.logging.ODL
HandlerFactory", addToLogger="oracle.security.fed")
```

6. Verify the generated log file to confirm the controller is logged at the WARNING level:

```
 DOMAIN_HOME/ostslogs/SERVER-NAME-diagnostics.log
```

7. Proceed to "Validating Run-time Event Logging Configuration" on page 22-10.

22.5 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 13, "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 logs appear in:

```
DOMAIN_HOME/oamlogs/
DOMAIN_HOME/ostslogs/SERVER-NAME-diagnostics.log
```

4. Open the log file and look for the last entries to confirm authentication failure and success, respectively.

23

Logging Webgate Event Messages

Each Webgate instance (both 10g and 11g Webgates) can write information about its processes and states to a log file. The logs can be configured to provide information at various levels of granularity. For example, you can record errors, errors plus state information, or errors, states, and other information to the level of a debug trace. You can also eliminate sensitive information from the logs.

> **Note:** Unless explicitly stated, all information in this section applies equally to 10g and 11g Webgates. For instance, the location of the log configuration, oblog_config_wg.xml, has changed for 11g while the content of the file and most other specifics have not changed.

This chapter provides the following sections:

- About Logging, Log Levels, and Log Output
- About Log Configuration File Paths and Contents
- About Directing Log Output to a File or the System File
- Structure and Parameters of the Log Configuration File
- About Activating and Suppressing Logging Levels
- Mandatory Log-Handler Configuration Parameters
- Configuring Different Threshold Levels for Different Types of Data
- Filtering Sensitive Attributes

About Logging, Log Levels, and Log Output

The logging feature enables you to analyze system performance and health, and to troubleshoot issues.

You can configure logging for individual Webgate instances of the following components:

- 10g Webgates
- 11g Webgates
- Custom Access Clients (Access Manager SDK)

You can configure different logging levels for different functional areas of a component instance. For example, you can capture debug data for LDAP activity while recording only error-level data for all other component activity. You can also record the time taken for each request that a component processes, and you can send different levels of log data to different destinations. For example, you can send error information to a file and all other log data to the system log.

Securing Sensitive Information: Oracle Access Manager handles sensitive information about users. On some sites, this includes user password, date of birth, a social security number, security questions and answers for lost password requests. Sensitive data on your site might include a security number or other information you want to secure. At certain logging levels, sensitive information might be captured. Today, you can filter sensitive information out of log files, as described in "Filtering Sensitive Attributes" on page 23-26.

Configuring Logging: You configure logging by editing a configuration file that is stored with the Webgate. See "About Log Configuration File Paths and Contents" on page 23-4.

Logging Levels: You can request logging at various levels. The highest level is Fatal and the lowest level is Trace. See "About Log Levels" on page 23-2 for details.

Logging Destinations: In the log configuration file, a parameter known as a log writer determines the destination for log output. See "About Directing Log Output to a File or the System File" on page 23-9 for details. You create a complete definition for your log output by identifying a log writer and a log level. This complete definition is known as a log-handler. See "The Second Compound List and Log Handlers" on page 23-13 for details.

The rest of this section discusses the following topics:

- About Log Levels
- About Log Output

About Log Levels

A logging level determines the amount of data that is written to the log data file. Each logging level is cumulative, that is, each level contains all the data generated by the higher levels. For example, Error logs contain all the data generated by the Fatal logs, plus the events that are specific to the Error category.

Table 23–5 describes the levels. The default log level is Warning: LOGLEVEL_WARNING.

Table 23–1 Logging Levels

Level	Number of Events Reported	Description
LOGLEVEL_FATAL	> 60	Records critical errors. Generally, these events can cause the component to exit. In the event of a system failure, Fatal-level messages are always flushed to the log file.
LOGLEVEL_ERROR	> 960	Records events that may require corrective action, for example, a component is unavailable. Error logs can also be generated for transient or self-correcting problems, for example, failure to connect to another component.
LOGLEVEL_WARNING	> 1200	Records issues that may lead to an error or require corrective action in the future.
LOGLEVEL_INFO	> 400	Records completed actions or the current state of a component, for example, the component is initializing.

Table 23–1 (Cont.) Logging Levels

Level	Number of Events Reported	Description
LOGLEVEL_DEBUG1	> 400	Records debugging information. Typically, the information at this level is only meaningful to a developer.
LOGLEVEL_DEBUG2	> 100	Records advanced debugging information. This level augments the Debug1 log level. Typically, the information at this level is only meaningful to a developer.
LOGLEVEL_DEBUG3	> 900	Records a large amount of debugging information or data pertaining to an expensive section of the code. This level is useful for debugging a tight loop or a performance-sensitive function. Typically, the information at this log level is only meaningful to a developer. These logs can contain sensitive information.
LOGLEVEL_TRACE	> 900 Oracle Access Manager API > 150 third-party API	This log level is used to trace code path execution or to capture performance metrics. This information is captured at the entry and exit points for each component function. Typically, the information at this log level is only meaningful to a developer. These logs can contain sensitive information.
LOGLEVEL_ALL	> 5000	This level includes all the events and states from all other levels.

Compound Lists: You can collect log data from non-adjacent levels and send different levels of log data to different destinations. For example, you can send the Fatal logs to the system log, and write Error logs to a file. See "The Second Compound List and Log Handlers" on page 23-13 for details.

Threshold: You configure a global cutoff, or threshold, for logging on the LOG_THRESHOLD_LEVEL parameter in the log configuration file. By default, if a configured level for a log-handler exceeds the cutoff, the log data is not collected. Note that logs can fail to be written despite the configured level because the LOG_THRESHOLD_LEVEL parameter takes precedence over the level configured in the log-handler. Only the MODULE_CONFIG section of the log configuration file overrides the global threshold. See "The Simple List and Logging Threshold" on page 23-11 for details.

Overrides: You specify function- or module-specific overrides for the global logging threshold on the MODULE_CONFIG parameter. See "Configuring Different Threshold Levels for Different Types of Data" on page 23-21 for details.

Note: The Trace and Debug3 level logs can contain sensitive information. For more information about sensitive information, see "Filtering Sensitive Attributes" on page 23-26.

About Log Output

Each line of the log output file follows a particular structure. A line starts with a date and time stamp, followed by the thread that is processing the request, the name of the function or module being logged, and the log level.

The following is a snapshot of the left-most columns of the log output file:

```
2007/06/01@00:50:56.859000    5932  2672  DB_RUNTIME     DEBUG3
2007/06/01@00:50:56.859000    5932  2672  DB_RUNTIME     TRACE
2007/06/01@00:50:56.859000    5932  2672  LDAP           DEBUG1
2007/06/01@00:50:56.859000    5932  2672  LDAP           TRACE
2007/06/01@00:50:56.859000    5932  2672  LDAP           TRACE
```

The two columns to the right of the log level are internal code references, and can be ignored. The following is an example of these columns:

```
0x00000205   ldap_connection_mngr.cpp:212
```

To the right of the internal code reference columns, you see the log message that is associated with this log level, for example, "Function called" or "Function returned," followed by the name of the function, as illustrated in the following example:

```
"Function called"   _CallName^ldap_init
```

The log message and function name can be followed by additional information, for example, the duration of the process, the address space where the function is running, or state information, as illustrated in the following examples:

```
"Connection health check result"   Server^dlsun4072   Port^389   Server Priority^1
Connection available^true

"Function entered"    _TraceName^ConnectionWatcherThread::CheckPrimaries

"Function exited"     _TraceName^ConnectionWatcherThread::CheckPrimaries
TraceDuration^0.000028

"Connection Pool Status in ValidateConnections()    "NumLivePrimaryConnections^1
Maximum Connections^1     UpConnections^1     Failover Threshold^1     Max Session
Time^0    SleepFor^60
```

To secure sensitive information and ensure that it is not included in the output of the logging operation, see "Filtering Sensitive Attributes" on page 23-26.

See Also: "Log Configuration File Contents" on page 23-5

About Log Configuration File Paths and Contents

The log configuration file, oblog_config_wg.xml, is used to specify configuration details for Webgate logging (oblogs).

You configure parameters that control Webgate log output in XML-based log files that you edit with a plain text editor. Changes that you make to these files are effective immediately.

The rest of this section discusses the following topics:

- Log Configuration File Paths and Names
- Log Configuration File Contents

Log Configuration File Paths and Names

By default, Webgate logging is enabled and oblogs are generated in the Oracle HTTP Server (OHS) instance diagnostics directory: instance1/diagnostics/logs/OHS/ohs1/.

Each Webgate instance includes a log configuration file (oblog_config_wg.xml) where you can define what type of data is recorded in the log output. A log configuration file is distinct from the log output file. For details on log output files, see "About Log Output" on page 23-3.

The oblog_config_wg.xml file is updated when you edit to configure Webgate logging. For example, by setting a new log threshold level, changing a log file name, or filtering logs related to some modules and so on.

Log configuration, oblog_config_wg.xml, files reside in the following locations depending upon your Webgate version:

10g Webgates: *Webgate_install_dir*\oblix\config

11g Webgates: Webgate/Oracle_Home under webgate/ohs/config and Instance_Home/webgate/config (the later to be used when configuring logging).

The same oblog_config_wg.xml file is copied to the Webgate instance directory when the Webgate instance is created. In Instance_Home, this file is located at webgate/config.

Note: Do not change the path to this file. If you install more than one instance, a log configuration file is installed for each instance. When configuring logging, use oblog_config_wg.xml under Instance_Home should be updated.

After installation, oblog_config_wg.xml and oblog_config_wg_original.xml both contain comments to help guide your editing.

Table 23–2 lists the names of the log configuration files. Do not change the names.

Table 23–2 Log Configuration File Names for Components

Component	Log Configuration FIle Name
Webgate	oblog_config_wg.xml
Access Manager SDK (custom Access Client)	oblog_config.xml

Important: Do not change the default path or name for any logging configuration file.

The oblog_config_wg.xml file can be edited using any text editor as long as you ensure that after the update the file is still valid XML. After updates to the file, changes will take affect in about 60 seconds.

Log Configuration File Contents

The log configuration file controls items such as the following:

- What is logged for that component
- Where the data is sent
- In certain cases, the size of the write buffer used for the log
- Log file rotation intervals

The configuration file contains XML statements that you can edit in a text editor.

When Changes to the File Take Effect

A watcher thread picks up changes to the log configuration file every 60 seconds and ensures that changes take effect. It is unnecessary to restart the server

About Comments in the Log File

Each default log configuration file contains comments that are intended to assist with editing the file.

> **See Also:** The log configuration file on your system.

The commented default configuration file is shown here:

Comments can span one or multiple lines. Comments look similar to the following:

```
<!--NetPoint Logging Configuration File                        -->
<!--                                                           -->
<!--Changes to this file will be automatically taken into  effect  -->
<!--in one minute. This does not require any server restart.      -->
```

Example 23–1 shows a typical log configuration file with comments. Example 23–8 shows an example of a log file without comments.

Example 23–1 The Default Log Configuration File with Comments

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--============================================================   -->
<!--============================================================   -->
<!--NetPoint Logging Configuration File                            -->
<!--                                                               -->
<!--Changes to this file will be automatically taken into  effect  -->
<!--in one minute. This does not require any server restart.      -->
<!--                                                               -->
<!--============================================================   -->
<!--============================================================   -->
<!--Set the Log Threshold                                          -->
<!---->
<!--The log Threshold determines the amount of information to log. -->
<!--Selecting a lower level of logging includes the information    -->
<!--logged at the higher levels. For example, LOGLEVEL_ERROR       -->
<!--includes the information collected at LOGLEVEL_FATAL.          -->
<!---->
<!--Choices are:                                                   -->
<!--LOGLEVEL_FATAL - serious error, possibly a program halt.       -->
<!--LOGLEVEL_ERROR - a transient or self-correcting problem.       -->
<!--LOGLEVEL_WARNING - a problem that does not cause an error.     -->
<!--LOGLEVEL_INFO - reports the current state of the component.    -->
<!--LOGLEVEL_DEBUG1 - basic debugging information.                 -->
<!--LOGLEVEL_DEBUG2 - advanced debugging information.              -->
<!--LOGLEVEL_DEBUG3 - logs performance-sensitive code.             -->
<!--LOGLEVEL_TRACE - used when you need to trace the code path     -->
<!--execution or capture metrics. Includes all previous levels.    -->
<!--                                                               -->
<!--If you do not specify a threshold, the default is WARNING.     -->
<!--                                                               -->
<!--In addition to specifying a threshold, you need to specify     -->
<!--if changes that you make to the logging configuration in       -->
<!--the NetPoint GUI overwrite the settings in this file. The      -->
<!--AutoSync parameter accomplishes this. This parameter takes a   -->
<!--value of True or False. If set to True, changes made in the    -->
```

```
<!--GUI overwrite changes in this config file. If False, changes    -->
<!--made in the GUI are only in effect until the server is          -->
<!--stopped or restarted, after which the settings in this file     -->
<!--overwrite the GUI settings. The default is True.                -->
<!--                                                                -->
<!--                                                                -->
<CompoundList xmlns="http://www.oblix.com" ListName="logframework.xml.staging">
  <SimpleList>
   <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
   <NameValPair ParamName="AUTOSYNC" Value="True" />
   <!-- SECURE_LOGGING flag can be used to turn on/off Secure Logging -->
   <!-- feature. By defalut this feature is tunred on.                 -->
   <NameValPair ParamName="SECURE_LOGGING" Value="On" />
   <!-- In addition to specifying a log threshold, you need to       -->
   <!-- configure log level for which Secure Logging should be       -->
   <!-- applicable.Choices for this can be used same as that of      -->
   <!-- LOG_THRESHOLD_LEVEL. Secure log threshold can be set using   -->
   <!-- LOG_SECURITY_THRESHOLD_LEVEL flag. Default value for Secure  -->
   <!-- log threshold is TRACE.                                     -->
   <NameValPair ParamName="LOG_SECURITY_THRESHOLD_LEVEL"
    Value="LOGLEVEL_TRACE" />
   <!-- LOG_SECURITY_ESCAPE_CHARS is used to configure escape sequence -->
   <!-- characters. This can be used to avoid additional information   -->
   <!-- getting overwritten due to Secure Logging mechanism. Currently -->
   <!-- following characters have been identified as escape sequence.  -->
   <!-- Configuring inappropriate characters may lead to sensitive     -->
   <!-- information being unmasked.                                    -->
   <NameValPair ParamName="LOG_SECURITY_ESCAPE_CHARS" Value="),]" />
   <!-- LOG_SECURITY_MASK_LENGTH is used to specify default masking    -->
   <!-- length if none is specified in FILTER_LIST.                    -->
   <!-- Default value for LOG_SECURITY_MASK_LENGTH is 300.             -->
   <NameValPair ParamName="LOG_SECURITY_MASK_LENGTH" Value="300" /   >
  </SimpleList>
<!--                                                                -->
<!--                                                                -->
<!--=============================================================   -->
<!--=============================================================   -->
<!--Configure the Log Level                                         -->
<!--                                                                -->
<!--                                                                -->
<!--To configure a log level, you specify a name for the            -->
<!--configuration (for instance, MyErrorLog1) and                   -->
<!--the log level that you are configuring. You can create          -->
<!--more than one configuration per log level if you want           -->
<!--to output to more than one destination. You can output to       -->
<!--the system log or to a file, as specified on                    -->
<!--the LOG_WRITER parameter. The value for the LOG_WRITER          -->
<!--parameter may only be SysLogWriter, FileLogWriter or            -->
<!--MPFileLogWriter. The MPFileLogWriter is a multi-process safe    -->
<!--FileLogWriter. It should be used to log in webcomponents i.e    -->
<!--Webgate loaded on multiprocess     -->
<!--webservers like Apache and IPlanet(UNIX)                        -->
<!--                                                                -->
<!--If you do not specify an output destination, the default is     -->
<!--SysLogWriter.                                                   -->
<!--                                                                -->
<!--If outputting to a file, you also specify a file name and       -->
<!--other parameters. Default parameter values are:                 -->
<!--FILE_NAME: <installdir>/oblix/log/oblog.log                     -->
<!--BUFFER_SIZE: 32767 (number of bytes)                            -->
```

```
<!--MAX_ROTATION_SIZE: 5242880 (bytes, equivalent to 5MB)          -->
<!--MAX_ROTATION_TIME: 86400 (seconds, equivalent to one day)      -->
<!--                                                               -->
<!--Configuring the log level does not ensure that the data is     -->
<!--actually collected. Data collection for a log is               -->
<!--determined by the LOG_THRESHOLD_LEVEL parameter, above,        -->
<!--and the LOG_STATUS parameter in the log configuration.         -->
<!--                                                               -->
<!--If you do not provide a LOG_STATUS, the default for            -->
<!--LOGLEVEL_FATAL, LOGLEVEL_ERROR, and LOGLEVEL_WARNING,          -->
<!--is On.                                                         -->
<!---->
<!--This file contains several sample configurations that are      -->
<!--enclosed in comments. To use them, remove the comments.        -->
<!--                                                               -->
   <CompoundList xmlns="http://www.oblix.com" ListName="LOG_CONFIG">
    <!--Write all FATAL logs to the system logger. -->
    <ValNameList xmlns="http://www.oblix.com" ListName="LogFatal2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
    <!--Write all logs to the Oracle log file.                     -->
    <ValNameList xmlns="http://www.oblix.com" ListName="LogAll2File">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ALL" />
      <NameValPair ParamName="LOG_WRITER" Value="FileLogWriter" />
      <NameValPair ParamName="FILE_NAME" Value="oblog.log" />
      <!-- Buffer up to 64 KB (expressed in bytes) of log entries before
       flushing to the file.     -->
      <NameValPair ParamName="BUFFER_SIZE" Value="65535" />
      <!--Rotate the log file once it exceeds 50 MB (expressed in bytes). -->
      <NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" />
      <!--Rotate the log file after 24 hours (expressed in seconds). -->
      <NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
  </CompoundList>
<!--  List of values that can be specified in the module config      -->
<!--                                                                 -->
<!--  On - Uses loglevel set in the loglevel threshold               -->
<!--  Off - No information is logged                                 -->
<!--  LOGLEVEL_FATAL - serious error, possibly a program halt.       -->
<!--  LOGLEVEL_ERROR - a transient or self-correcting problem.       -->
<!--  LOGLEVEL_WARNING - a problem that does not cause an error.     -->
<!--  LOGLEVEL_INFO - reports the current state of the component.    -->
<!--  LOGLEVEL_DEBUG1 - basic debugging information.                 -->
<!--  LOGLEVEL_DEBUG2 - advanced debugging information.              -->
<!--  LOGLEVEL_DEBUG3 - logs performance-sensitive code.             -->
<!--  LOGLEVEL_TRACE - used when you need to trace the code path     -->
<!--  execution or capture metrics. Includes all previous levels.    -->
<!--                                                                 -->
<!--  List of modules that can be specified in the module config     -->
<!--                                                                 -->
<!--  ALL_MODULES - Applies to all log modules                       -->
<!--  Specific module name - Applies to specific module              -->
<!--                                                                 -->
<!--                                                                 -->
<!--    <ValNameList                                                 -->
<!--        xmlns="http://www.oblix.com"                             -->
<!--        ListName="MODULE_CONFIG">                                -->
```

```
<!--        <NameValPair                                             -->
<!--            ParamName="CONNECTIVITY"                             -->
<!--            Value="LOGLEVEL_TRACE"></NameValPair>                -->
<!--    </ValNameList>                                               --><!--
<!--FILTER_LIST is used to maintain list of attributes which need         -->
<!-- to be treated as sensitive and hence will be filtered out from   -->
<!-- from logs. FILTER_LIST consist of all attribute names along      -->
<!-- with corresponding masking lengths.There should be separate      -->
<!-- entry in the list for the display name of the attribute          -->
<!-- identified as sensitive. All attributes configured are case      -->
<!-- sensitive i.e. if we configured sensitive attribute homePhone    -->
<!-- as HomePhone then it will not get filtered out from logs.        -->
<!-- By default four attributes (password, Password, response and     -->
<!-- Response) are configured as sensitive                            -->
<!-- A sample configuration is shown below                            -->

<!-- <ValNameList                                                    -->
<!--    xmlns="http://www.oblix.com"                                 -->
<!--    ListName="FILTER_LIST">                                      -->
<!--    <NameValPair                                                 -->
<!--      ParamName="password"                                       -->
<!--      Value="40"></NameValPair>                                  -->
<!--    <NameValPair                                                 -->
<!--      ParamName="Password"                                       -->
<!--      Value="40"></NameValPair>                                  -->
<!--    <NameValPair                                                 -->
<!--      ParamName="response"                                       -->
<!--      Value="40"></NameValPair>                                  -->
<!--     <NameValPair                                                -->
<!--      ParamName="Response"                                       -->
<!--      Value="40"></NameValPair>                                  -->
<!--    <NameValPair                                                 -->
<!--      ParamName="homePhone"                                      -->
<!--      Value="40"></NameValPair>                                  -->
<!--  </ValNameList>                                                 -->
  <ValNameList xmlns="http://www.oblix.com" ListName="FILTER_LIST">
    <NameValPair ParamName="password" Value="40" />
    <NameValPair ParamName="Password" Value="40" />
    <NameValPair ParamName="passwd" Value="40" />
    <NameValPair ParamName="Passwd" Value="40" />
    <NameValPair ParamName="response" Value="40" />
    <NameValPair ParamName="Response" Value="40" />
  </ValNameList>
</CompoundList>
```

About Directing Log Output to a File or the System File

To send log output to a destination, you configure a log writer. A log writer can send log output to one, none, or both of the following:

- A log file.

  This file resides under the root installation directory of the component.

- The system file of the host for the component.

  If more than one component resides on the same host, all components send data to the system log file on that host.

You can send logs of a particular level, or logs of different levels, to more than one type of log writer. For instance, you can send Fatal data to the system log, and send Trace data to a file. Or, you can send Fatal data to both the system log and a file.

You define log writers in the log configuration file using the `LOG_WRITER` parameter in a log-handler definition. See "The Second Compound List and Log Handlers" on page 23-13 for details.

The log writers are described in Table 23–3.

Table 23–3 Log Writers

Writer	Description
`SysLogWriter`	Sends data to the system log file for the computer that hosts the component being logged. Typically, the system log file contains event information from multiple applications and the host operating system. For Windows, this is the application log file located at My Computer, Manage, Event Viewer, Application. For UNIX platforms, the name and location of the system log file can vary according to the computer and the preferences of the system administrator. Consult the administrator of the computer for the file location. The default log configuration file sends Fatal, Error, and Warning messages to the system log file.
`FileLogWriter`	This writer is recommended when you want to save log data for an OAM Server or other single-process application on a disk file. The FileLogWriter opens the log file and holds it open for disk writes until the approximate file size limit or file rotation interval has been reached. Oracle does not recommend this log writer for situations where more than one process needs to write to the same log file. For these situations, use the MPFileLogWriter.
`MPFileLogWriter`	This writer resembles the FileLogWriter, except that it opens and closes the log file each time it writes data to the file. This enables multiple processes to write to the file in turn. However, this practice can slow performance substantially. Oracle recommends using MPFileLogWriter only when FileLogWriter fails to record logging data from some of the processes associated with a multi-process application, for example, an Access Client installed on a multi-process Web server (such as Apache) or the Solaris version of the iPlanet Web server.

Structure and Parameters of the Log Configuration File

The log configuration file conforms to a standard format. You can edit parameters and add or subtract sections known as log-handler definitions, but do not change the underlying format of the log configuration file.

See Example 23–1 or Example 23–8 for a listing of the default log configuration file.

The rest of this section discusses the following topics:

- The Log Configuration File Header
- The Initial Compound List
- The Simple List and Logging Threshold
- The Second Compound List and Log Handlers
- The List for Per-Module Logging

- The Filter List
- About XML Element Order

The Log Configuration File Header

At the beginning of the log configuration file there is an XML file header:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

The header serves the following purposes:

- The header declares the relevant XML version, which is always 1.0.
- It also declares the encoding format, which is always ISO-8559-1.

The Initial Compound List

The header is followed by an initial compound list that is delimited as follows:

```
<CompoundList xmlns="http://www.oblix.com" ListName="logframework.xml.staging">
. . .
</CompoundList>
```

The first compound list is structured as follows:

- The compound list start-tag shows the relevant XML name space for the log configuration file in the `xmlns` parameter.
- The compound list start-tag also provides a name for the compound list in the `ListName` parameter.
- The compound list end-tag occurs near the end of the file.

  This compound list delimits all log configuration information.

The Simple List and Logging Threshold

After the start-tag for the first compound list, a simple list sets the global defaults for logging, as follows:

```
<SimpleList>
. . .
</SimpleList>
```

Between the start and end tags of the simple list, you configure the following:

Table 23–4 Global Parameters in the First Compound List

Parameter	Description
LOG_LEVEL_THRESHOLD	Sets the default logging threshold. Default value: LOGLEVEL_WARNING Possible Values: Refer to log levels in "About Log Levels" on page 23-2 The global threshold allows logs of a particular level and more general levels to be collected, and prevents lower-level logs from being collected. This threshold can be overridden by a per-module threshold. See "Configuring Different Threshold Levels for Different Types of Data" on page 23-21 for details.

Table 23–4 (Cont.) Global Parameters in the First Compound List

Parameter	Description
SECURE_LOGGING	Dynamically enables or disables the secure logging mechanism. This does not require a server or component restart. Default value: On Possible Values: On or Off
LOG_SECURITY_THRESHOLD_LEVEL	Indicates the log threshold for which secure logging is effective. Default value: LOGLEVEL_TRACE Possible Values: Refer to log levels in "About Log Levels" on page 23-2 Note: Ensure that LOG_THRESHOLD_LEVEL and LOG_SECURITY_THRESHOLD_LEVEL are the same or are consistent with one another. For example, if LOG_THRESHOLD_LEVEL is set to LOGLEVEL_TRACE while LOG_SECURITY_THRESHOLD_LEVEL is set at LOGLEVEL_WARNING, then secure logging applies to LOGLEVEL_WARNING and above but does not apply to LOGLEVEL_TRACE.
LOG_SECURITY_ESCAPE_CHARS	Configure escape sequence characters used to avoid additional information being overwritten due to the secure logging mechanism. Use a comma separated list. Default value:),] Possible Values: Characters only Note: Default values are recommended. Configuring inappropriate characters may lead to sensitive information being unmasked.
LOG_SECURITY_MASK_LENGTH	Specifies the default masking length if none is specified in FILTER_LIST. Default value: 300 Possible Values: Positive integer **Note**: FILTER_LIST appears after the second compound list (log handlers). For more information, see "Filtering Sensitive Attributes" on page 23-26.

Example 23–2 shows the simple lists containing global settings, which appear in the first compound list in the oblog_config_wg.xml file.

Example 23–2 Simple Lists with Global Settings (First Compound List in oblog_config_wg.xml)

```
<SimpleList>
   <NameValPair
      ParamName="LOG_THRESHOLD_LEVEL"
      Value="LOGLEVEL_WARNING">
   </NameValPair>
   <NameValPair
      ParamName="AUTOSYNC"
      Value="True">
</NameValPair>
   <NameValPair
      ParamName="SECURE_LOGGING"
      Value="On">
</NameValPair>
   <NameValPair
      ParamName="LOG_SECURITY_THRESHOLD_LEVEL"
      Value="LOGLEVEL_TRACE">
</NameValPair>
   <NameValPair
      ParamName="LOG_SECURITY_ESCAPE_CHARS"
      Value="),]">
</NameValPair>
   <NameValPair
      ParamName="LOG_SECURITY_MASK_LENGTH"
      Value="300">
</NameValPair>
</SimpleList>
```

The Second Compound List and Log Handlers

After the simple list containing global settings, and within the start and end tags for the initial compound list, you specify an additional compound list. This compound list contains log-handler definitions. The start and end tags for this list are as follows:

```
<CompoundList xmlns="http://www.oblix.com" ListName="LOG_CONFIG">
. . .
</CompoundList>
```

This compound list tag is configured as follows:

- In the start tag for the compound list, the `xmlns` parameter indicates the relevant XML name space.
- Also in the start tag, you specify the name of the list on the `ListName` parameter.

  Typically, the name of this list is `LOG_CONFIG`.

Between the start and end tags for the compound list for the log-handler, you specify one or more `ValNameList` elements. Each `ValNameList` element contains the definition for a log-handler. Each instance of this element begins and ends as follows:

```
<ValNameList xmlns="http://www.oblix.com" ListName="Unique_Name">
. . .
</ValNameList>
```

The `ValNameList` elements are configured as follows:

- The opening tag sets the relevant XML name space on the `xmlns` parameter.
- The opening tag also sets a name for the log-handler on the `ListName` parameter.

Within the opening and closing `ValNameList` tags, you configure the log-handler. A log-handler definition contains three mandatory `NameValPair` elements:

- The first mandatory `NameValPair` element defines the logging level for the log-handler.

  This element contains the statement `ParamName="LOG_LEVEL"`, whose value is a reserved name in Table 23–1, as follows:

  ```
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL" />
  ```

- The second mandatory `NameValPair` element defines the destination for log output.

  This element contains a statement `ParamName="LOG_WRITER"`, whose value is a reserved name in Table 23–3, as follows:

  ```
  <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
  ```

- The third mandatory `NameValPair` element toggles this log-handler on and off.

  This element contains a statement `ParamName="LOG_STATUS"`, with a value of `On` or `Off`, as follows:

  ```
  <NameValPair ParamName="LOG_STATUS" Value="On" />
  ```

Finally, within the opening and closing `ValNameList` tags, if you specify `FileLogWriter` or `MPFileLogWriter` as the log writer, you can add none, some, or all of the following. See Table 23–7 for details:

- A destination file name, as follows:

  ```
  <NameValPair ParamName="FILE_NAME" Value="oblog.log" />
  ```

- A buffer size, as follows:

  ```
  <NameValPair ParamName="BUFFER_SIZE" Value="65535" />
  ```

- A a file size that determines when a new log file is generated, as follows:

  ```
  <NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" />
  ```

- A time in minutes that determines the interval at which a new log file is generated, as follows:

  ```
  <NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" />
  ```

The List for Per-Module Logging

After the end tag for the compound list that delimits the log-handlers, and before the end tag for the initial compound list, you can add per-module logging parameters.

See "Configuring Different Threshold Levels for Different Types of Data" on page 23-21 for details.

The Filter List

After the per-module logging parameters a filter list identifies sensitive information that you might want to filter out of the log file. For example, passwords and responses for lost password management are sensitive information that you might want to filter out of the log file.

Each name value pair associated with the FILTER_LIST parameter provides the name of a word or phrase to be checked before the log is written and the corresponding masking length for that word or phrase. During logging, the value of the word or phrase is masked and omitted from the log file.

Simply put, during logging Oracle Access Manager does not recognize whether a value to be masked is an attribute or its display name or something different (plain text). Secure Logging works by searching for words or phrases added in the FILTER_LIST and then masking out any data that is followed by the occurrence of those words or phrases. For example, in the following statement:

```
\csabuild\coreid1014\np_common\db\ldap\util\ldap_util3.cpp:3107 "ldap_parse_result
of Simple Bind"           ld handle^0x0779FA00       result^0x09FB0088
bind^cn=orcladmin        LDAP bind operation status code^0         Additional
error message^ freeit^0 parse_rc^0
```

After turning Secure Logging ON and adding “bind” in the FILTER_LIST (which is neither an attribute nor a display name), whatever follows the word in the FILTER_LIST (in this case, "bind") is masked. In this case, you would see the following in logs:

```
\csabuild\coreid1014\np_common\db\ldap\util\ldap_util3.cpp:3107 "ldap_parse_result
of Simple Bind"          ld handle^0x0779FA00       result^0x09FB0088
bind^cn=orcladmin        LDAP bind**********  status code^0         Additional
error message^ freeit^0 parse_rc^0
```

All attributes are case sensitive. For example, if you enter "password" instead of "Password" as a display name for an attribute, then "Password" is not filtered. By default, four attributes are always configured in the filter list: password, Password, response, and Response.

The default masking length, 40, is specified for each of the four default attributes. The default mask length can be altered for the default attributes if needed. If you add other attributes to the filter list, you might need a larger mask length (300, for example).

The default filter list is shown in Example 23–3.

Example 23–3 FILTER_LIST Masks Sensitive Attributes in Log Files

```
<ValNameList>
   xmlns="http://www.oblix.com"
   ListName="FILTER_LIST">
   <NameValPair
      ParamName="password"
      Value="40"></NameValPair>
   <NameValPair
      ParamName="Password"
      Value="40"></NameValPair>
   <NameValPair
      ParamName="passwd"
      Value="40"></NameValPair>
   <NameValPair
      ParamName="Passwd"
      Value="40"></NameValPair>
   <NameValPair
      ParamName="response"
      Value="40"></NameValPair>
   <NameValPair
      ParamName="Response"
      Value="40"></NameValPair>
</SimpleList>
```

When you add another attribute to the filter list, you must include the display name as well as the attribute name in the directory server.

About XML Element Order

When using XML, you can specify parallel elements in a list in any order as long as the elements remain intact and within the tags that originally bracketed them. For example, the lists in Example 23–4 and Example 23–5 are equivalent:

Example 23–4 Valid Name/Value List

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
   <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
   <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
   <NameValPair ParamName="LOG_STATUS" Value="On" />
```

```
</ValNameList>
```

Example 23–5 Another Valid Name/Value List

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
   <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
   <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
   <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
```

Similarly, within a given tag, the attributes (except for the tag name, which must always be the first element within the tag brackets) can be reordered, as long as they remain intact and within the tag elements that originally bracketed them. The opening tags for a name-value list in Example 23–6 and Example 23–7 are equivalent:

Example 23–6 Opening tag for a Name/Value List

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
```

Example 23–7 Opening tag for a Name/Value List

```
<ValNameList ListName="LogError2Sys" xmlns="http://www.example.com">
```

About Activating and Suppressing Logging Levels

Several factors determine if logging is active for a particular log-handler. Table 23–5 lists these factors.

Table 23–5 Factors that Determine Whether Logging Is Active

Factor	Importance	Description
LOG_THRESHOLD_LEVEL	Primary	This parameter sets a cutoff for logging. Any log level that is more detailed than the threshold is suppressed. See Table 23–1 for valid log levels. You override this parameter for a subset of items that can be logged using the MODULE_CONFIG parameter. See "Configuring Different Threshold Levels for Different Types of Data" on page 23-21 for details.
MODULE_CONFIG	Primary	This sets a per-module override for the global logging threshold. See "Configuring Different Threshold Levels for Different Types of Data" on page 23-21 for details.
LOG_STATUS	Secondary	This parameter toggles logging on or off, as long as it is not overridden by the logging threshold or a module-specific override.
The physical position of a log handler	Secondary	See "About Log Handler Precedence" on page 23-16.

About Log Handler Precedence

You can configure up to three log-handler definitions for a single log level in a log configuration file. Three different log handlers are required to send output for a particular log level to each of the three log writers described in Table 23–3.

If you specify different LOG_STATUS settings in these log handlers, the setting in the log-handler definition closest to the physical end of the log configuration file sets the status for the other log-handler definitions of the same log level. For example, you can set LOG_STATUS to Off for the first two log handlers for the Error log level, but if LOG_STATUS is On for the third and final log handler in the configuration file, logging still occurs for all three handlers.

The LOG_STATUS settings are moot if that level is more fine-grained than the current LOG_THRESHOLD_LEVEL. In this case, logging cannot be activated at this level unless the threshold is overridden by a module-specific threshold. See "Configuring Different Threshold Levels for Different Types of Data" on page 23-21 for details.

Mandatory Log-Handler Configuration Parameters

At minimum, each log-handler definition contains five parameters listed in Table 23–6.

Table 23–6 Mandatory Log Configuration File Parameters

Parameter	Comment
xmlns	This parameter is specified in the opening ValNameList tag. It specifies the relevant XML namespace for the current list and is identical for all log-handler definitions in a given logging configuration file. Example: http://www.example.com
ListName	This parameter is specified in the opening ValNameList tag. Where possible, use the default names. When creating a new log-handler definition, select a memorable name that you cannot confuse with other log handlers. Examples: WarningsAndAboveToSyslog sends Fatal, Error, and Warning messages to the system log file. WarningsOnlyToFileLog128KBuffer sends messages from just the Warning level to a 128KB buffer, and hence to a disk file. TraceOnlyToMPRotateDaily sends messages from just the Trace level to the multi-process file writer, which opens and closes the file each time it writes to disk. This file is replaced with a fresh (empty) file every day, regardless of the size of the file at the time of replacement.
LOG_LEVEL	This specifies a log level. See Table 23–1 for details. The default logging configuration file activates logging for three levels: Fatal, Error, and Warning.
LOG_WRITER	This specifies the destination for log output for this log-handler. See Table 23–3 for details. The default log configuration file sends output to both the system log and the log data file for the component doing the logging.
LOG_STATUS	This parameter turns the log handler on or off.

If you specify FileLogWriter or MPFileLogWriter as the value for the LOG_WRITER parameter, the four parameters in Table 23–7 are relevant.

Table 23–7 Log Data File Configuration Parameters

Parameter	Description	Default
`FILE_ NAME`	Mandatory. Used only for the FileLogWriter or MPFileLogWriter. It is the name and location of the file where log data is written. You can prepend an absolute path to the file name to store it somewhere other than the default location, which is: *Webgate_install_dir*\oblix\logs Where *component_install_dir* is the root installation directory for the component whose system events you are logging. When you create more than one log-handler definition that sends output to FileLogWriter or MPFileLogWriter, provide unique file names so that multiple handlers do not write to the same file. This caution does not apply to log handlers accessing the SysLogWriter.	oblog.log
`BUFFER_SIZE`	Optional. This is the size of the buffer, in bytes, for logged data as it is being written to the log file. If you set the buffer value to 0 or a negative number, the default value is used. To write to the log file immediately, without buffering, set the value to a small number, for example, 5. Oracle recommends that you set a small buffer size in situations where there are system failures.	65535 (64KB)
`MAX_ ROTATION_ SIZE`	Optional. When the log file reaches this size (in bytes), a time stamp is appended to the file name, for example "oblog.log" becomes "oblog.log1081303126." New data is written to the file with the original name.	52428800 (512KB)
`MAX_ ROTATION_ TIME`	Optional. A time interval, in seconds, when the log file is renamed, whether or not it has reached the maximum rotation size. If the rotation time determines when the file is rotated, the numbers appended to the log files differ by the number of seconds in the rotation interval. For example, "oblog.log.1081389526" and "oblog.log.1081303126" differ by 84,600, which is the number of seconds in 24 hours. This is the rotation interval set in the log configuration file.	86400 (1 day, in seconds)

Settings in the Default Log Configuration File

As installed with each component, the log configuration file activates only the highest three levels (Fatal, Error, and Warning) and directs all log output to the system log.

On Windows, you can view the system log for the computer that hosts the component you are logging by navigating to My Computer, Manage, Event Viewer, Application. System event entries for the components being logged are interspersed among the system events for the operating system and applications other than Oracle Access Manager.

For Solaris and Linux environments, the location of the system log is recorded in a system configuration file whose particulars can vary from computer to computer. For the name and location of this system file or the system log, consult the owner of the computer that hosts the component whose system log you want to examine.

Example 23–8 shows the default log configuration file with comments removed to expose the file structure.

Example 23–8 A Default Log Configuration File Without Embedded Comments

```
<?xml version="1.0" encoding="utf-8"?>
<CompoundList
   xmlns="http://www.oblix.com
   ListName="oblog_config_wg.xml.staging">
   <SimpleList>
      <NameValPair
         ParamName="LOG_THRESHOLD_LEVEL"
         Value="LOGLEVEL_WARNING"></NameValPair>
   </SimpleList>
   <SimpleList>
      <NameValPair
         ParamName="AUTOSYNC"
         Value="True"></NameValPair>
   </SimpleList>
   <SimpleList>
      <NameValPair
          ParamName="SECURE_LOGGING"
          Value="On"></NameValPair>
   </SimpleList>
   <SimpleList>
      <NameValPair
         ParamName="LOG_SECURITY_THRESHOLD_LEVEL"
         Value="LOGLEVEL_TRACE"></NameValPair>
   </SimpleList>
   <SimpleList>
      <NameValPair
         ParamName="LOG_SECURITY_ESCAPE_CHARS"
         Value="),]"></NameValPair>
   </SimpleList>
   <SimpleList>
      <NameValPair
        ParamName="LOG_SECURITY_MASK_LENGTH"
        Value="300"></NameValPair>
   </SimpleList>
   <CompoundList
      xmlns="http://www.oblix.com"
      ListName="LOG_CONFIG">
      <ValNameList
         xmlns="http://www.oblix.com"
         ListName="LogFatal2Sys">
         <NameValPair
            ParamName="LOG_LEVEL"
            Value="LOGLEVEL_FATAL"></NameValPair>
         <NameValPair
            ParamName="LOG_WRITER"
            Value="SysLogWriter"></NameValPair>
         <NameValPair
            ParamName="LOG_STATUS"
            Value="On"></NameValPair>
      </ValNameList>
      <ValNameList
         xmlns="http://www.oblix.com"
         ListName="LogAll2File">
         <NameValPair
            ParamName="LOG_LEVEL"
            Value="LOGLEVEL_ALL"></NameValPair>
         <NameValPair
            ParamName="LOG_WRITER"
            Value="FileLogWriter"></NameValPair>
         <NameValPair
            ParamName="FILE_NAME"
            Value="oblog.log"></NameValPair>
         <NameValPair
            ParamName="BUFFER_SIZE"
            Value="65535"></NameValPair>
         <NameValPair
            ParamName="MAX_ROTATION_SIZE"
            Value="52428800"></NameValPair>
         <NameValPair
            ParamName="MAX_ROTATION_TIME"
            Value="86400"></NameValPair>
         <NameValPair
            ParamName="LOG_STATUS"
            Value="On"></NameValPair>
      </ValNameList>
    </CompoundList>
    <ValNameList
       xmlns="http://www.oblix.com"
       ListName="FILTER_LIST">
       <NameValPair
          ParamName="password"
          Value="40"></NameValPair>
       <NameValPair
          ParamName="Password"
          Value="40"></NameValPair>
       <NameValPair
          ParamName="passwd"
          Value="40"></NameValPair>
       <NameValPair
          ParamName="Passwd"
          Value="40"></NameValPair>
       <NameValPair
          ParamName="response"
          Value="40"></NameValPair>
       <NameValPair
          ParamName="Response"
          Value="40"></NameValPair>
    </ValNameList>
  </CompoundList>
```

Description of the Settings in the Default Log Configuration File

The default configuration file sends Fatal, Error, and Warning messages to both the system log and to a log data file named oblog.log.

The simple list near the top of the file sets the following parameters:

- It sets the `LOG_THRESHOLD_LEVEL` to `Warning`.

  The threshold suppresses logging for levels that are more fine-grained than Warning. You can override this threshold. See "Configuring Different Threshold Levels for Different Types of Data" on page 23-21 for details.

The nested compound list contains four log-handler definitions:

- The first, named `LogFatal2Sys`, sets the logging level to `Fatal` and sets `LOG_STATUS` to On.

The threshold level is Warning, which is more fine-grained than Fatal, so this definition is in effect. The log output is written to the system log, as specified by the `LOG_WRITER` parameter.

- The `LogError2Sys` log-handler definition sends Error level messages to the system log.

  Error is located before the current threshold level (Warning), so this definition is in effect.

- The `LogWarning2Sys` definition sends Warning level output to the system log.

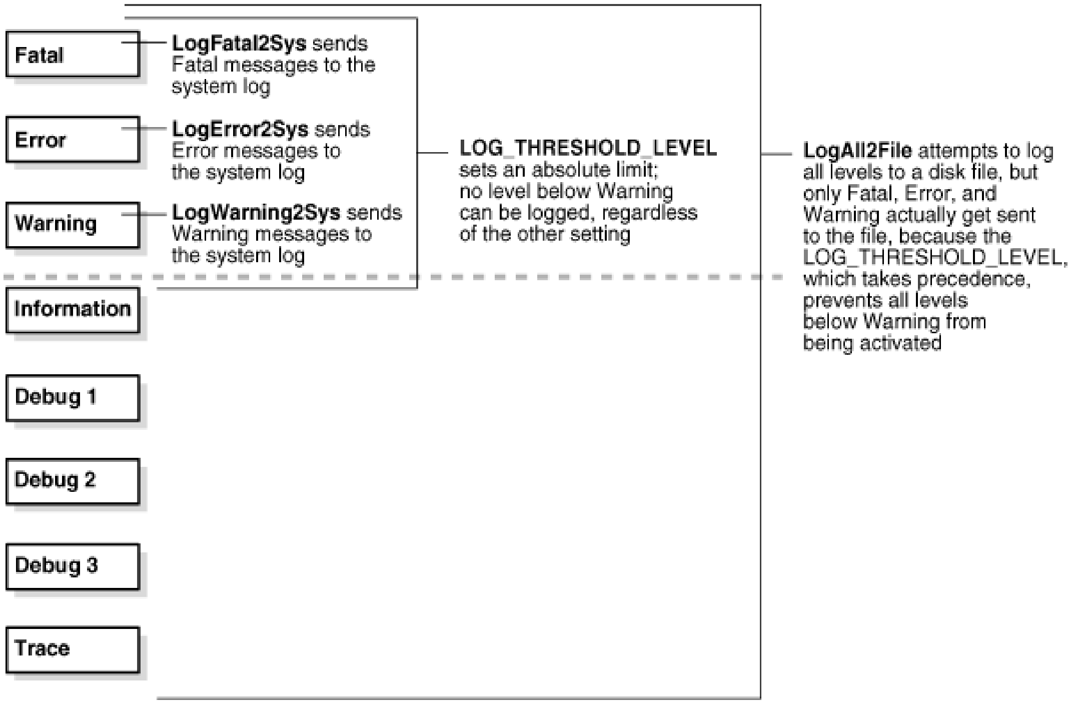
  Like the two previous log-handler definitions, it is not overridden by the current `LOG_THRESHOLD_LEVEL` parameter.

- `LogAll2File`, the final log-handler definition, appears to send output from all log levels to a disk file named oblog.log.

  The `LOG_THRESHOLD_LEVEL` parameter is set to Warning, so only the output from the Fatal, Error, and Warning levels are recorded in this log data file. Since output from `LogAll2File` goes to the `FileLogWriter`, the parameters governing file name, buffer size, rotation size, and rotation interval all take effect.

Figure 23–1 illustrates log-level activation in the default log confirmation file.

Figure 23–1 Log-Level Activation in the Default Log Configuration File


Fatal
Error
Warning
Information
Debug 1
Debug 2
Debug 3
Trace
LogFatal2Sys sends Fatal messages to the system log
LogError2Sys sends Error messages to the system log
LogWarning2Sys sends Warning messages to the system log
LOG_THRESHOLD_LEVEL sets an absolute limit; no level below Warning can be logged, regardless of the other setting
LogAll2File attempts to log all levels to a disk file, but only Fatal, Error, and Warning actually get sent to the file, because the LOG_THRESHOLD_LEVEL, which takes precedence, prevents all levels below Warning from being activated

Configuring Different Threshold Levels for Different Types of Data

When diagnosing a problem, you may not want detailed logs for every operation that a component performs. For example, to diagnose slow response times for requests that an Identity Server submits to its directory, you would want detailed information on LDAP operations and fewer details about other types of operations.

As of release 10.1.4.2, you can configure per-module or per-function threshold levels in the log configuration file, so that Oracle Access Manager generates detailed logs for some components while generating concise logs, or no logs, for others.

You configure per-module logging thresholds in a `MODULE_CONFIG` section in the oblog_config_wg.xml file. The `MODULE_CONFIG` section overrides the global default that you specify on the `LOG_THRESHOLD_LEVEL` in the simple list section of this file.

The rest of this section discusses the following topics:

- About the MODULE_CONFIG Section
- Configuring a Log Level Threshold for a Function or Module

About the MODULE_CONFIG Section

As described in "Structure and Parameters of the Log Configuration File" on page 23-10, in the log configuration file you configure a global logging threshold. The following is an example of the global `LOG_THRESHOLD_LEVEL` setting:

```
<SimpleList>
   <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
   . . .
</SimpleList>
```

In addition to the global threshold, the configuration file can contain a `ValNameList` that defines function- or module-specific log thresholds. The name of this list is always `MODULE_CONFIG`. Only one instance of this list is permitted in the log configuration file, and the information in the list applies to all log writers defined in the file. As of release 10.1.4.2, the default log configuration file contains a commented sample of the `MODULE_CONFIG` list.

Each item in the `MODULE_CONFIG` list sets a logging level for a module, as shown in the following example:

```
<ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">
    <NameValPair ParamName="LDAP" Value="LOGLEVEL_TRACE"></NameValPair>
    <NameValPair ParamName="DB_RUNTIME" Value="LOGLEVEL_TRACE"></NameValPair>
</ValNameList>
```

The elements in this section are as follows:

- The `ValNameList` tag delimits the list of per-module logging thresholds.
- One `NameValPair` tag delimits each specific per-module logging threshold.
- The `ParamName` parameter sets the name of a module or function.

  See Table 23–8 for a list of valid values.
- The `Value` parameter sets the logging threshold for the module that you specify as a value for the `ParamName` parameter.

  Table 23–1 lists the permissible values for the `Value` parameter. In addition to these values, you can specify the value `ON` to enable logging for the module and a value of `OFF` to disable logging for the specific module.

Location of the Per-Module Logging Section in the Log Configuration File

You add the per-module logging threshold section near the end of the log configuration file, after the closing tag for the compound list for the log-handlers and before the closing tag for the first compound list in the file.

This section contains an example of the per-module logging section. See "To configure a module-specific log threshold" on page 23-24 for details.

List of Modules That Can Be Logged

Table 23–8 describes the a partial list of the values that you can specify for the `ParamName` parameter in the `MODULE_CONFIG` list.

Table 23–8 ParamName Values You Can Configure for Per-Module Logging Threshold

ParamName Value	Logging Threshold That This Parameter Sets

`AAA_ACTIONS`	Sets a logging threshold for triggered actions that are configured as part of a policy in the OAM Server.
	`<ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">`
	`<NameValPair Paramname="AAA_ACTIONS" Value="OFF">`
	`</NameValPair>`

`AAA_AMENGINE`	Sets a logging threshold for activity performed by the Access Manager engine.
`AAA_ISRESRCOPPROT`	Sets a logging threshold for all OAM Server activities related to determining if a resource operation is protected.
`ACCESS_CLIENT`	Sets a logging threshold for operations performed by an access client, that is, an Access Client or Webgate.
`ACCESS_GATE`	Sets a logging threshold for operations performed by an Access Client.
`ACCESS_SDK`	Sets a logging threshold for operations performed by the Access Manager SDK interface. See the Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service for details.
`ACCESS_SERVER`	Sets a logging threshold for operations performed in the OAM Server.
`AM_SDK`	Sets a logging threshold for the Access Manager SDK. See the Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service for details.
`AUDIT`	Sets a logging threshold for auditing. See Chapter 24 for details.
`AUTHENTICATION`	Sets a logging threshold for user authentication operations.
`AUTHN_MGMT`	Sets a logging threshold for authentication scheme management.
`AUTHN_PLUGIN`	Sets a logging threshold for operations performed by an authentication plug-in.
`AUTHORIZATION`	Sets a logging threshold for user authorization operations.
`AUTHZ_MGMT`	Sets a logging threshold for authorization scheme management.
`AUTHZ_PLUGIN`	Sets a logging threshold for authorization plug-in operations.
`CACHE`	Sets a logging threshold for cache management and operations on the caches.
`CONN_MGMT`	Sets a logging threshold for connection management.
`CONN_RUNTIME`	Sets a logging threshold for connection run time.
`CONNECTIVITY`	Sets a logging threshold for client-sever connectivity and messaging.

Table 23–8 (Cont.) ParamName Values You Can Configure for Per-Module Logging

ParamName Value	Logging Threshold That This Parameter Sets
`DB_CONFIGURATION`	Sets a logging threshold for the data store interface layer configuration.
`DB_RUNTIME`	Sets a logging threshold for the data store interface layer run time.
`DIAGNOSTIC_FRAMEWORK`	Sets a logging threshold for the diagnostic framework.
`GROUPDB`	Sets the threshold for logging accesses of Group Manager data in the directory.
`GROUP_MGR`	Sets the threshold for logging Group Manager operations.
`HTTP_REQ`	Sets the threshold for logging HTTP request processing.
`IDXML`	Sets the threshold for logging IDXML operations. See the Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service for details.

`LDAP`	Sets a logging threshold for LDAP SDK, for example:
	`<ValNameList xmlns="http://www.oblix.com"`
	`ListName="MODULE_CONFIG">`
	`<NameValPair Paramname="LDAP" Value="LOGLEVEL_TRACE">`
	`</NameValPair>`

`NET`	Sets a logging threshold for network APIs.
`OBMYGROUPS`	Sets a logging threshold for ObMyGroups processing. This refers to searches of groups where the person who initiated the search is a member.
`OIS_CLIENT`	Sets a logging threshold for the Identity client.
`POLICY_MGMT`	Sets a logging threshold for policy and policy domain management.
`PPP`	Sets a logging threshold for Identity Event Plug-in API operations. See the Oracle Fusion Middleware Developer's Guide for Oracle Access Manager and Oracle Security Token Service for details.
`QUERY_BUILDER`	Sets a logging threshold for Query Builder operations.
`SECURITY`	Sets a logging threshold for the security and encryption library.
`SELECTOR`	Sets a logging threshold for Selector operations.
`SERVER`	Sets a logging threshold for server infrastructure.
`SSOTOKEN`	Single sign-on token management.
`UTILS`	Sets a logging threshold for utility classes.
`WEB`	Sets a logging threshold for the Web server plug-in interface.
`XML`	Sets a logging threshold for the XML Infrastructure.

Configuring a Log Level Threshold for a Function or Module

The following procedure describes how to configure a function- or module-specific log level threshold.

To configure a module-specific log threshold

1. Open the log configuration file in the following location:

   Webgate_install_dir\identity|access\oblix\config

2. If a ValNameList section with a ListName of MODULE_CONFIG does not already exist in this file, create one that is similar to the following:

   ```
   <ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">
   </ValNameList>
   ```

   Place this list after the end tag for the compound list that contains the log handler definitions. If there are comments immediately after this end tag, place the list after the comments.

3. Between the opening and closing tags of the new ValNameList element, configure one or more NameValPair elements.

   This element contains a ParamName parameter and a Value parameter. See Table 23–8 for the modules that you can supply on the ParamName parameter. See Table 23–1 for values, or you can specify a value of On or Off. The following is an example:

   ```
   <NameValPair ParamName="LDAP" Value="LOGLEVEL_TRACE"></NameValPair>
   ```

   You can specify multiple ValNamePair elements within the ValNameList.

   A complete per-module logging threshold section is illustrated in **bold** in the following example:

   ```
   <!--  ============================================================   -->
   <!--  Configure the Log Level                                        -->
   . . .
   <CompoundList xmlns="http://www.oblix.com" ListName="LOG_CONFIG">

   <!-- Write all FATAL logs to the system logger. -->
   <ValNameList xmlns="http://www.oblix.com" ListName="LogFatal2Sys">
        <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL">
        </NameValPair>
        <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter">
        </NameValPair>
        <NameValPair ParamName="LOG_STATUS" Value="On">
        </NameValPair>
   </ValNameList>
   . . .
   </CompoundList>
   <!--  List of values that can be specified in the module config      -->
   <!--                                                                 -->
   <!--  On - Uses loglevel set in the loglevel threshold               -->
   <!--  Off - No information is logged                                 -->
   <!--  LOGLEVEL_FATAL - serious error, possibly a program halt.       -->
   <!--  LOGLEVEL_ERROR - a transient or self-correcting problem.       -->
   <!--  LOGLEVEL_WARNING - a problem that does not cause an error.     -->
   <!--  LOGLEVEL_INFO - reports the current state of the component.    -->
   <!--  LOGLEVEL_DEBUG1 - basic debugging information.                 -->
   <!--  LOGLEVEL_DEBUG2 - advanced debugging information.              -->
   <!--  LOGLEVEL_DEBUG3 - logs performance-sensitive code.             -->
   <!--  LOGLEVEL_TRACE - used when you need to trace the code path     -->
   <!--  execution or capture metrics. Includes all previous levels.    -->
   <!--                                                                 -->
   <!--  List of modules that can be specified in the module config     -->
   <!--                                                                 -->
   <!--  ALL_MODULES - Applies to all log modules                       -->
   <!--  Specific module name - Applies to specific module              -->
   <!--                                                                 -->
   <!--                                                                 -->
   <!--    <ValNameList                                                 -->
   ```

```
<!--        xmlns="http://www.oblix.com"                                   -->
<!--        ListName="MODULE_CONFIG">                                      -->
<!--        <NameValPair                                                   -->
<!--            ParamName="CONNECTIVITY"                                   -->
<!--            Value="LOGLEVEL_TRACE"></NameValPair>                      -->
<!--    </ValNameList>                                                     -->

    <ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG">
        <NameValPair ParamName="LDAP" Value="LOGLEVEL_TRACE"></NameValPair>
        <NameValPair ParamName="DB_RUNTIME" Value="LOGLEVEL_TRACE">
        </NameValPair>
    </ValNameList>

</CompoundList>
```

Filtering Sensitive Attributes

As described earlier, you can activate secure logging and expand the default filter list to mask sensitive information from the log file.

When you add another attribute to the filter list, you must include the display name as well as the attribute name in the directory server. The following procedure describes how to perform this task. In this example, you are instructed to filter the user's home phone number: display name Home Phone; attribute name homePhone. However, you can filter the attribute of your choice.

Note: Each value added to FILTER_LIST increases the runtime cost of using Secure Logging.

Oracle recommends that you optimize the use of FILTER_LIST to reduce the runtime cost. For example, rather than adding two ParamName variations ("User Password" and "userPassword"), you could use only one. Using "Password" as the ParamName masks values for "User Password", "userPassword", and other words that end with "Password". Also, instead of including both "Home Phone" and "homePhone" in FILTER_LIST, you could use only "Phone".

See Also:

- "About Logging, Log Levels, and Log Output" on page 23-1
- "The Simple List and Logging Threshold" on page 23-11
- "The Filter List" on page 23-14
- "Settings in the Default Log Configuration File" on page 23-18

To add sensitive attributes to the filter list

1. Open the log configuration file in a text editor:

   Webgate_install_dir\identity|access\oblix\config\oblog_config_wg.xml

2. In oblog_config_wg.xml:

   a. Confirm that secure logging is active. For example:

   ```
   <SimpleList>
      <NameValPair
         ParamName="SECURE_LOGGING"
         Value="On"></NameValPair>
   </SimpleList>
   ```

b. Locate the FILTER_LIST parameter at the end of the file. For example:

```
<ValNameList xmlns="http://www.oblix.com" ListName="FILTER_LIST">
   <NameValPair ParamName="password" Value="40" />
   <NameValPair ParamName="Password" Value="40" />
   <NameValPair ParamName="response" Value="40" />
   <NameValPair ParamName="Response" Value="40" />
</ValNameList>
```

c. Add the display name to mask and the value for the mask length, then add the attribute and the value for the mask length. For example:

```
   <NameValPair ParamName="Home Phone" Value="300" />
   <NameValPair ParamName="homePhone" Value="300" />
```

Note: For testing, set the LOG_THRESHOLD_LEVEL and LOG_SECURITY_THRESHOLD_LEVEL to TRACE. See Step 6a.

d. Confirm that LOG_THRESHOLD_LEVEL and LOG_SECURITY_THRESHOLD_LEVEL are at the same level or are consistent with each other, as described in Table 23–4 on page 23-11. For example:

```
<SimpleList>
   <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
</SimpleList>
...
<SimpleList>
   <NameValPair ParamName="LOG_SECURITY_THRESHOLD_LEVEL" Value="LOGLEVEL_
   WARNING" />
</SimpleList>
```

e. Save the oblog_config_wg.xml file.

3. **Filtering User Password**: Perform the following steps and see "The Filter List" on page 23-14:

In the filter list in oblog_config_wg.xml, add the User Password display name and the corresponding attribute, and set the mask length for each. For example:

```
<ValNameList xmlns="http://www.oblix.com" ListName="FILTER_LIST">
  ...
   <NameValPair ParamName="User Password" Value="40" />
   <NameValPair ParamName="userPassword" Value="40" />
</ValNameList>
```

4. Test secure logging and filtering of sensitive information as follows:

a. In the oblog_config_wg.xml file, set the LOG_THRESHOLD_LEVEL and LOG_SECURITY_THRESHOLD_LEVEL to TRACE:

```
   <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_TRACE" />
...
   <NameValPair ParamName="LOG_SECURITY_THRESHOLD_LEVEL" Value="LOGLEVEL_
   TRACE" />
```

b. Perform a task that involves the component for which you have configured secure logging. For example:

Access a resource

View or modify the value of the attribute in the user's profile: Home Phone (if the filtered attribute is homePhone).

c. Check the oblog and confirm that the filtered attribute value is masked by a string like ***********.

Webgate_install_dir/access/oblix/log/oblog.log

d. In the oblog_config_wg.xml file, reset the `LOG_THRESHOLD_LEVEL` and `LOG_SECURITY_THRESHOLD_LEVEL` to the desired level for your enterprise.

e. Adjust the mask length of filtered attributes if needed in the oblog_config_wg.xml file. For example:

```
<NameValPair ParamName="Home Phone" Value="340" />
<NameValPair ParamName="homePhone" Value="340"/>
```

5. Repeat Steps 1 through 6 for each component in your deployment with one or more masked attributes.

24 Auditing Administrative and Run-time Events

In Oracle Access Manager, Oracle Security Token Service, and Oracle Fusion Middleware, 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 administrative and run-time events that can be audited for Oracle Access Manager and Oracle Security Token Service. Configuring common auditing settings and validating your auditing configuration is the subject of this chapter. Analyzing and using audit data is outside the scope of this chapter.

> **Note:** Unless explicitly stated, information in this chapter is the same whether you are using Oracle Access Manager alone or with Oracle Security Token Service.

This chapter includes the following topics:

- Prerequisites
- Introduction to Auditing
- Oracle Access Manager Events You Can Audit
- Setting Up Auditing for Oracle Access Manager with Oracle Security Token Service
- Validating Oracle Access Manager Auditing and Reports

24.1 Prerequisites

This section identifies requirements for tasks in this chapter: Review Introduction to Auditing

24.2 Introduction to 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 and Oracle Security Token Service use 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 Oracle Access Manager 10g was based on OAM policies. However, auditing for Oracle Access Manager and Oracle Security Token Service is based on configuration parameters set in the Oracle Access Manager Console which enables data capture for a user or set of users.

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 Oracle Access Manager Auditing Configuration
- About Oracle Access Manager Audit Record Storage
- About Audit Reports and Oracle Business Intelligence Publisher
- About the Audit Log

See Also:

- "Introduction to Auditing for Oracle Security Token Service" on page 17-21
- "Introduction to Oracle Fusion Middleware Audit Framework" in the Oracle Fusion Middleware Application Security Guide

24.2.1 About Oracle Access Manager Auditing Configuration

An Administrator controls certain auditing parameters using the Oracle Access Manager Console. This 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. Oracle Access Manager does not use JPS infrastructure to configure the audit configuration. There are no WebLogic Scripting Tool (WLST) commands for Oracle Access Manager auditing.

Within the Oracle Access Manager 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:

- "Oracle Access Manager Events You Can Audit" on page 24-5
- "Oracle Security Token Service Administrative Events You Can Audit" on page 17-23
- "Oracle Security Token Service Run-time Events You Can Audit" on page 17-25

24.2.2 About Oracle Access Manager 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.

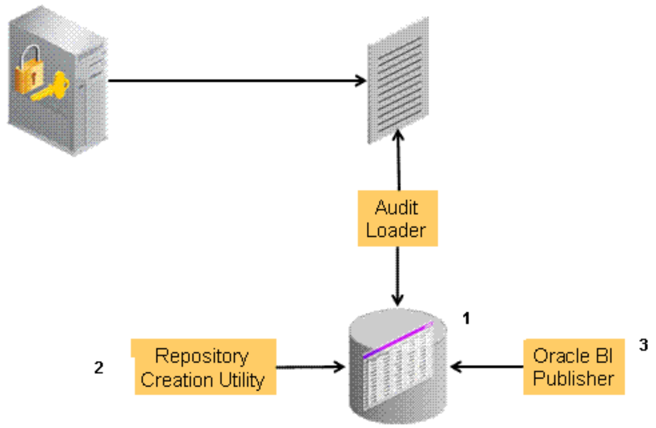
In production environments, Oracle recommends using a database audit store to provide scalability and high-availability for the Common Audit Framework. Audit data is cumulative and grows over time. Ideally this is a database for only audit data; not used by other applications.

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 24–1 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 24–1 Audit to Database Architecture



See Also:

- Oracle Fusion Middleware Application Security Guide
- "Setting Up the Audit Database Store" on page 24-11

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.

24.2.3 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 24-12.

24.2.4 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

24.3 Oracle Access Manager Events You Can Audit

This section provides the following topics:

- Oracle Access Manager Administrative Events You Can Audit
- OAM Run-time Events You Can Audit
- About Authentication Event Auditing

See Also:

- "Auditing Oracle Security Token Service Administrative and Run-time Events" on page 17-22

24.3.1 Oracle Access Manager Administrative Events You Can Audit

Administrative events are those generated when the Oracle Access Manager Console is used.

The Oracle Access Manager-specific administrative events that can be audited and the details captured in them are listed in Table 24–1. These event definitions and configurations are implemented as part of the audit service in Oracle Platform Security Services.

Note: With Oracle Access Manager 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 Oracle Access Manager 11g.

Table 24–1 Oracle Access Manager Administrative Audit Events

Administrative Event	Event Data Include
Oracle Access Manager 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
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

Table 24–1 (Cont.) Oracle Access Manager Administrative Audit Events

Administrative Event	Event Data Include
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
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

Table 24–1 (Cont.) Oracle Access Manager Administrative Audit Events

Administrative Event	Event Data Include
Server configuration change	■ New details ■ Old details ■ Instance Name ■ Host Name ■ Application Name ■ User Name ■ Remote ID ■ Roles ■ Date and time

24.3.2 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 24–2. 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 24–2 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

Table 24–2 (Cont.) OAM Run-time Audit Events

Run-time Event	Issued When	Event Details Include
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
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

Table 24–2 (Cont.) OAM Run-time Audit Events

Run-time Event	Issued When	Event Details Include
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

24.3.3 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.

24.4 Setting Up Auditing for Oracle Access Manager with Oracle Security Token Service

The following overview provides a list of the tasks that must be performed before you can perform auditing.

See Also: "Auditing Oracle Security Token Service Administrative and Run-time Events" on page 17-22

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 24-11.
2. Set up publishing for audit reports, as described in "Preparing Oracle Business Intelligence Publisher EE" on page 24-12.
3. Edit the Audit Configuration in the Oracle Access Manager Console, as described in:
   - About the Auditing Configuration Section in Oracle Access Manager Console
   - 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 24-14.

24.4.1 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.

24.4.2 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.

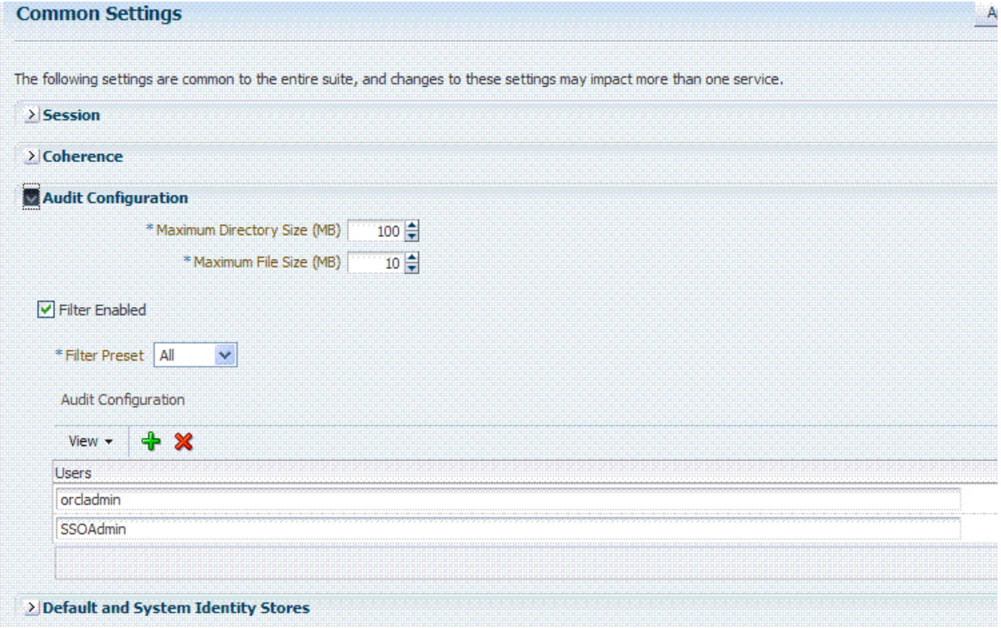
4. Set up audit report filters, as described in the section "Set Up Audit Report Filters" of the Oracle Fusion Middleware Application Security Guide.

5. View reports from the following path: Reports/Oracle_Fusion_Middleware_Audit reports.

24.4.3 About the Auditing Configuration Section in Oracle Access Manager Console

Within Oracle Access Manager, certain Audit Configuration settings are accessible as Common Settings under the System Configuration. These settings are not required when you audit to a database. Figure 24–2 shows the Audit Configuration section of the Common Settings page.

Figure 24–2 Common Settings: Auditing Configuration



Common Settings

The following settings are common to the entire suite, and changes to these settings may impact more than one service.

- Session
- Coherence
- Audit Configuration

*Maximum Directory Size (MB) 100

*Maximum File Size (MB) 10

☑ Filter Enabled

*Filter Preset All

Audit Configuration

Users
orcladmin
SSOAdmin

- Default and System Identity Stores

The Auditing section provides settings for the Log Directory, Filter Settings, and Audit Configuration Users.

> **Note:** The actual log directory cannot be configured using the Oracle Access Manager 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 24–3 describes the elements in the Audit Configuration page.

Table 24–3 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.

Table 24–3 (Cont.) Audit Configuration Elements

Elements	Description
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.
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.

24.4.4 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.

To view or edit auditing configuration in the Oracle Access Manager Console

1. From the System Configuration tab, Common Configuration section, double-click Common Settings in the navigation tree.
2. In the Audit Configuration section, enter appropriate details for your environment (Table 24–3):
   - Maximum Log directory size
   - Maximum Log file size
   - Filter settings to include specific users from the audit by clicking the Add (+) button above the Users table and entering a value in the field.
3. Click Apply to submit the Audit Configuration (or close the page without applying changes).
4. Restart AdminServer and OAM Servers after changes are applied.

24.5 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 13, "Managing Policies to Protect Resources and Enable SSO".
- Prepare BI EE Publisher as described in "Preparing Oracle Business Intelligence Publisher EE" on page 24-12.

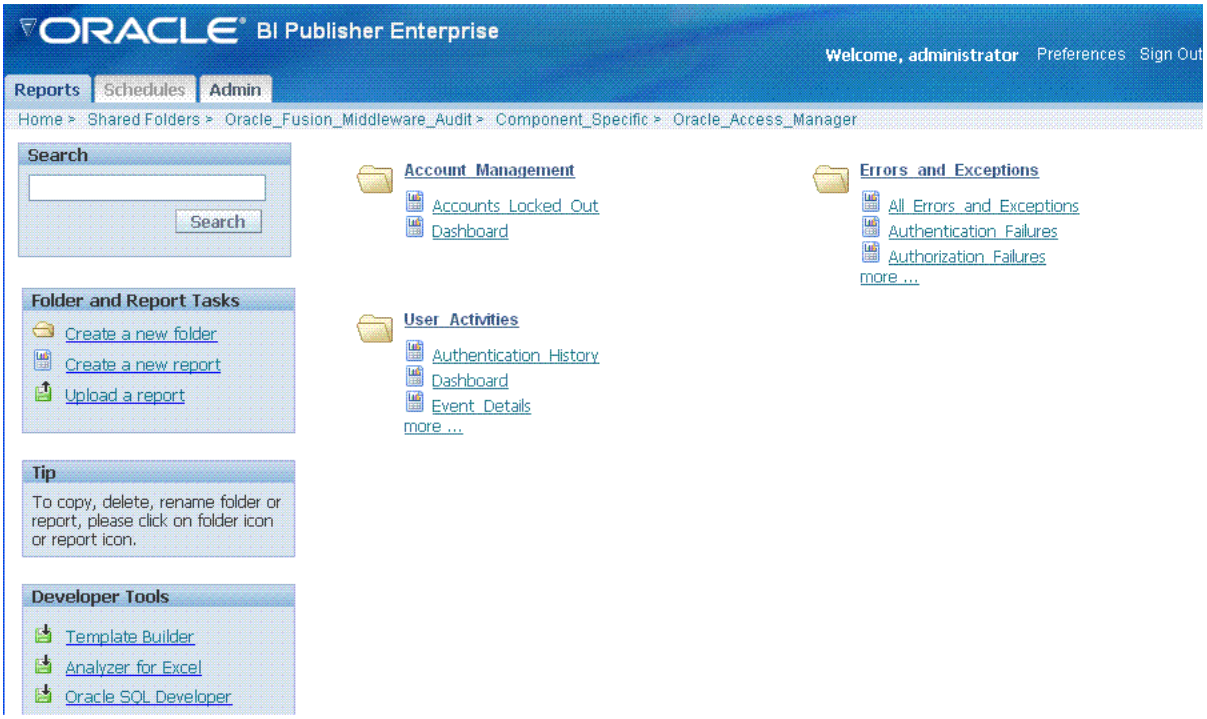
To validate your OAM 11g auditing configuration

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.

25

Monitoring Performance by Using Oracle Access Manager Console

There are several methods to view performance metrics. This chapter provides the following topics, with emphasis on using Oracle Access Manager Console:

- Introduction to Performance Monitoring
- Monitoring Server Performance Metrics Using the Console
- Monitoring SSO Agent Performance Metrics
- OXM Proxy Performance Tuning Parameters

See Also:

- Chapter 26 to use Oracle Enterprise Manager Fusion Middleware Control to monitor Oracle Access Manager performance metrics

25.1 Introduction to Performance Monitoring

Oracle Access Manager uses the Oracle Dynamic Monitoring Systems (DMS) to measure application-specific performance information for OAM Servers and registered OAM Agents.

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.

dmsSpy is a Fusion Middleware tool that is part of the WebLogic Application Server. dmsSpy displays the raw DMS data specific to the WebLogic Application Server instance. Displayed information is categorized by Noun Types (OAMS.OAM_ prefix for Oracle Access Manager 11g) and includes metrics pertaining to all DMS instrumented applications running in the Weblogic Application Server instance. To see the metrics on a Weblogic instance, go to http://*hostname:port*/dms/. For example:

```
http://adc1234:7001/dms/
```

See Also:

- Oracle Fusion Middleware Performance and Tuning Guide for details about instrumenting applications with DMS

Administrators can monitor performance for Oracle Access Manager 11g using the Monitoring command on the Actions menu under the System Configuration tab.

25.2 Monitoring Server Performance Metrics Using the Console

This section provides the following topics:

- Monitoring Server Instance Performance
- Reviewing Server Metrics

25.2.1 Monitoring Server Instance Performance

Users with valid OAM Administrator credentials can use the following procedure to display various performance metrics using the Oracle Access Manager Console.

Prerequisites

The server must be running.

To monitor performance using Oracle Access Manager Console

1. Go to the System Configuration tab.
2. Server Instance:
   a. From the Common Configuration section, locate and select the name of the server instance to monitor.
   b. From the Actions menu, click Monitor Menu.
   c. Click the desired subtab to view the results for the selected server instance:

      Server Processes Overview
      Session Operations
      Server Operations
      OAM Agents

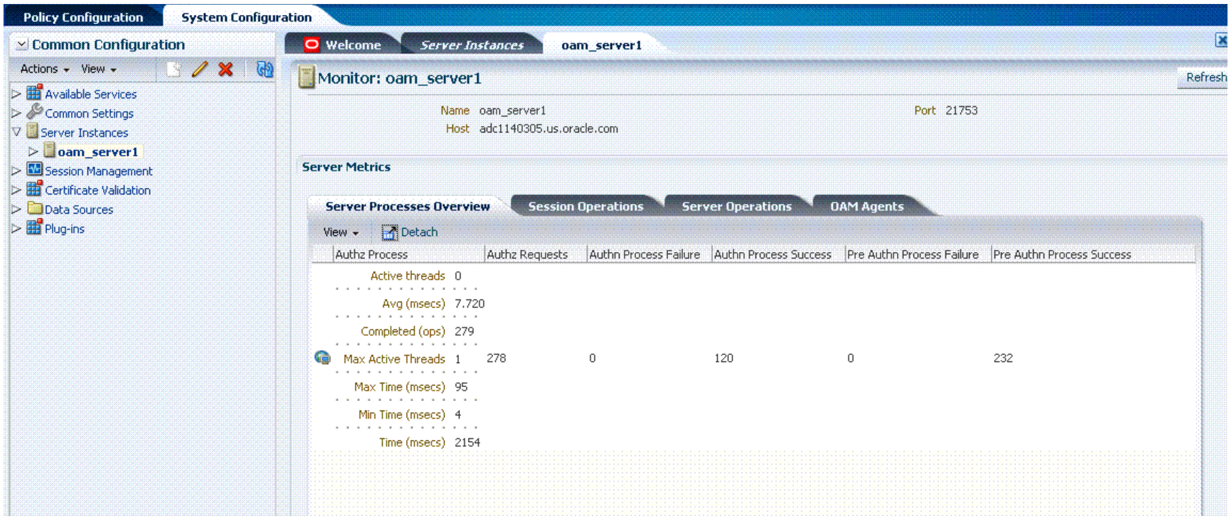
   d. Proceed to Reviewing Server Metrics.
3. OSSO Agent: On the instance page that opens, view the results.
   - Processes Overview
   - Operation Detail

25.2.2 Reviewing Server Metrics

This topic provides a look at the Server metrics available when you have a server instance selected in the navigation tree and you choose the Monitoring Menu command on the Actions menu under the System Configuration tab.

Figure 25–1 shows the Server Processes page.

Figure 25–1 Server Processes Overview Page



Monitor: oam_server1

Name oam_server1

Host adc1140305.us.oracle.com

Port 21753

Server Metrics

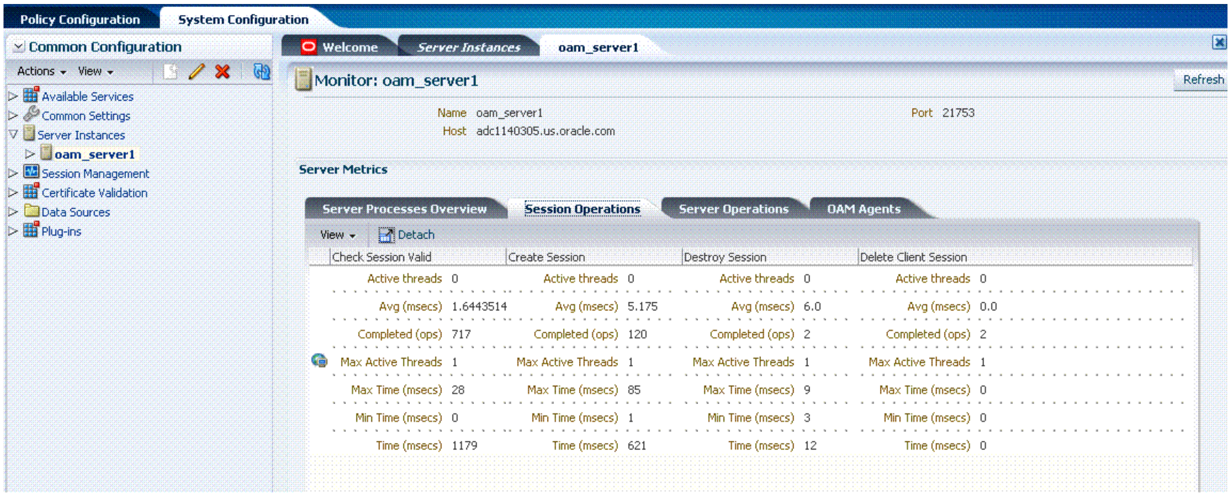
Authz Process		Authz Requests	Authn Process Failure	Authn Process Success	Pre Authn Process Failure	Pre Authn Process Success
Active threads	0					
Avg (msecs)	7.720					
Completed (ops)	279					
Max Active Threads	1	278	0	120	0	232
Max Time (msecs)	95					
Min Time (msecs)	4					
Time (msecs)	2154					

Server Processes Overview provides the following OAM Server performance metrics:

- Authorization Process
- Authorization Requests
- Authentication Process Failure
- Authentication Process Success
- Pre Authentication Process Failure
- Pre Authentication Process Success

Figure 25–2 shows the Session Operations Monitoring page.

Figure 25–2 Session Operations Monitoring Page



Monitor: oam_server1

Name oam_server1

Host adc1140305.us.oracle.com

Port 21753

Server Metrics

Check Session Valid		Create Session		Destroy Session		Delete Client Session	
Active threads	0	Active threads	0	Active threads	0	Active threads	0
Avg (msecs)	1.6443514	Avg (msecs)	5.175	Avg (msecs)	6.0	Avg (msecs)	0.0
Completed (ops)	717	Completed (ops)	120	Completed (ops)	2	Completed (ops)	2
Max Active Threads	1	Max Active Threads	1	Max Active Threads	1	Max Active Threads	1
Max Time (msecs)	28	Max Time (msecs)	85	Max Time (msecs)	9	Max Time (msecs)	0
Min Time (msecs)	0	Min Time (msecs)	1	Min Time (msecs)	3	Min Time (msecs)	0
Time (msecs)	1179	Time (msecs)	621	Time (msecs)	12	Time (msecs)	0

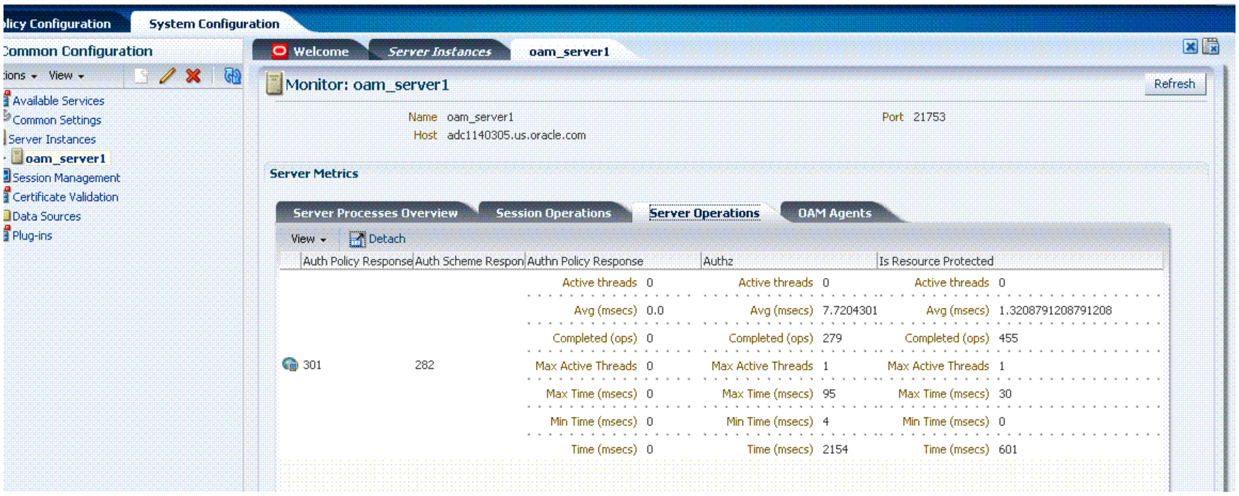
Session Operations performance metrics include:

- Check Session Valid

- Create Session
- Destroy Session
- Delete Client Session

Figure 25–3 shows the Server Operations Monitoring page.

Figure 25–3 Server Operations Monitoring Page



Monitor: oam_server1

Name oam_server1
Host adc1140305.us.oracle.com
Port 21753

Server Metrics

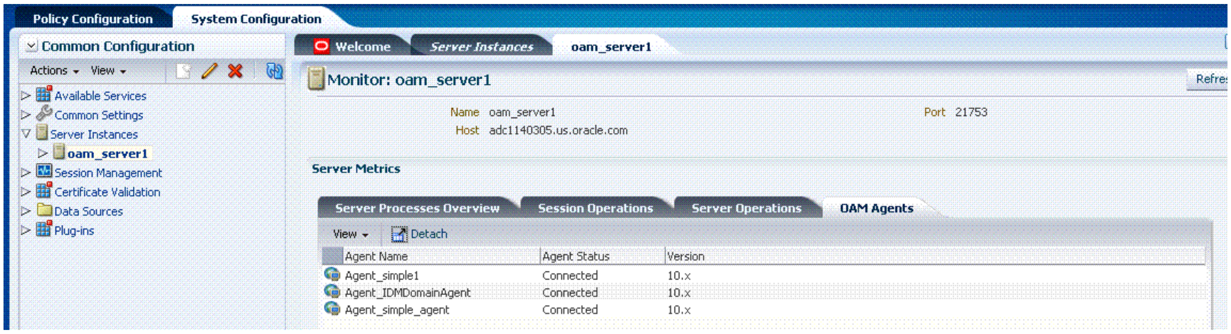
Auth Policy Response	Auth Scheme Respon	Authn Policy Response		Authz		Is Resource Protected	
		Active threads	0	Active threads	0	Active threads	0
		Avg (msecs)	0.0	Avg (msecs)	7.7204301	Avg (msecs)	1.3208791208791208
		Completed (ops)	0	Completed (ops)	279	Completed (ops)	455
301	282	Max Active Threads	0	Max Active Threads	1	Max Active Threads	1
		Max Time (msecs)	0	Max Time (msecs)	95	Max Time (msecs)	30
		Min Time (msecs)	0	Min Time (msecs)	4	Min Time (msecs)	0
		Time (msecs)	0	Time (msecs)	2154	Time (msecs)	601

Server Operations performance metrics include:

- Authentication Policy Response success
- Authentication Scheme Response success
- Authentication Policy Response
- Authorizations
- Statistics for Protected Resource

Figure 25–4 shows the Agents Monitoring page.

Figure 25–4 OAM Agents Monitoring Page



Monitor: oam_server1

Name oam_server1
Host adc1140305.us.oracle.com
Port 21753

Server Metrics

Agent Name	Agent Status	Version
Agent_simple1	Connected	10.x
Agent_IDMDomainAgent	Connected	10.x
Agent_simple_agent	Connected	10.x

OAM Agent performance metrics include:

- Name
- Status

- Version

25.3 Monitoring SSO Agent 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:

- Monitoring SSO Agent Performance Metrics
- Reviewing OAM Agent Metrics
- Reviewing OSSO Agent Metrics

25.3.1 Monitoring SSO Agent Performance Metrics

Users with valid OAM Administrator credentials can use the following procedure to display various SSO Agent performance metrics using the Oracle Access Manager Console.

Prerequisites

The server and agent must be running.

To monitor SSO Agent performance using Oracle Access Manager Console

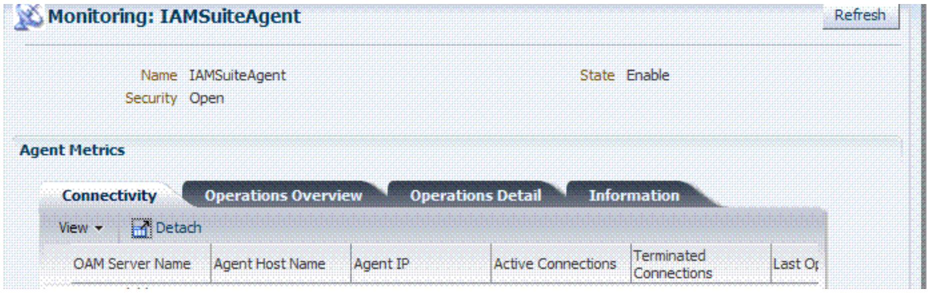
1. Go to the System Configuration tab, Access Manager Settings section.
2. Open the SSO Agents node, and then open the desired agent type node:
   - OAM Agents
   - OSSO Agents
3. Search for the desired agent to monitor using the controls for the open node.
4. In the Search Results table, highlight the row containing the agent you want to monitor.
5. Proceed as needed.
   - Reviewing OAM Agent Metrics
   - Reviewing OSSO Agent Metrics

25.3.2 Reviewing OAM Agent Metrics

Figure 25–5 shows the OAM Agent monitoring characteristics.

> **See Also:** "Tuning 10g and 11g Webgate Caches" on page 9-31

Figure 25–5 OAM Agent Monitoring Characteristics



Monitoring: IAMSuiteAgent

Name IAMSuiteAgent

Security Open

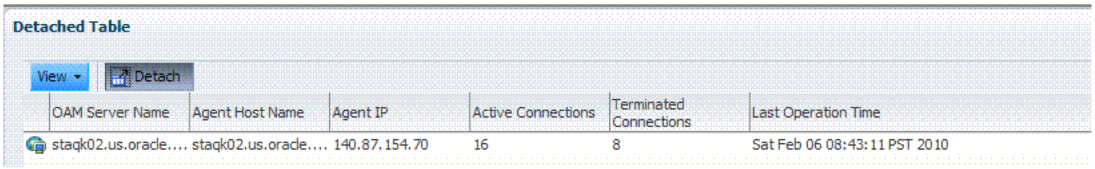
State Enable

Agent Metrics

Following figures illustrate detached tables:

- Figure 25–6, "Detached OAM 10g Agent Connection Table"
- Figure 25–7, "Detached OAM 10g Agent Operations Overview Table"
- Figure 25–8, "Detached OAM 10g Agent Operations Detail Table"
- Figure 25–9, "Detached OAM 10g Agent Information Table"

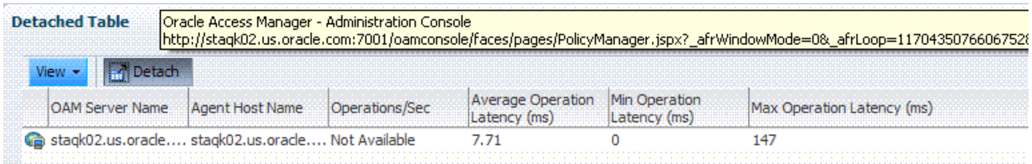
Figure 25–6 Detached OAM 10g Agent Connection Table



Detached Table

OAM Server Name	Agent Host Name	Agent IP	Active Connections	Terminated Connections	Last Operation Time
staqk02.us.oracle....	staqk02.us.oracle....	140.87.154.70	16	8	Sat Feb 06 08:43:11 PST 2010

Figure 25–7 Detached OAM 10g Agent Operations Overview Table

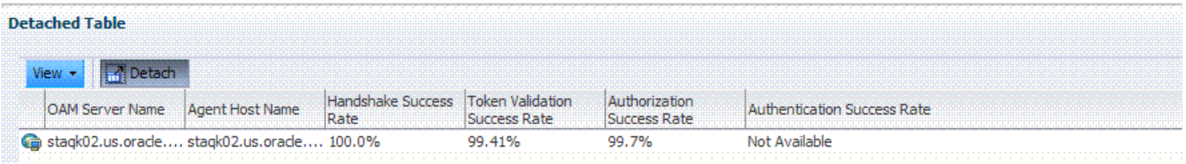


Detached Table

Oracle Access Manager - Administration Console
http://staqk02.us.oracle.com:7001/oamconsole/faces/pages/PolicyManager.jspx?_afrWindowMode=0&_afrLoop=11704350766067528

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

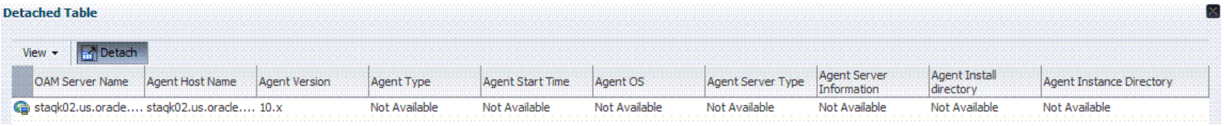
Figure 25–8 Detached OAM 10g Agent Operations Detail Table



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 25–9 Detached OAM 10g Agent Information Table



Detached Table

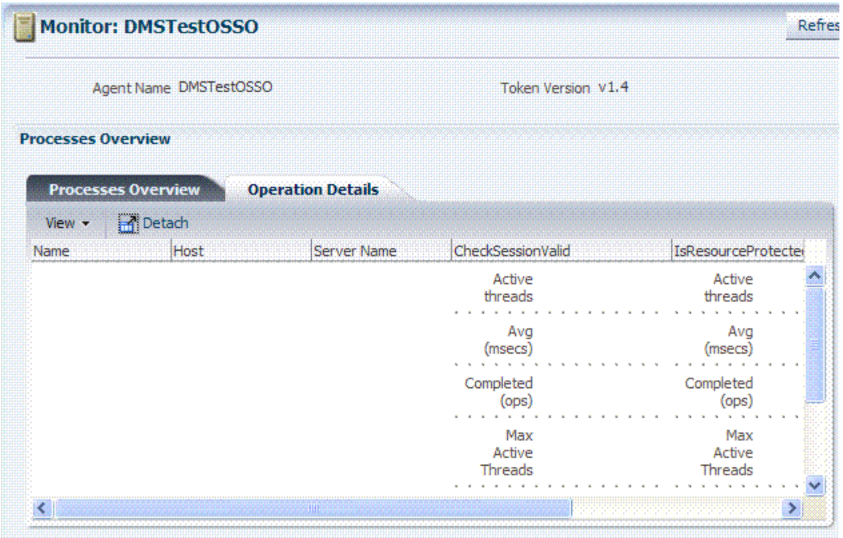
OAM Server Name	Agent Host Name	Agent Version	Agent Type	Agent Start Time	Agent OS	Agent Server Type	Agent Server Information	Agent Install directory	Agent Instance Directory
staqk02.us.oracle....	staqk02.us.oracle....	10.x	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available

25.3.3 Reviewing OSSO Agent Metrics

When you have an OSSO Agent selected in the navigation tree and choose Monitor Menu from the Actions menu, the following metrics pages are available:

- Figure 25–10, "OSSO 10g Agent Monitoring Page with Operation Details"
- Figure 25–11, "OSSO 10g Agent Monitoring Process Overview Table Detached"
- Figure 25–12, "OSSO 10g Agent Information Table Detached"

Figure 25–10 OSSO 10g Agent Monitoring Page with Operation Details



Monitor: DMSTestOSSO Refres

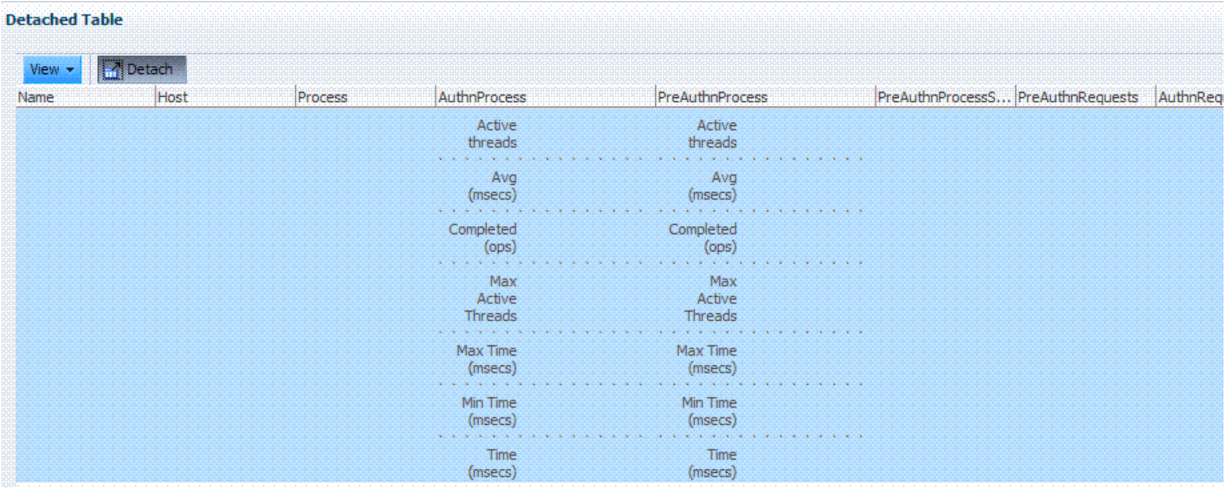
Agent Name DMSTestOSSO   Token Version v1.4

Processes Overview

Name	Host	Server Name	CheckSessionValid	IsResourceProtecte
			Active threads	Active threads
			Avg (msecs)	Avg (msecs)
			Completed (ops)	Completed (ops)
			Max Active Threads	Max Active Threads

Figure 25–11 illustrates the detached OSSO 10g Agent Monitoring Process Overview table.

Figure 25–11 OSSO 10g Agent Monitoring Process Overview Table Detached

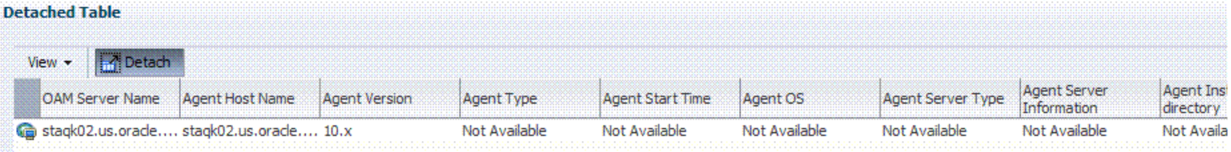


Detached Table

Name	Host	Process	AuthnProcess	PreAuthnProcess	PreAuthnProcessS...	PreAuthnRequests	AuthnReq
			Active threads	Active threads			
			Avg (msecs)	Avg (msecs)			
			Completed (ops)	Completed (ops)			
			Max Active Threads	Max Active Threads			
			Max Time (msecs)	Max Time (msecs)			
			Min Time (msecs)	Min Time (msecs)			
			Time (msecs)	Time (msecs)			

Figure 25–12 illustrates the detached OSSO 10g Agent Information table.

Figure 25–12 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

25.4 OXM Proxy Performance Tuning Parameters

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 Administrator can tune performance.

This section provides the following topics:

- About OAM Proxy Metrics
- OAM Proxy Server Tuning Parameters

25.4.1 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.

Table 25–1 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

25.4.2 OAM Proxy Server Tuning Parameters

Table 25–2 provides the tuning parameters for the OAM Proxy.

Table 25–2 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

26

Monitoring Performance and Logs with Fusion Middleware Control

This chapter describes how to monitor performance and log messages for Oracle Access Manager and Oracle Security Token Service using Oracle Fusion Middleware Control. This chapter focuses on general tasks that administrators can perform from Fusion Middleware Control, which does not replace details in the Oracle Fusion Middleware Administrator's Guide.

> **Note:** Unless explicitly stated, information in this chapter is the same whether you are using Oracle Access Manager alone or with Oracle Security Token Service.

This chapter includes the following topics:

- Prerequisites
- Introduction to Fusion Middleware Control
- Logging In to and Out of Fusion Middleware Control
- Displaying Menus and Pages in Fusion Middleware Control
- Viewing Performance in Fusion Middleware Control
- Managing Log Level Changes in Fusion Middleware Control
- Displaying MBeans in Fusion Middleware Control
- Displaying Farm Routing Topology in Fusion Middleware Control

26.1 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.

26.2 Introduction to Fusion Middleware Control

Within Fusion Middleware Control, information is updated dynamically during live sessions of Oracle Access Manager with Oracle Security Token Service (and other products).

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.

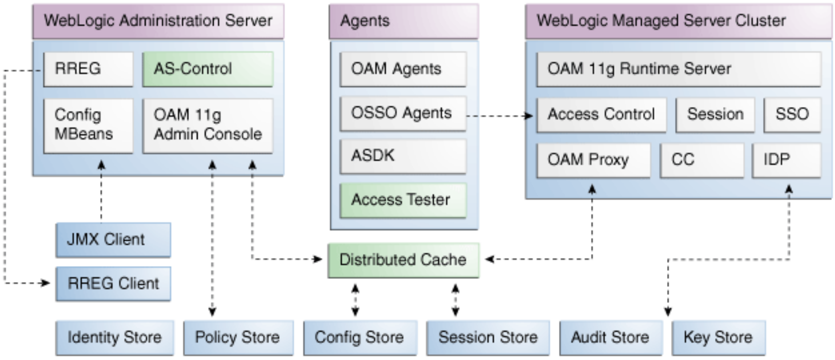
> **Note:** Enterprise Manager Grid Control is an independently licensed product that provides additional capabilities not found in Fusion Middleware Control (primarily, the ability to collect and maintain data for historical purposes and trending).

Oracle Access Manager 11g is deployed as a Java EE application in a WebLogic container. For high availability and failover, Oracle Access Manager with Oracle Security Token Service 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 Oracle Access Manager with Oracle Security Token Service, 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 26–1.

Figure 26–1 Fusion Middleware Control (AS-Control) Deployment Architecture



Using Fusion Middleware Control for Oracle Access Manager with Oracle Security Token Service targets is supported through the Oracle Dynamic Monitoring Systems instrumentation within Oracle Access Manager. This instrumentation is used to provide:

- Performance overview and drill down
- Log message searches and dynamic log level changes
- Routing topology overview
- Mbean browser
- Component- and cluster-level metrics for Oracle Access Manager with Oracle Security Token Service

26.3 Logging In to and Out of Fusion Middleware Control

This section provides the following topics:

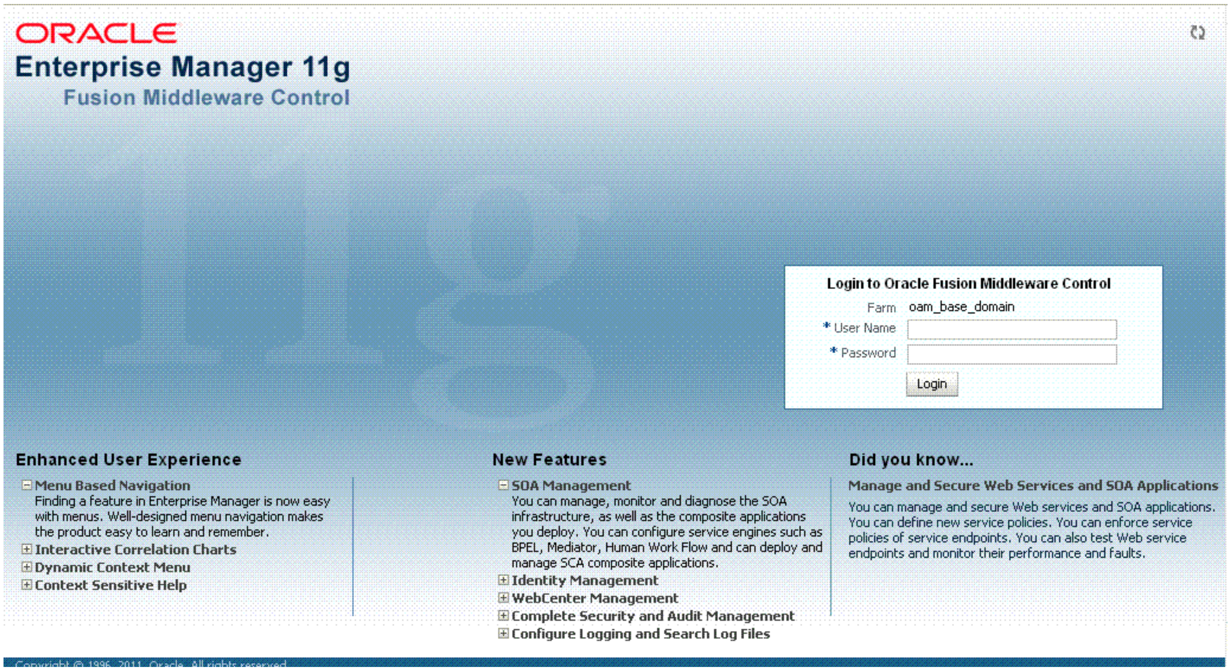
- About the Farm Page in Fusion Middleware Control

- Logging In To Fusion Middleware Control
- Logging Out of Fusion Middleware Control

26.3.1 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 26–2.

Figure 26–2 Fusion Middleware Control Login Page with Help Topics



ORACLE
Enterprise Manager 11g
Fusion Middleware Control

Login to Oracle Fusion Middleware Control

Farm oam_base_domain

* User Name

* Password

Login

Enhanced User Experience

⊟ Menu Based Navigation
Finding a feature in Enterprise Manager is now easy with menus. Well-designed menu navigation makes the product easy to learn and remember.
⊞ Interactive Correlation Charts
⊞ Dynamic Context Menu
⊞ Context Sensitive Help

New Features

⊟ SOA Management
You can manage, monitor and diagnose the SOA infrastructure, as well as the composite applications you deploy. You can configure service engines such as BPEL, Mediator, Human Work Flow and can deploy and manage SCA composite applications.
⊞ Identity Management
⊞ WebCenter Management
⊞ Complete Security and Audit Management
⊞ Configure Logging and Search Log Files

Did you know...

Manage and Secure Web Services and SOA Applications

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.

Copyright © 1996, 2011, Oracle. All rights reserved.

26.3.2 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 Oracle Access Manager 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.

26.3.3 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.

26.4 Displaying Menus and Pages in Fusion Middleware Control

This section provides the following topics for Oracle Access Manager with Oracle Security Token Service:

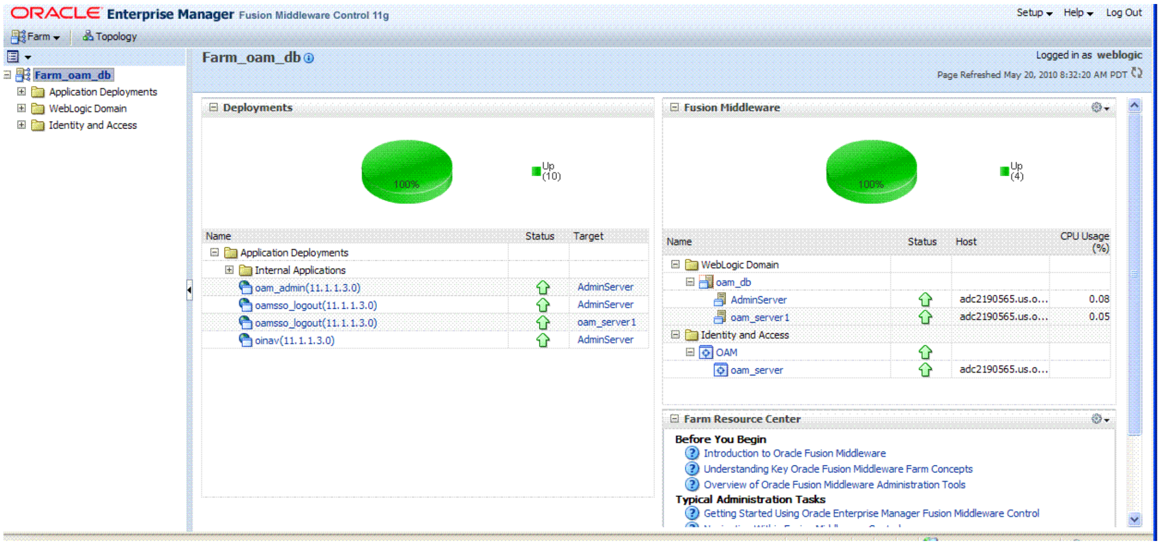
- About the Farm Page in Fusion Middleware Control
- About Context Menus and Pages in Fusion Middleware Control
- Displaying Context Menus and Target Details in Fusion Middleware Control

> **See Also:** *Oracle Fusion Middleware Administrator's Guide* for details about getting started using Fusion Middleware Control

26.4.1 About the Farm Page in Fusion Middleware Control

Figure 26–3 illustrates the Oracle Access Manager Farm page in Fusion Middleware Control. Each Farm page includes similar information. The Farm Resource Center provides immediate access to online information.

Figure 26–3 OAM Farm Page in Fusion Middleware Control



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

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

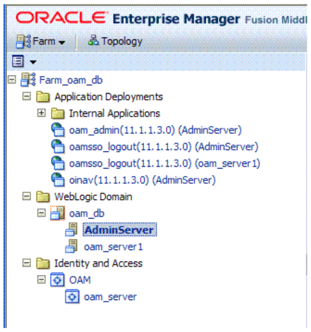
Sections on the Farm page are described in Table 26–1.

Table 26–1 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 26–4, 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 26–4 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 Menus and Pages in Fusion Middleware Control" on page 26-4

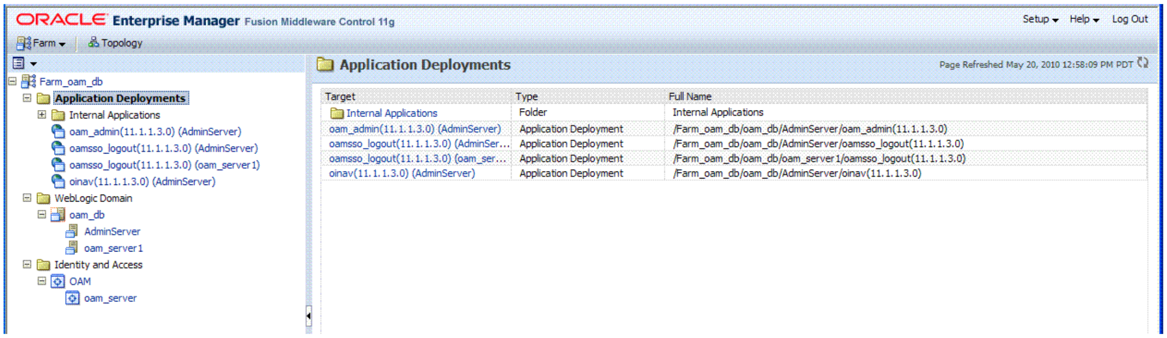
26.4.2 About Context Menus and Pages in Fusion Middleware Control

For Oracle Access Manager with Oracle Security Token Service, 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 Oracle Access Manager Cluster or individual Oracle Access Manager Server instances, which includes Oracle Security Token Service)

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 26–5 for Application Deployments.

Figure 26–5 Node Information Page in Fusion Middleware Control



- 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

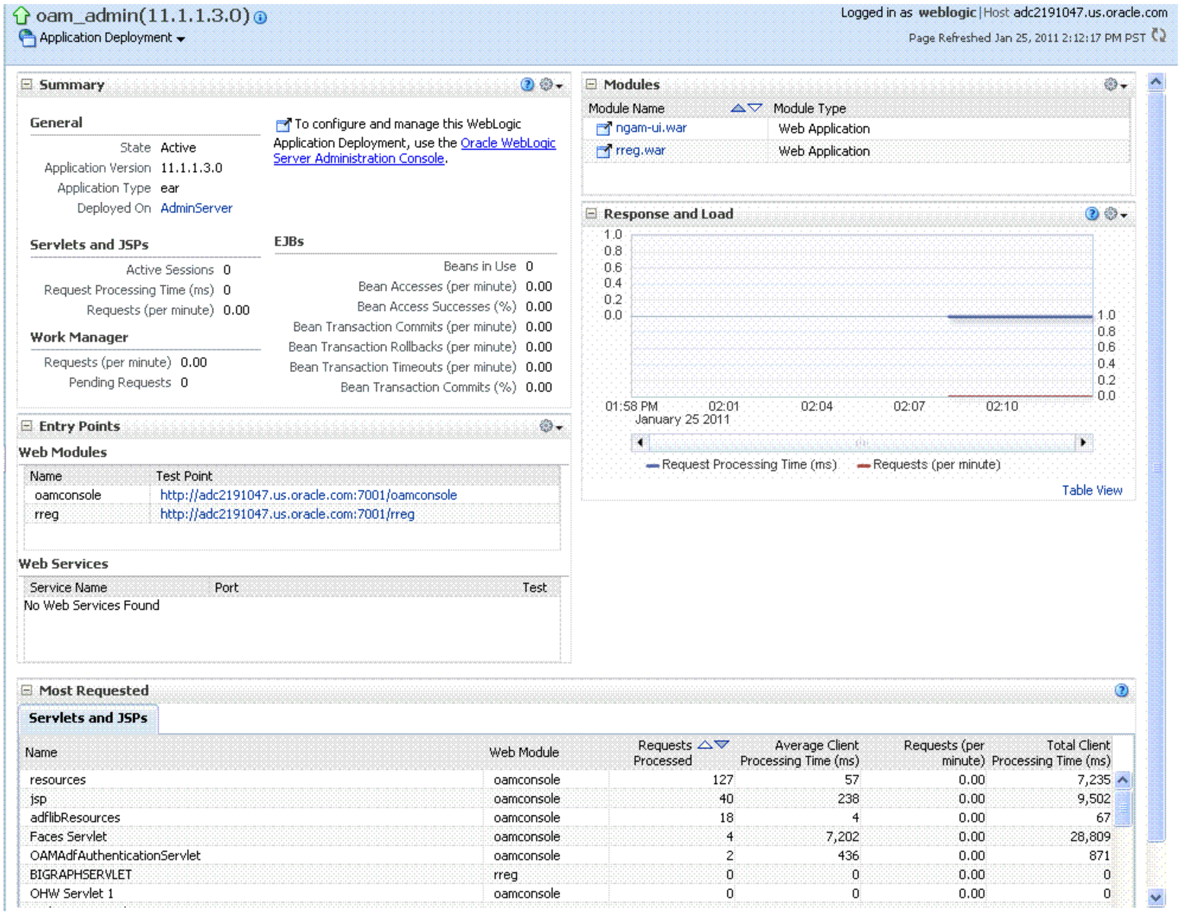
Application Deployments

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. 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, as shown in Figure 26–6.

Figure 26–6 Application Deployment Summary for the Selected Internal Application



oam_admin(11.1.1.3.0)

Application Deployment ▾

Logged in as weblogic | Host adc2191047.us.oracle.com

Page Refreshed Jan 25, 2011 2:12:17 PM PST

Summary

General

State	Active
Application Version	11.1.1.3.0
Application Type	ear
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

Modules

Module Name	Module Type
ngam-ui.war	Web Application
rreg.war	Web Application

Response and Load

Entry Points

Web Modules

Name	Test Point
oamconsole	http://adc2191047.us.oracle.com:7001/oamconsole
rreg	http://adc2191047.us.oracle.com:7001/rreg

Web Services

Service Name	Port	Test

No Web Services Found

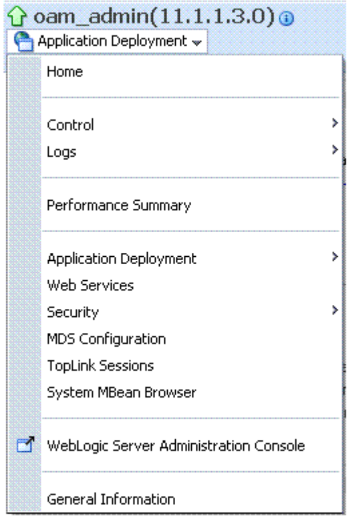
Most Requested

Servlets and JSPs

Name	Web Module	Requests Processed	Average Client Processing Time (ms)	Requests (per minute)	Total Client Processing Time (ms)
resources	oamconsole	127	57	0.00	7,235
jsp	oamconsole	40	238	0.00	9,502
adflibResources	oamconsole	18	4	0.00	67
Faces Servlet	oamconsole	4	7,202	0.00	28,809
OAMAdfAuthenticationServlet	oamconsole	2	436	0.00	871
BIGRAPHSERVLET	rreg	0	0	0.00	0
OHW Servlet 1	oamconsole	0	0	0.00	0

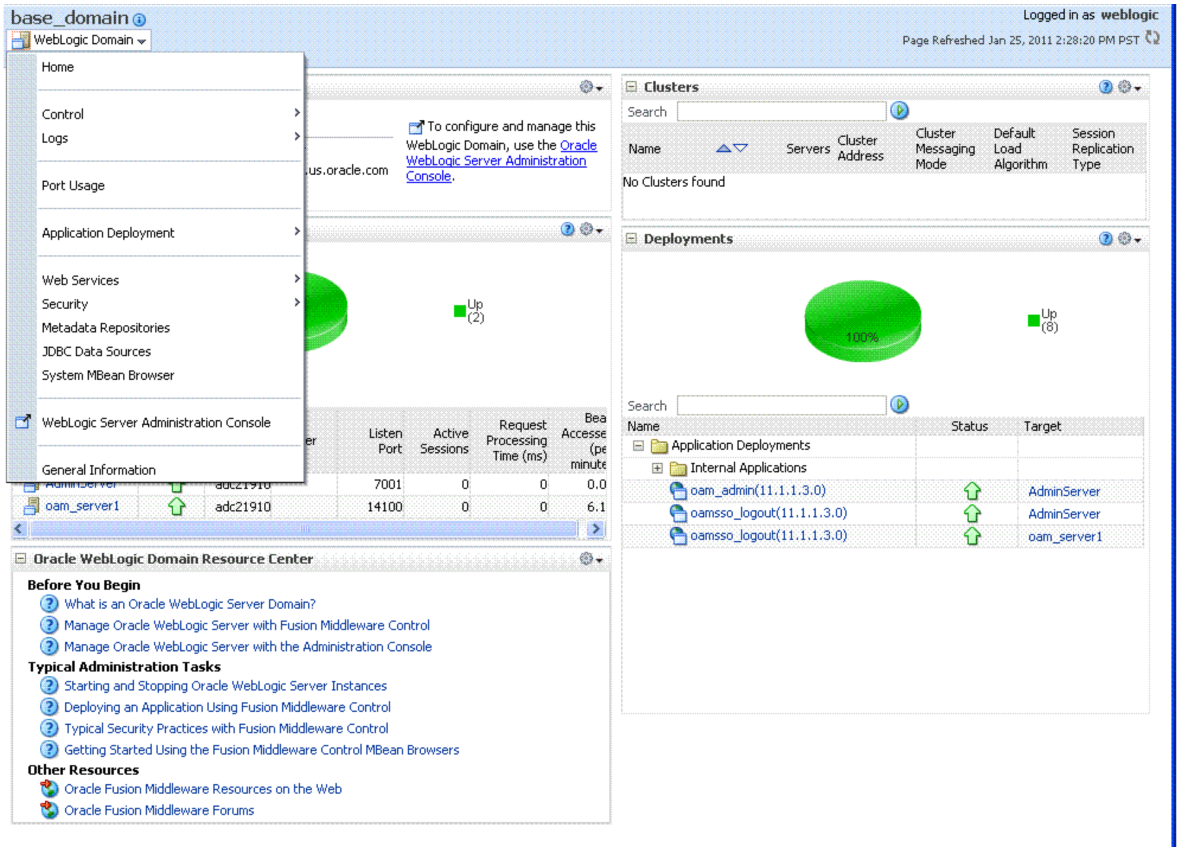
The Application Deployment menu is shown in Figure 26–7.

Figure 26–7 Application Deployment Menu



WebLogic Server domain: The WebLogic Server domain page is shown in Figure 26–8 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 26–8 WebLogic Server Domain Summary with Context Menu Exposed



Logged in as **weblogic**

Page Refreshed Jan 25, 2011 2:28:20 PM PST

us.oracle.com

To configure and manage this WebLogic Domain, use the Oracle WebLogic Server Administration Console.

er			Listen Port	Active Sessions	Request Processing Time (ms)	Bea Accesse (pe minute
AdminServer		adc21910	7001	0	0	0.0
oam_server1		adc21910	14100	0	0	6.1

Clusters

Search

Name	Servers	Cluster Address	Cluster Messaging Mode	Default Load Algorithm	Session Replication Type

No Clusters found

Deployments

Search

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

Oracle WebLogic Domain Resource Center

Before You Begin
- What is an Oracle WebLogic Server 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 Server Instances
- Deploying an Application Using Fusion Middleware Control
- Typical Security Practices with Fusion Middleware Control
- Getting Started Using the Fusion Middleware Control MBean Browsers

Other Resources
- Oracle Fusion Middleware Resources on the Web
- Oracle Fusion Middleware Forums

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 26–6.

For more information, see "Displaying Context Menus and Target Details in Fusion Middleware Control".

> **See Also:** "Viewing Performance in Fusion Middleware Control" on page 26-8 for information about the Identity and Access node and related pages.

26.4.3 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 for Oracle Access Manager with Oracle Security Token Service.

> **Note:** From the Farm Resource Center on the Oracle Access Manager 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 26-5

To display context menus and target information

1. Log in as described in "Logging In To Fusion Middleware Control" on page 26-3.
2. Expand the Farm containing Oracle Access Manager, 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 26–6 and Figure 26–7).
5. **Oracle Access Manager Cluster or Server Pages**: See "Viewing Performance in Fusion Middleware Control".

26.5 Viewing Performance in Fusion Middleware Control

Fusion Middleware Control provides administrators with:

- A cluster-wide view of performance for Oracle Access Manager with Oracle Security Token Service
- 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 sessions in a variety of formats. Table 26–2 summarizes the pages for selected nodes and target instances.

Table 26–2 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

Note: Oracle Security Token Service performance is included with both Oracle Access Manager Cluster and Oracle Access Manager Server pages.

This section provides the following topics:

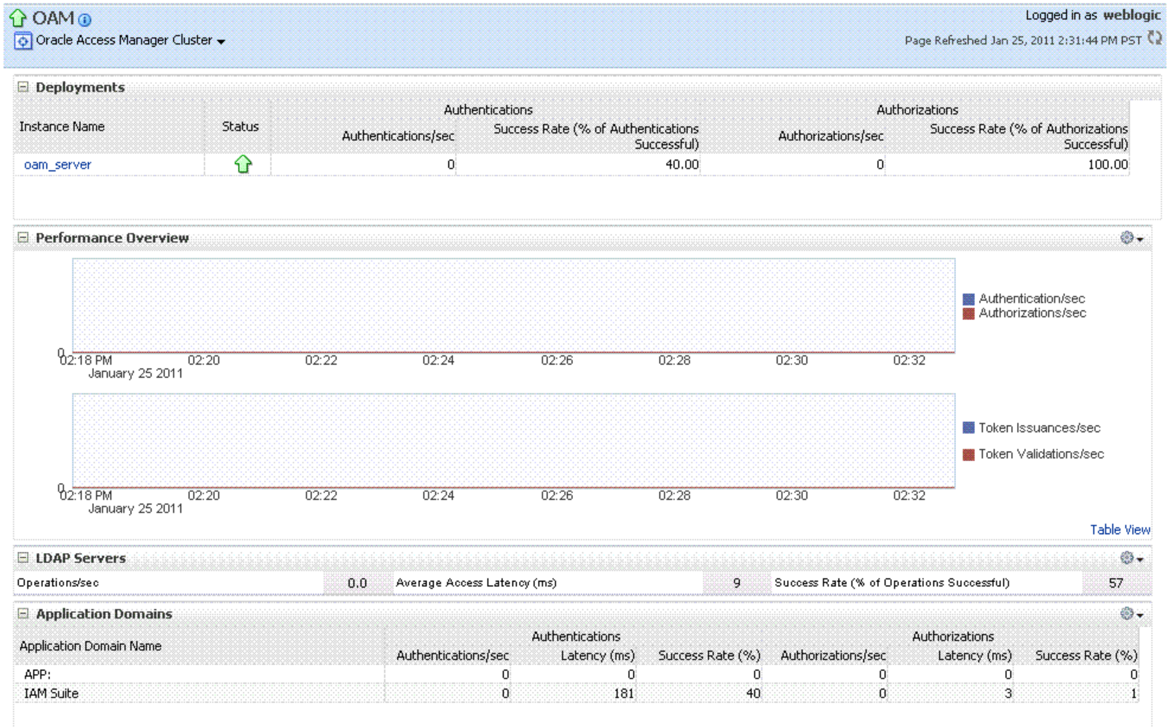
- About Performance Overview Pages in Fusion Middleware Control
- About the Metrics Palette and the Performance Summary Page
- Displaying Performance Metrics in Fusion Middleware Control
- Displaying Component-Specific Performance Details

26.5.1 About Performance Overview Pages in Fusion Middleware Control

The Fusion Middleware Control Performance Overview for Oracle Access Manager with Oracle Security Token Service can be used to reflect WebLogic cluster information down to specific performance metrics for individual Oracle Access Manager Cluster and Server targets.

Oracle Access Manager Cluster Page: The top node within Identity and Access leads to a page for the OAM Cluster Deployment, which includes a Performance Overview. For Figure 26–9, the Oracle Access Manager Cluster is selected in the navigation tree, beneath the Identity and Access node. Figure 26–9 illustrates the Oracle Access Manager Cluster Deployments and Performance Overview sections. This page includes a table for Token Issuance and Token Validations.

Figure 26–9 Oracle Access Manager Cluster Page



OAM

Oracle Access Manager Cluster ▾

Logged in as **weblogic**

Page Refreshed Jan 25, 2011 2:31:44 PM PST

Deployments

Instance Name	Status	Authentications		Authorizations	
		Authentications/sec	Success Rate (% of Authentications Successful)	Authorizations/sec	Success Rate (% of Authorizations Successful)
oam_server	⇧	0	40.00	0	100.00

Performance Overview

LDAP Servers

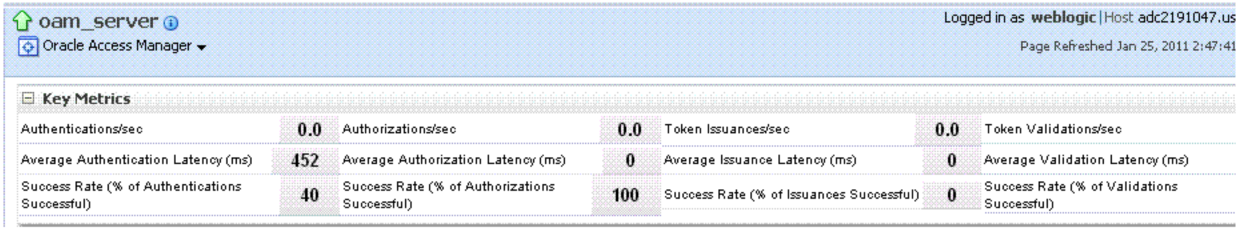
Operations/sec	0.0	Average Access Latency (ms)	9	Success Rate (% of Operations Successful)	57

Application Domains

Application Domain Name	Authentications			Authorizations		
	Authentications/sec	Latency (ms)	Success Rate (%)	Authorizations/sec	Latency (ms)	Success Rate (%)
APP:	0	0	0	0	0	0
IAM Suite	0	181	40	0	3	1

OAM Server Pages: Selecting an OAM Server target name from the navigation tree (or the open page), displays a Performance Overview for the target. At the top of the OAM Server page, a summary of Key Metrics for the server instances appears instead of the Oracle Access Manager Cluster Deployment section. Figure 26–10 illustrates the OAM Server instance Key Metrics, which include Token Issuance and Token Validations per second. The Token Validation success rate is included.

Figure 26–10 Key Metrics for Oracle Access Manager Server Pages



oam_server

Oracle Access Manager ▾

Logged in as **weblogic** | Host adc2191047.us

Page Refreshed Jan 25, 2011 2:47:41

Key Metrics

Authentications/sec	0.0	Authorizations/sec	0.0	Token Issuances/sec	0.0	Token Validations/sec
Average Authentication Latency (ms)	452	Average Authorization Latency (ms)	0	Average Issuance Latency (ms)	0	Average Validation Latency (ms)
Success Rate (% of Authentications Successful)	40	Success Rate (% of Authorizations Successful)	100	Success Rate (% of Issuances Successful)	0	Success Rate (% of Validations Successful)

Table 26–3 describes the elements of the Performance Overview for Oracle Access Manager Clusters and Oracle Access Manager Server instances in Fusion Middleware Control. There are only a few differences.

Table 26–3 Summary of Performance Overviews in Fusion Middleware Control

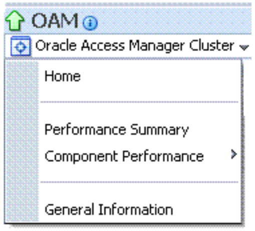
Section or Column Name	Description
Oracle Access Manager Cluster Menu	Dynamic context menus provide functions related to the selected target (also available when you right-click a target in the navigation tree). This menu is available for the selected Oracle Access Manager Cluster.  OAM Oracle Access Manager Cluster Home Performance Summary Component Performance General Information The Component Performance command enables you to choose between displaying Access Manager or Security Token Service metrics. See Also: "Access Manager Component Pages" and "Security Token Service Component Pages".
Deployments, OAM Cluster pages	This section appears only on OAM Cluster pages. It describes the status of each instance in the cluster. The following information is included: ■ Instance Name ■ Status ■ Authentications ■ Authorizations
Instance Name	This column includes the name of each OAM Server instance in the cluster. For example: *OAM_server_name*
Status	This column identifies the status of each OAM Server instance in the cluster with either a: ■ Green Up Arrow (running) ■ Red Down Arrow (not running)
Authentications	Authentications columns identify: ■ Authentications/sec: The number of authentications per second for each OAM Server instance in the cluster ■ Success Rate (% of Authentications Successful): A numeric value representing the percentage of successful authentications for each OAM Server instance in the cluster
Authorizations	This column identifies the number of authorizations per second for each OAM Server instance in the cluster. Authorizations columns identify: ■ Authorizations/sec: The number of authorizations per second for each OAM Server instance in the cluster ■ Success Rate (% of Authorizations Successful): A numeric value representing the percentage of successful authorizations for each OAM Server instance in the cluster

Table 26–3 (Cont.) Summary of Performance Overviews in Fusion Middleware Control

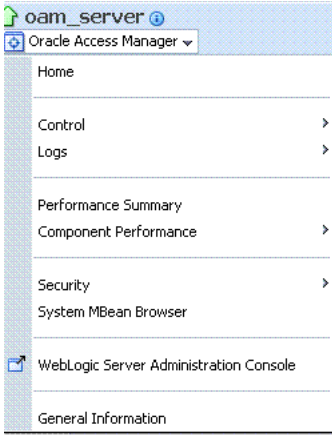
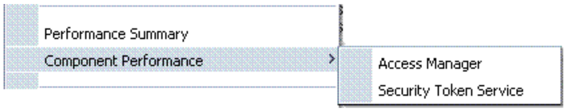
Section or Column Name	Description
Oracle Access Manager Server Instance Menu	Dynamic context menus provide functions related to the selected target (also available when you right-click a target in the navigation tree). This menu is available for the selected Oracle Access Manager server instance.  oam_server Oracle Access Manager Home Control Logs Performance Summary Component Performance Security System MBean Browser WebLogic Server Administration Console General Information The Component Performance command enables you to choose between displaying specific Access Manager or Security Token Service metrics. See Also: "Access Manager Component Pages" and "Security Token Service Component Pages".
Key Metrics, OAM Server Page	This table provides a summary of statistics for only the selected OAM Server instance. Key metrics include details for both Oracle Access Manager and Oracle Security Token Service: ■ Authentications/sec, Average Authentication Latency (ms), and Success ratio ■ Authorizations/sec, Average Authorization Latency (ms), and Success ratio ■ Token Issuances/sec, Average Issuance Latency (ms), and Success ratio ■ Token Validations/sec, Average Validation Latency (ms), and Success ratio
Performance Overview, OAM Cluster and OAM Server Pages	This section provides a graphic representations of Oracle Access Manager authentication and authorization operations and Oracle Security Token Service Token Issuance and Token Validation operations. Metrics in the Performance Overview are not configurable. The Metrics Palette is available for only the Performance Summary. Whether you have an OAM Cluster or OAM Server instance selected, the Performance Overview includes: ■ Authentications/sec and Authorizations/sec ■ Token Issuances/sec and Token Validations/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 named transactions that occured during the time period.

Table 26–3 (Cont.) Summary of Performance Overviews in Fusion Middleware Control

Section or Column Name	Description
Table View	Click the Table View link on the bottom-right side of the Performance Overview to display performance information in columns within a pop up window.
LDAP Servers, OAM Cluster and OAM Server Pages	This section is available when either an OAM Cluster or a single OAM Server instance is selected. It provides information for the default LDAP user identity store: ■ LDAP operations/sec ■ LDAP Latency (milliseconds) ■ LDAP Success Rate
Application Domains, OAM Cluster and OAM Server Pages	This section of the OAM Cluster and OAM Server pages provides information for all Application Domains that were used during authentication and authorization processing. Columns in this section provide the: ■ Application Domain Name: Each application domain that contains the authentication and authorization policies used for a request. ■ Authentications/sec, Authentications Latency (ms), Success Ratio (%) for each application domain ■ Authorizations/sec, Authorization Latency (ms), Success Ratio (%) for each application domain

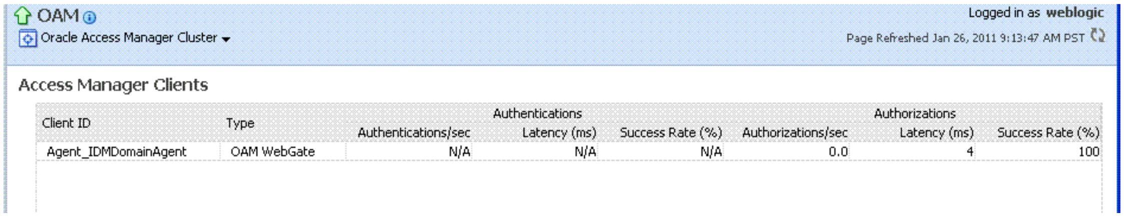
26.5.1.1 Access Manager Component Pages

The Component Performance command on both the Oracle Access Manager Cluster and Oracle Access Manager Server instance menus enables you to display Access Manager-specific metrics.



Oracle Access Manager Cluster component-specific metrics are aggregated across the cluster. illustrated in Figure 26–11. Details follow in Table 26–4.

Figure 26–11 Aggregated Access Manager Component Metrics for the Cluster

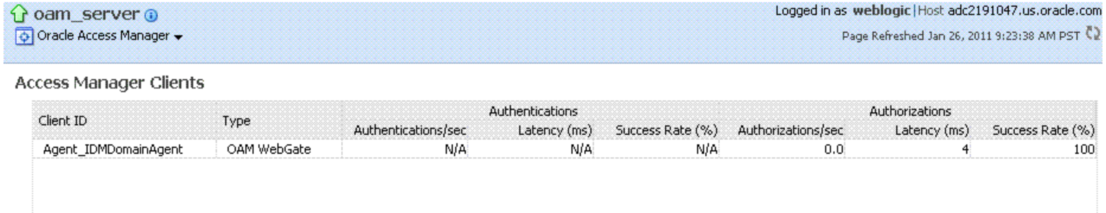

OAM
Oracle Access Manager Cluster
Logged in as weblogic
Page Refreshed Jan 26, 2011 9:13:47 AM PST

Access Manager Clients

Client ID	Type	Authentications			Authorizations		
		Authentications/sec	Latency (ms)	Success Rate (%)	Authorizations/sec	Latency (ms)	Success Rate (%)
Agent_IDMDomainAgent	OAM WebGate	N/A	N/A	N/A	0.0	4	100

Figure 26–12 illustrates the Access Manager component metrics for a single OAM Server instance.

Figure 26–12 Access Manager Component Metrics for a Single OAM Server Instance



oam_server
Oracle Access Manager
Logged in as weblogic | Host adc2191047.us.oracle.com
Page Refreshed Jan 26, 2011 9:23:38 AM PST

Access Manager Clients

Client ID	Type	Authentications			Authorizations		
		Authentications/sec	Latency (ms)	Success Rate (%)	Authorizations/sec	Latency (ms)	Success Rate (%)
Agent_IDMDomainAgent	OAM WebGate	N/A	N/A	N/A	0.0	4	100

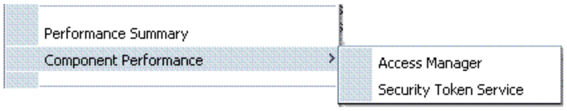
Table 26–4 describes the component-specific metrics for Oracle Access Manager.

Table 26–4 Access Manager Component Metrics

Access Manager Component Metrics	Description
Access Manager Clients	Based on your selection (Cluster or Server instance), this page provides information for all active Access Clients in a cluster (or for the active Access Clients of an individual OAM Server). Details include: ■ Client ID ■ Type ■ Authentications ■ Authorizations
Client ID	Displays the name of the Agent, as defined in the Agent registration in the Oracle Access Manager Console. For example: IAMSuiteAgent
Type	Displays the Agent. type For example: OAM Webgate
Authentications	Authentications columns identify: ■ Authentications/sec: The number of authentications per second for each OAM Server instance in the cluster ■ Latency (ms): The number of milliseconds the authentication was delayed ■ Success Rate (%): A numeric value representing the percentage of successful authentications for each OAM Server instance in the cluster
Authorizations	Authorizations columns identify: ■ Authorizations/sec: The number of authorizations per second for each OAM Server instance in the cluster ■ Latency (ms): The number of milliseconds the authorization was delayed ■ Success Rate (%): A numeric value representing the percentage of successful authorizations for each OAM Server instance in the cluster

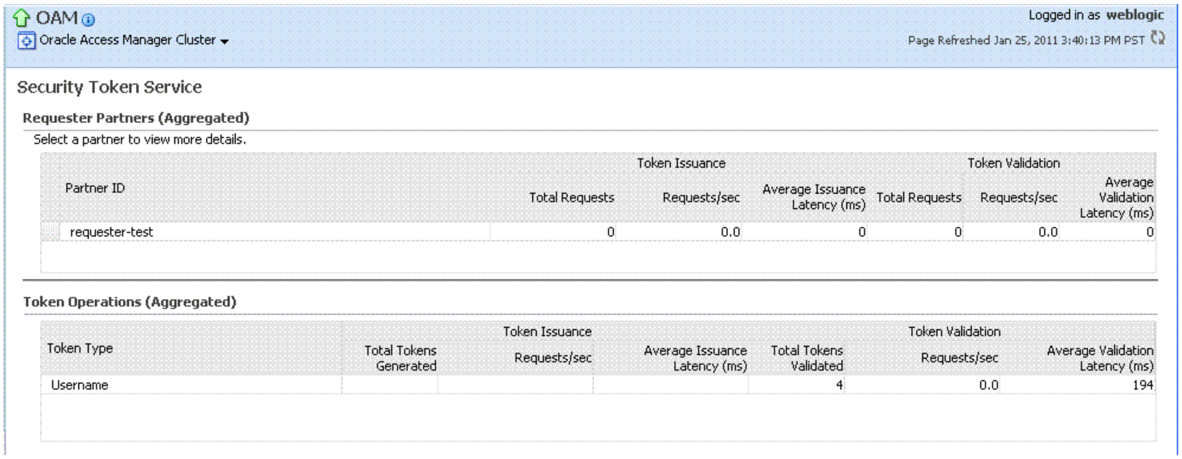
26.5.1.2 Security Token Service Component Pages

The Component Performance command on both the Oracle Access Manager Cluster and Oracle Access Manager Server instance menus enables you to display Security Token Service (STS) component-specific metrics.



Component-specific metrics are aggregated for the Oracle Access Manager Cluster, as illustrated in Figure 26–11.

Figure 26–13 Aggregated STS Component Metrics for the Cluster


OAM
Oracle Access Manager Cluster
Logged in as weblogic
Page Refreshed Jan 25, 2011 3:40:13 PM PST

Security Token Service

Requester Partners (Aggregated)

Select a partner to view more details.

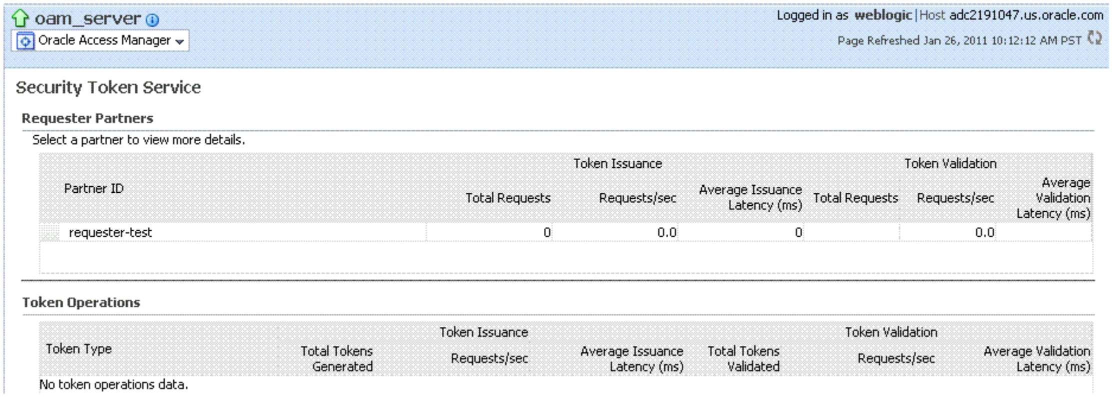
Partner ID	Token Issuance			Token Validation		
	Total Requests	Requests/sec	Average Issuance Latency (ms)	Total Requests	Requests/sec	Average Validation Latency (ms)
requester-test	0	0.0	0	0	0.0	0

Token Operations (Aggregated)

Token Type	Token Issuance			Token Validation		
	Total Tokens Generated	Requests/sec	Average Issuance Latency (ms)	Total Tokens Validated	Requests/sec	Average Validation Latency (ms)
Username				4	0.0	194

For each individual server instance, STS component-specific metrics are also available, as illustrated in Figure 26–11.

Figure 26–14 STS Component Metrics for an Individual OAM Server Instance


oam_server
Oracle Access Manager
Logged in as weblogic | Host adc2191047.us.oracle.com
Page Refreshed Jan 26, 2011 10:12:12 AM PST

Security Token Service

Requester Partners

Select a partner to view more details.

Partner ID	Token Issuance			Token Validation		
	Total Requests	Requests/sec	Average Issuance Latency (ms)	Total Requests	Requests/sec	Average Validation Latency (ms)
requester-test	0	0.0	0		0.0	

Token Operations

Token Type	Token Issuance			Token Validation		
	Total Tokens Generated	Requests/sec	Average Issuance Latency (ms)	Total Tokens Validated	Requests/sec	Average Validation Latency (ms)

No token operations data.

Table 26–5 introduces the STS component specific metrics.

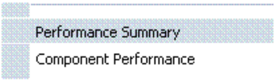
Table 26–5 STS Component-Specific Metrics

Security Token Service Component Metrics	Description
Requestor Partners	Statistics summary for either the selected OAM Server instance (or an aggregated summary for the Cluster): ■ Partner ID ■ Token Issuances ■ Token Validations Selecting a Requestor Partner ID reveals Relying Party Details with specific information for only the named partner.
Token Operations	Metrics for STS Token Operations include: ■ Token Type ■ Token Issuances: Total Requests, Requests per second, Average Issuance Latency (ms) ■ Token Validations: Total Requests, Requests per second, Average Issuance Latency (ms)

26.5.2 About the Metrics Palette and the Performance Summary Page

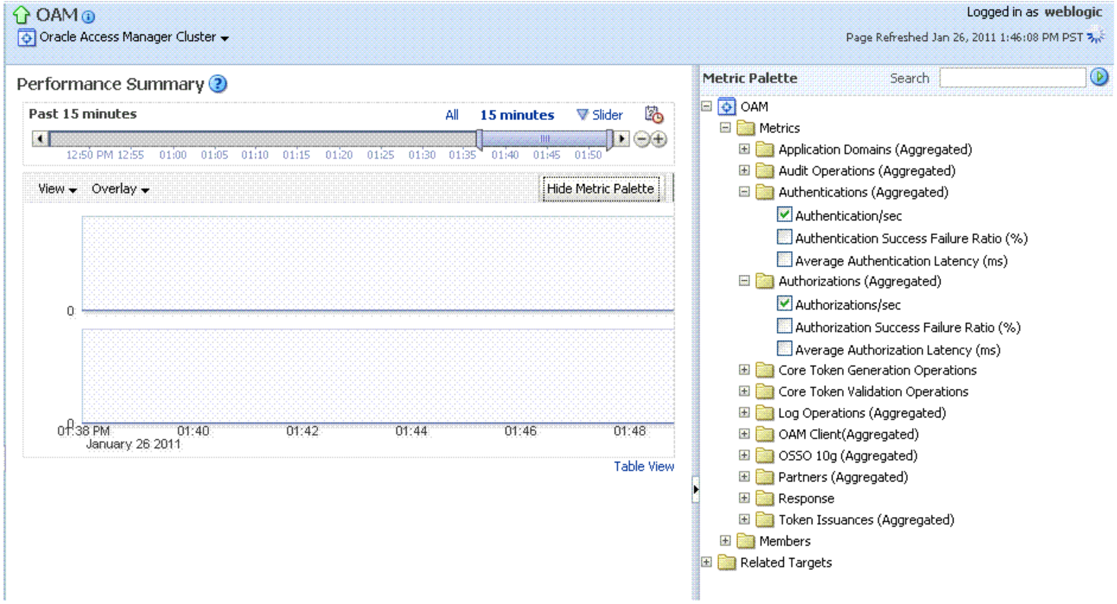
The Performance Summary command on the Oracle Access Manager Cluster or Server menu displays metrics charts for the selected target.

Figure 26–15 Performance Summary Command



On the Performance Summary page, a chart is displayed for each selected metric. An OAM Server Performance Summary page. Figure 26–16 shows the Performance Summary page with an open Metric Palette from which you can choose metrics 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 26–16 Performance Summary Page with Metric Palette



- OAM
  - Metrics
    - Application Domains (Aggregated)
    - Audit Operations (Aggregated)
    - Authentications (Aggregated)
      - ☑ Authentication/sec
      - □ Authentication Success Failure Ratio (%)
      - □ Average Authentication Latency (ms)
    - Authorizations (Aggregated)
      - ☑ Authorizations/sec
      - □ Authorization Success Failure Ratio (%)
      - □ Average Authorization Latency (ms)
    - Core Token Generation Operations
    - Core Token Validation Operations
    - Log Operations (Aggregated)
    - OAM Client(Aggregated)
    - OSSO 10g (Aggregated)
    - Partners (Aggregated)
    - Response
    - Token Issuances (Aggregated)
  - Members
- Related Targets

Table 26–6 describes the status and controls available on the Performance Summary page.

Table 26–6 Status and Controls on Performance Summary Pages

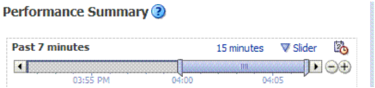
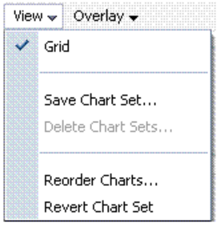
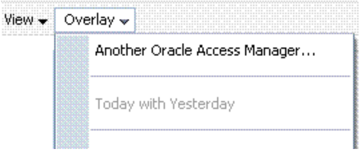
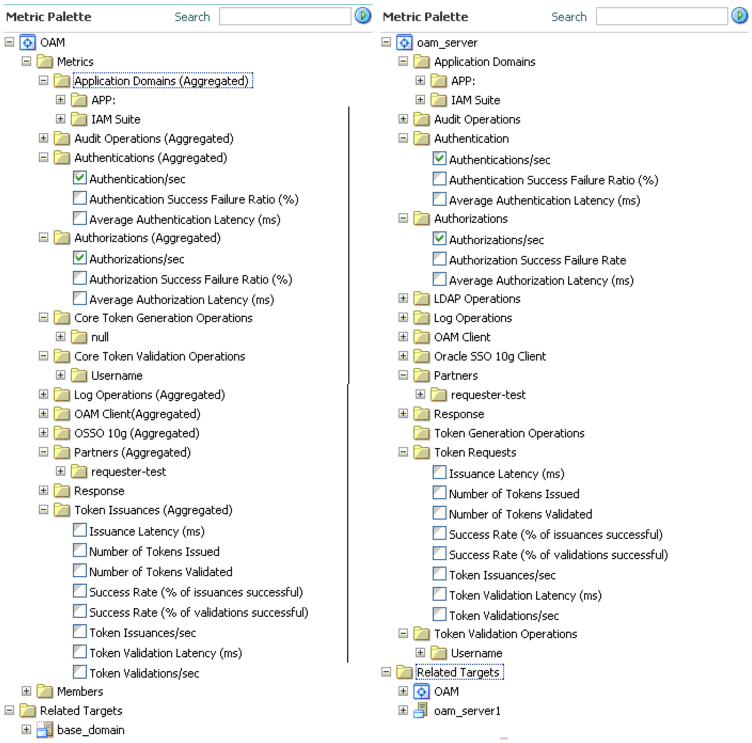
Status or Control	Description
Past *n* minutes	Status is based on the specified time period, which can be adjusted using the slider.
All	
n Minutes	The specified time period, which can be adjusted using the slider.
Slider	The tool you use to adjust the time period.  Performance Summary Past 7 minutes 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 26–6 (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.  View Overlay ✓ Grid Save Chart Set... Delete Chart Sets... Reorder Charts... Revert Chart Set
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.  View Overlay Another Oracle Access Manager... Today with Yesterday
Metric Palette	A listing from which you can select performance metrics to chart. Items unique to Oracle Access Manager with Oracle Security Token Service are shown here. Left: Metric Palette for the Cluster Right: Metric Palette for a Single OAM Server  Metric Palette Search OAM Metrics Application Domains (Aggregated) APP: IAM Suite Audit Operations (Aggregated) Authentications (Aggregated) ☑ Authentication/sec □ Authentication Success Failure Ratio (%) □ Average Authentication Latency (ms) Authorizations (Aggregated) ☑ Authorizations/sec □ Authorization Success Failure Ratio (%) □ Average Authorization Latency (ms) Core Token Generation Operations null Core Token Validation Operations Username Log Operations (Aggregated) OAM Client(Aggregated) OSSO 10g (Aggregated) Partners (Aggregated) requester-test Response Token Issuances (Aggregated) □ Issuance Latency (ms) □ Number of Tokens Issued □ Number of Tokens Validated □ Success Rate (% of issuances successful) □ Success Rate (% of validations successful) □ Token Issuances/sec □ Token Validation Latency (ms) □ Token Validations/sec Members Related Targets base_domain Metric Palette Search oam_server Application Domains APP: IAM Suite Audit Operations Authentication ☑ Authentications/sec □ Authentication Success Failure Ratio (%) □ Average Authentication Latency (ms) Authorizations ☑ Authorizations/sec □ Authorization Success Failure Rate □ Average Authorization Latency (ms) LDAP Operations Log Operations OAM Client Oracle SSO 10g Client Partners requester-test Response Token Generation Operations Token Requests □ Issuance Latency (ms) □ Number of Tokens Issued □ Number of Tokens Validated □ Success Rate (% of issuances successful) □ Success Rate (% of validations successful) □ Token Issuances/sec □ Token Validation Latency (ms) □ Token Validations/sec Token Validation Operations Username Related Targets OAM oam_server1

26.5.3 Displaying 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 Performance Summary. for Oracle Access Manager with Oracle Security Token Service.

> **See Also:**
>
> - "About Performance Overview Pages in Fusion Middleware Control"
> - "About the Metrics Palette and the Performance Summary Page"

To add or change metrics displayed in the Performance Summary

1. Log in as described in "Logging In To Fusion Middleware Control" on page 26-3.
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.
3. **Performance Summary**:
   a. Select a target (Step 1).
   b. From the context menu, select Performance Summary.
   c. Review the Summary Page.
4. **Changing Metrics**:
   a. From the Performance Summary page (Step 2), click the **Show Metrics Palette** button.
   b. From the Metrics Palette, expand nodes and check (or clear) boxes to add (or remove) metrics from the summary.
   c. Review the updated the Summary page.
   d. Click **Hide Metrics Palette** when you finish.
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, Oracle Access Manager**:
   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. **Adding an Overlay, Today with Yesterday**:

   a. From the Overlay menu on the Performance Summary page, click **Today with Yesterday**.

   b. When finished viewing the overlay, click **Remove Overlay** from the Overlay menu.

8. **Testing**:

   a. Using the Access Tester, perform several authentication and authorization tests (see Chapter 14).

   b. In Fusion Middleware Control, check performance metrics.

26.5.4 Displaying Component-Specific Performance Details

Fusion Middleware Control administrators can use the following procedure to view and compare component-specific performance data for either Oracle Access Manager or Oracle Security Token Service.

See Also:

- "Access Manager Component Pages"
- "Security Token Service Component Pages"

To display component-specific performance details

1. Log in as described in "Logging In To Fusion Middleware Control" on page 26-3.

2. Expand the desired node and select a target. For example:

   Identity and Access
     oam_server

3. From the context menu, select **Component Performance**.

4. Choose **Access Manager** (or **Security Token Service**).

5. **STS Partner ID**: Choose a Partner ID in the Security Token Service results table for more details, if needed.

6. **Component Performance**:

   a. From the context menu, select Component Performance.

   b. Choose either Access Manager or Security Token Service.

   c. Choose an item in the results table to get more details, if available.

7. **Testing**:

   a. Using the Access Tester, perform several authentication and authorization tests (see Chapter 14).

   b. In Fusion Middleware Control, check performance metrics.

26.6 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 22.

Topics in this section include:

- About Dynamic Log Level Changes
- Setting Log Levels Dynamically Using Fusion Middleware Control

26.6.1 About Dynamic Log Level Changes

Using Fusion Middleware Control, administrators can change log levels dynamically for Oracle Access Manager with Oracle Security Token Service.

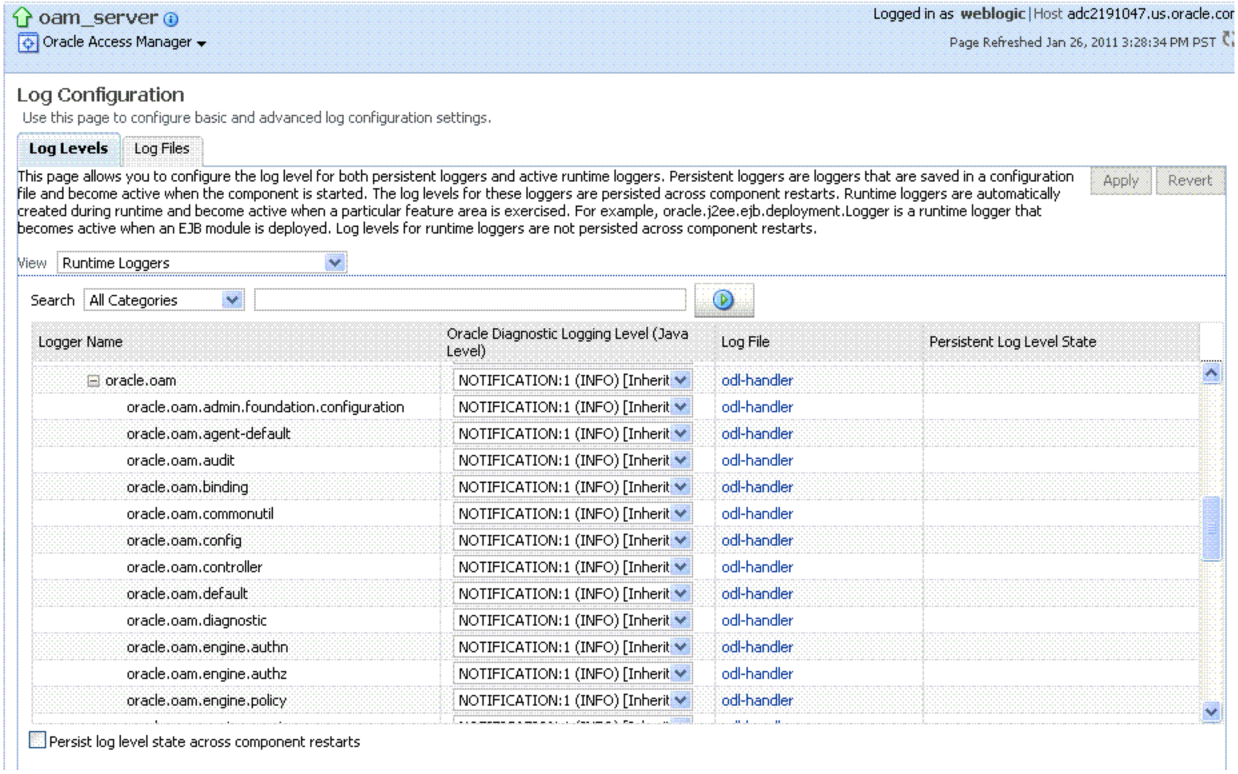
Table 26–7 outlines log availability and functions in Fusion Middleware Control.

Table 26–7 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 26–17 shows the Log Levels configuration page in Fusion Middleware Control. Notice that Runtime Loggers is the selected View and oracle.oam logger names are currently displayed. With Oracle Security Token Service there is only one logger that affects the log levels for Oracle Security Token Service: `oracle.security.fed`.

Figure 26–17 Oracle Access Manager Log Levels on the Log Configuration Tab



oam_server
Oracle Access Manager
Logged in as weblogic | Host adc2191047.us.oracle.cor
Page Refreshed Jan 26, 2011 3:28:34 PM PST

Log Configuration

Use this page to configure basic and advanced log configuration settings.

Log Levels
Log Files
Apply
Revert

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.

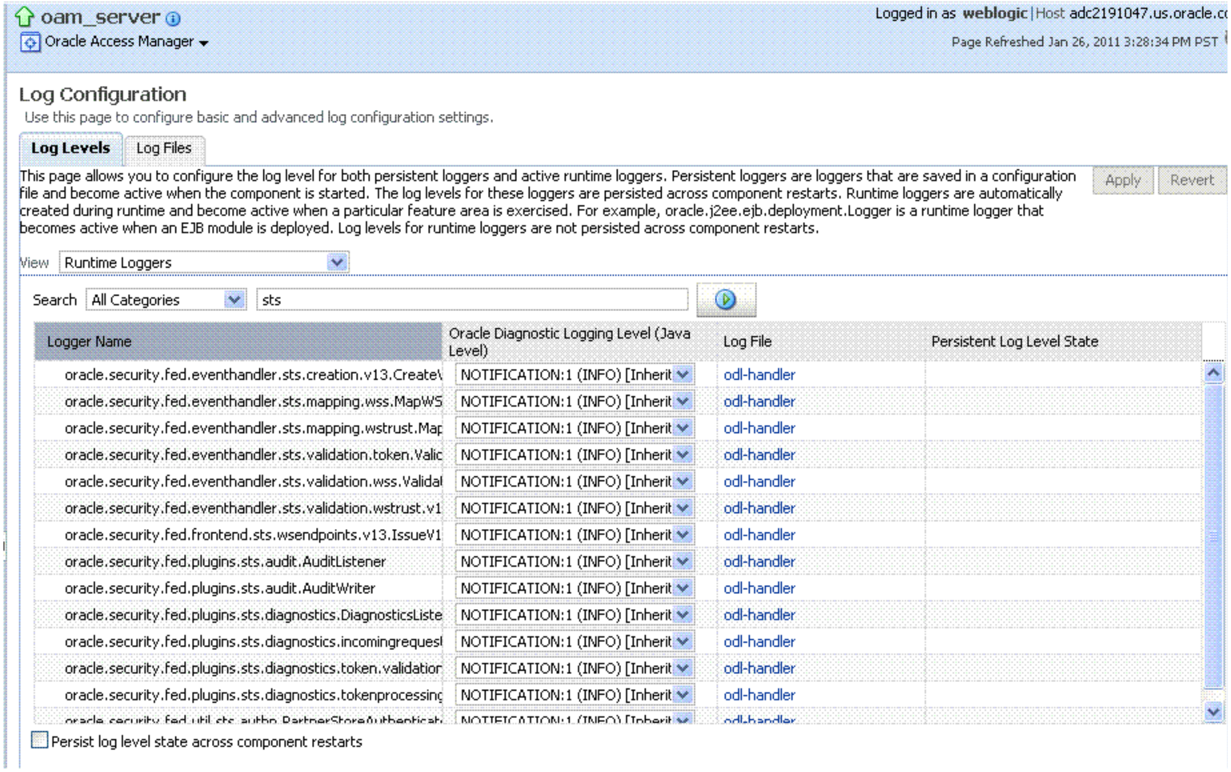
View: Runtime Loggers

Search: All Categories

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
⊟ oracle.oam	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.admin.foundation.configuration	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.agent-default	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.audit	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.binding	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.commonutil	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.config	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.controller	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.default	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.diagnostic	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.engine.authn	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.engine.authz	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.oam.engine.policy	NOTIFICATION:1 (INFO) [Inherit	odl-handler	

□ Persist log level state across component restarts

Figure 26–18 Log Levels for Oracle Security Token Service



oam_server

Oracle Access Manager

Logged in as weblogic | Host adc2191047.us.oracle.c

Page Refreshed Jan 26, 2011 3:28:34 PM PST

Log Configuration

Use this page to configure basic and advanced log configuration settings.

Log Levels | Log Files

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.

Apply | Revert

View: Runtime Loggers

Search: All Categories | sts

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
oracle.security.fed.eventhandler.sts.creation.v13.Create\	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.eventhandler.sts.mapping.wss.MapWS	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.eventhandler.sts.mapping.wstrust.Map	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.eventhandler.sts.validation.token.Valic	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.eventhandler.sts.validation.wss.Validal	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.eventhandler.sts.validation.wstrust.v1	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.frontend.sts.wsendpoints.v13.IssueV1	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.plugins.sts.audit.AuditListener	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.plugins.sts.audit.AuditWriter	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.plugins.sts.diagnostics.DiagnosticsListe	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.plugins.sts.diagnostics.incomingreques	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.plugins.sts.diagnostics.token.validatior	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.plugins.sts.diagnostics.tokenprocessinc	NOTIFICATION:1 (INFO) [Inherit	odl-handler	
oracle.security.fed.util.sts.authn.PartnerStoreAuthenticat	NOTIFICATION:1 (INFO) [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.

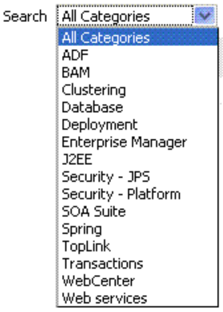
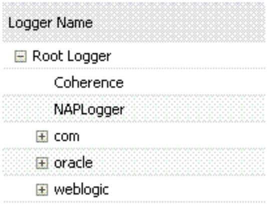
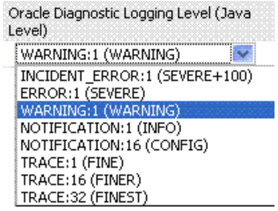
  For example, oracle.j2ee.ejb.deployment.Logger is a runtime logger that becomes active when an EJB module is deployed. Log levels for runtime loggers are not persisted across component restarts.

Table 26–8 explains the configuration status and options for log levels.

Table 26–8 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

Table 26–8 (Cont.) Log Levels Tab on Log Configuration Page

Element	Description
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
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 Log File Configuration from Fusion Middleware Control" on page 26-25.
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.

26.6.2 Setting Log Levels Dynamically Using Fusion Middleware Control

Fusion Middleware Control administrators can use the following procedure to set the log level dynamically for Oracle Access Manager with Oracle Security Token Service.

See Also: "About Dynamic Log Level Changes" on page 26-21

Note: Alternatively, administrators can set logger levels using custom WebLogic Scripting Tool (WLST) commands, as described in Chapter 22.

To configure logging levels dynamically in Fusion Middleware Control

1. Log in as described in "Logging In To Fusion Middleware Control" on page 26-3.
2. Expand the desired node, and select a target. For example:

   Identity and Access
   oam_server

3. From the Oracle Access Manager context menu, select Logs and then choose Log Configuration.
4. From the Log Levels tab, View list, choose the loggers to display. For example: **Runtime Loggers**.
5. From the Search list, choose a category, enter your search criteria, and click the search button. For example: **All Categories sts**.
6. In the results table, expand nodes to reveal information as needed.
7. In the results table, choose log levels for your environment, then click Apply (or Revert).
8. Proceed to "Managing Log File Configuration from Fusion Middleware Control"

26.7 Managing Log File Configuration from Fusion Middleware Control

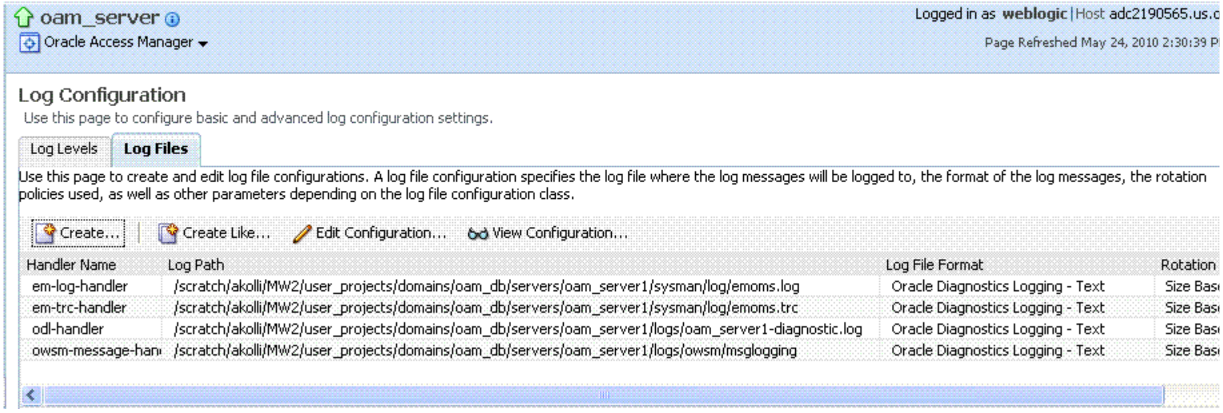
This section provides the following information for Oracle Access Manager with Oracle Security Token Service:

- About Log File Configuration
- Managing Log File Configuration by Using Fusion Middleware Control

26.7.1 About Log File Configuration

Figure 26–8 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 26–19 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 26–9 describes the log files configuration parameters for Oracle Access Manager with Oracle Security Token Service.

Table 26–9 Log Files Elements

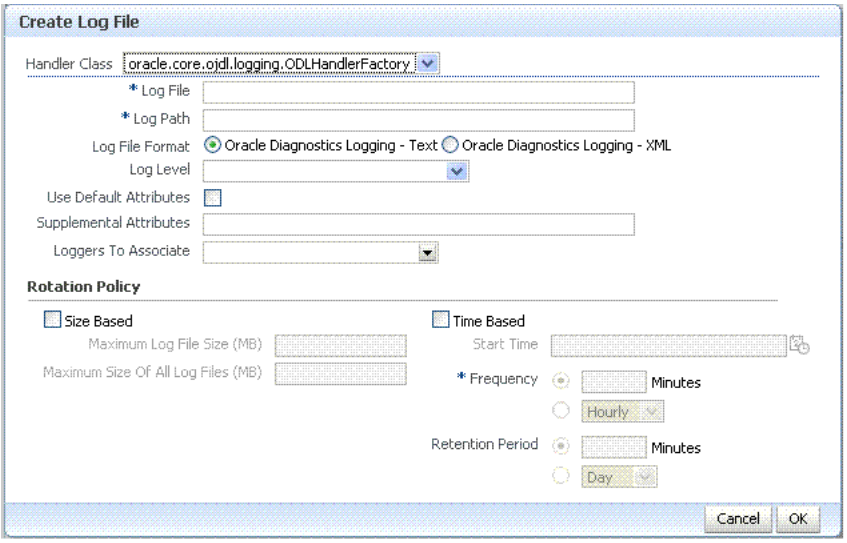
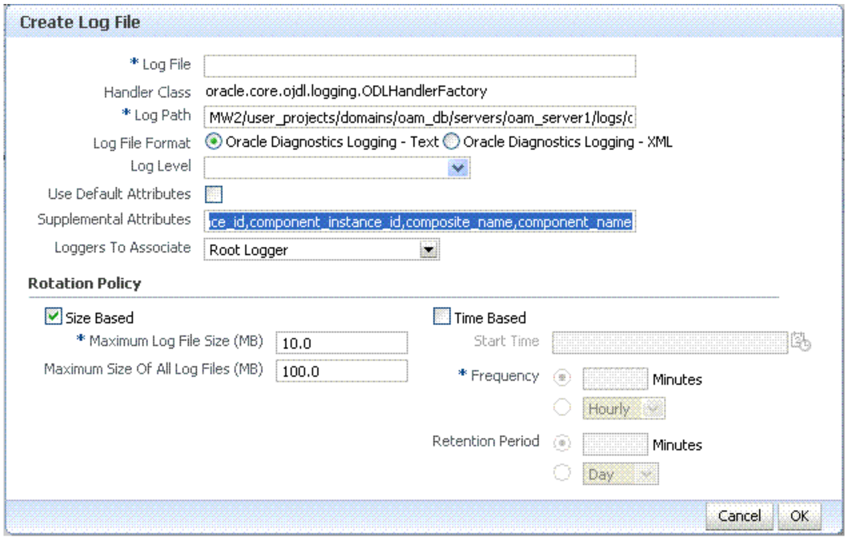
Element	Description
Create	Click this button to display the fresh form to create a new file for logged messages. Notes: - Log File is the name of the log handler (odl-handler for OAM) - Log Path points to the logging output file in your environment, which you can change. - The output logging file in 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 26–9 (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.

26.7.2 Managing Log File Configuration by Using Fusion Middleware Control

Fusion Middleware Control administrators can use the following procedure to create a log file, edit the configuration, or view a read-only version of the log file configuration.

> **See Also:** "About Log File Configuration" on page 26-25

To manage log files for OAM in Fusion Middleware Control

1. Log in as described in "Logging In To Fusion Middleware Control" on page 26-3.
2. Expand the desired node, and select a target. For example:

   Identity and Access
   oam_server

3. From the Oracle Access Manager menu, select Logs and then Log Configuration.
4. **Create a Log File**: From the Log Files tab (Table 26–9):
   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.
5. **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 your own information:

      Log File name

      Log Level

      Attributes

   d. Edit any other details as needed, then click **OK** to submit the configuration.

6. **Edit Configuration**:
   a. From the Log Files tab, click the name of an existing log file.
   b. Click the **Edit Configuration** button.
   c. Change configuration details as needed.
   d. Click **OK** to submit the changes.
7. **View Configuration**:
   a. From the Log Files tab, click the name of an existing log file.
   b. Click the **View Configuration** button.
   c. Review the information, then click **OK** to dismiss the configuration page.
8. Proceed to "Viewing Log Messages in Fusion Middleware Control".

26.8 Viewing Log Messages in Fusion Middleware Control

This section includes the following topics:

- About Finding, Viewing, and Exporting Log Messages
- Viewing Logged Messages With Fusion Middleware Control

26.8.1 About Finding, Viewing, and Exporting Log Messages

By using the context menu for an Oracle Access Manager Server instance in Fusion Middleware Control, administrators can locate, view, and export key log information for:

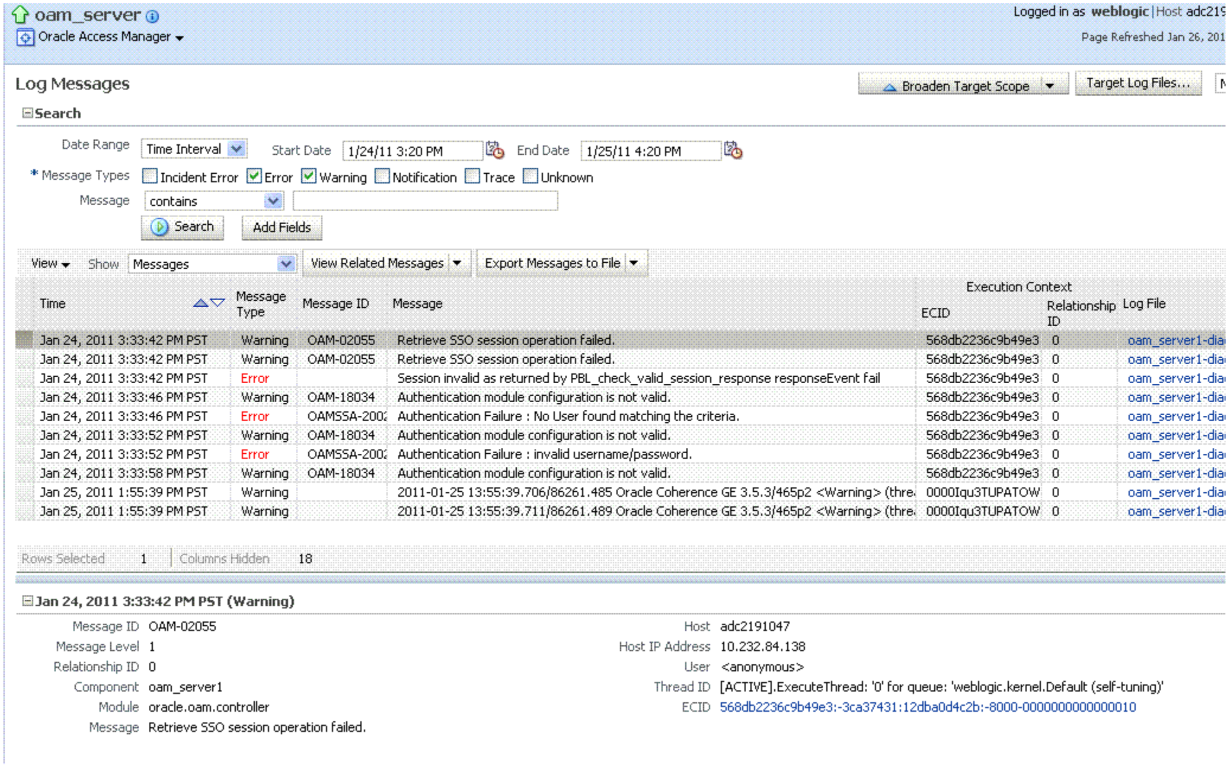
- Application Deployment targets, including the WebLogic (and OAM) AdminServer and the OAM SSO logout pages on both AdminServer and OAM Servers
- WebLogic Server domain targets, including the OAM Farm, AdminServer, and OAM Servers
- Identity and Access targets, including the Oracle Access Manager Farm, Clusters, and individual 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 26–20 shows the Log Messages page for Oracle Access Manager with Oracle Security Token Service in Fusion Middleware Control.

Figure 26–20 Typical Log Messages Page in Fusion Middleware Control



Search

Date Range	Time Interval Start Date 1/24/11 3:20 PM End Date 1/25/11 4:20 PM
* Message Types	□ Incident Error ☑ Error ☑ Warning □ Notification □ Trace □ Unknown
Message	contains

Show: Messages

Time	Message Type	Message ID	Message	Execution Context ECID	 Relationship ID	Log File
Jan 24, 2011 3:33:42 PM PST	Warning	OAM-02055	Retrieve SSO session operation failed.	568db2236c9b49e3	0	oam_server1-dia
Jan 24, 2011 3:33:42 PM PST	Warning	OAM-02055	Retrieve SSO session operation failed.	568db2236c9b49e3	0	oam_server1-dia
Jan 24, 2011 3:33:42 PM PST	Error		Session invalid as returned by PBL_check_valid_session_response responseEvent fail	568db2236c9b49e3	0	oam_server1-dia
Jan 24, 2011 3:33:46 PM PST	Warning	OAM-18034	Authentication module configuration is not valid.	568db2236c9b49e3	0	oam_server1-dia
Jan 24, 2011 3:33:46 PM PST	Error	OAMSSA-200:	Authentication Failure : No User found matching the criteria.	568db2236c9b49e3	0	oam_server1-dia
Jan 24, 2011 3:33:52 PM PST	Warning	OAM-18034	Authentication module configuration is not valid.	568db2236c9b49e3	0	oam_server1-dia
Jan 24, 2011 3:33:52 PM PST	Error	OAMSSA-200:	Authentication Failure : invalid username/password.	568db2236c9b49e3	0	oam_server1-dia
Jan 24, 2011 3:33:58 PM PST	Warning	OAM-18034	Authentication module configuration is not valid.	568db2236c9b49e3	0	oam_server1-dia
Jan 25, 2011 1:55:39 PM PST	Warning		2011-01-25 13:55:39.706/86261.485 Oracle Coherence GE 3.5.3/465p2 <Warning> (thre:	0000Iqu3TUPATOW	0	oam_server1-dia
Jan 25, 2011 1:55:39 PM PST	Warning		2011-01-25 13:55:39.711/86261.489 Oracle Coherence GE 3.5.3/465p2 <Warning> (thre:	0000Iqu3TUPATOW	0	oam_server1-dia

Rows Selected 1 | Columns Hidden 18

Jan 24, 2011 3:33:42 PM PST (Warning)

Message ID	OAM-02055	Host	adc2191047
Message Level	1	Host IP Address	10.232.84.138
Relationship ID	0	User	<anonymous>
Component	oam_server1	Thread ID	[ACTIVE].ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'
Module	oracle.oam.controller	ECID	568db2236c9b49e3:-3ca37431:12dba0d4c2b:-8000-0000000000000010
Message	Retrieve SSO session operation failed.		

Table 26–10 describes elements on the Log Messages page in Fusion Middleware Control, which you can use to locate and view messages.

Table 26–10 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

Table 26–10 (Cont.) OAM Log Message Search Controls in Fusion Middleware Control

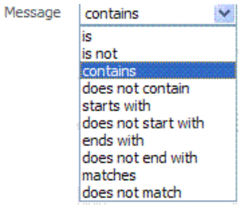
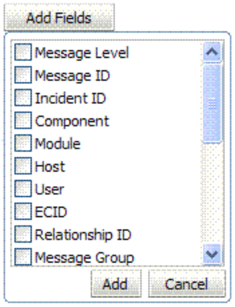
Element	Description
Date Range	The period during which the desired set of messages was logged: ■ Most Recent Minutes Hours Days ■ Time interval Date Range Start Date End Date
Message Types	Check all message types that apply for this search: ■ Incident Error ■ Error ■ Warning ■ Notification ■ Trace ■ Unknown
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	

Table 26–10 (Cont.) OAM Log Message Search Controls in Fusion Middleware Control

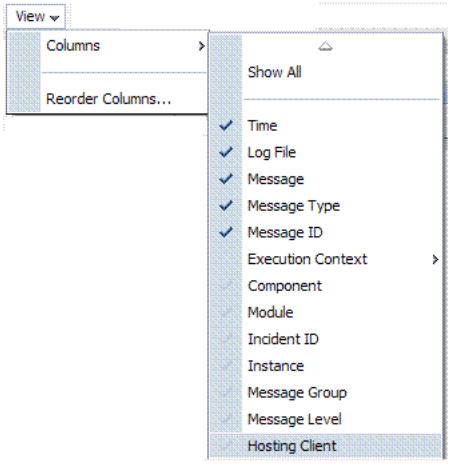
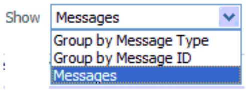
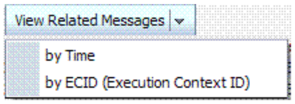
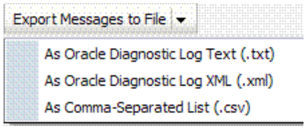
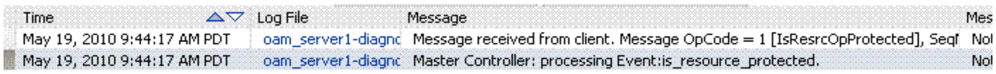
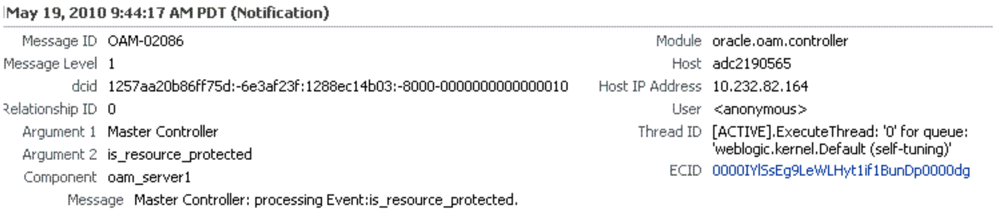
Element	Description
View	Choose items from this menu to view or reorder columns in the search results table:  View Columns Reorder Columns... Show All Time Log File Message Message Type Message ID Execution Context Component Module Incident ID Instance Message Group Message Level Hosting Client
Show	Select the entity to view:  Show Messages Group by Message Type Group by Message ID Messages
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)

Table 26–10 (Cont.) OAM Log Message Search Controls in Fusion Middleware Control

Element	Description
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 | Mes May 19, 2010 9:44:17 AM PDT | oam_server1-diagnc | Message received from client. Message OpCode = 1 [IsResrcOpProtected], Seql | Nol May 19, 2010 9:44:17 AM PDT | oam_server1-diagnc | Master Controller: processing Event:is_resource_protected. | Nol
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.

26.8.2 Viewing Logged Messages With 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 26-29

To view OAM Server log messages within Fusion Middleware Control

1. Log in as described in "Logging In To Fusion Middleware Control" on page 26-3.
2. Expand the desired node and select a target. For example:

   Identity and Access
     oam_server

3. From the OAM context menu, select Logs and then choose View Log Messages.
4. **Search** (Table 26–10):
   a. Specify a 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:** From the table of search results, click one or more messages to view on the lower half of the page.
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 a 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 14).
    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 26-21.

26.9 Displaying 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 relies 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
- Managing Mbeans

26.9.1 About the System MBean Browser

The Fusion Middleware Control System Mbean Browser can be used to view the items outlined in Table 26–11.

Table 26–11 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

Note: Oracle Security Token Service MBeans are also available as described here.

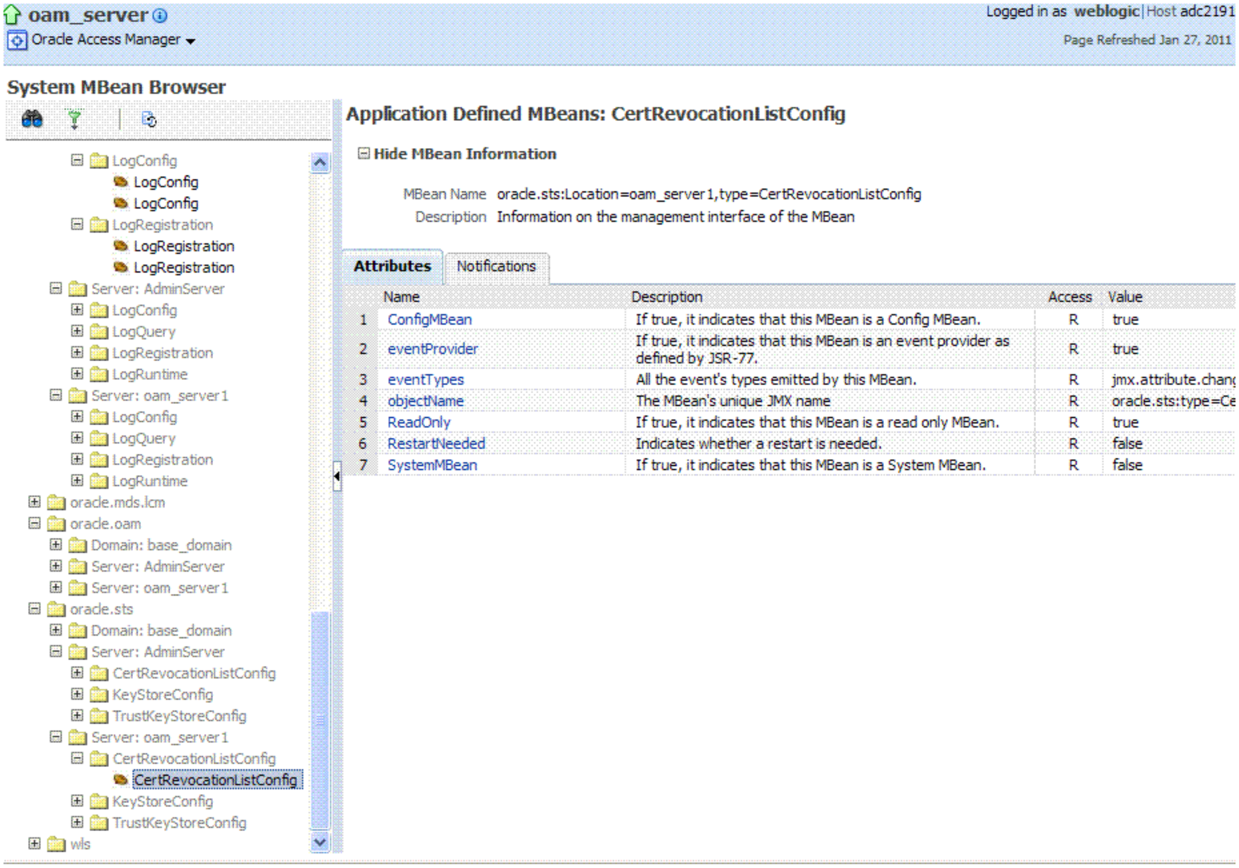
Table 26–12 describes the MBeans that Oracle Access Manager and Oracle Security Token Service deploy on the AdminServer on the domain runtime server (OAM Server).

Table 26–12

MBeans For	Description
Configuration Service	oracle.oam:type=Config
Partner and Trust Service	oracle.oam:type=PATConfig
STS MBeans	oracle.sts:type=Config
Certificate Validation Module	These are used for CRL management. oracle.sts:type=CertRevocationListConfig

Figure 26–21 Shows the System MBean Browser and the related Attributes tab displaying information for the Oracle Security Token Service CertRevocationListConfig: `oracle.sts:Location=oam_server1,type=CertRevocationListConfig`.

Figure 26–21 System MBean Browser and Attributes Tab



oam_server

Oracle Access Manager

Logged in as weblogic | Host adc2191

Page Refreshed Jan 27, 2011

System MBean Browser

- LogConfig
  - LogConfig
  - LogConfig
- LogRegistration
  - LogRegistration
  - LogRegistration
- Server: AdminServer
  - LogConfig
  - LogQuery
  - LogRegistration
  - LogRuntime
- Server: oam_server1
  - LogConfig
  - LogQuery
  - LogRegistration
  - LogRuntime
- oracle.mds.lcm
- oracle.oam
  - Domain: base_domain
  - Server: AdminServer
  - Server: oam_server1
- oracle.sts
  - Domain: base_domain
  - Server: AdminServer
    - CertRevocationListConfig
    - KeyStoreConfig
    - TrustKeyStoreConfig
  - Server: oam_server1
    - CertRevocationListConfig
      - CertRevocationListConfig
    - KeyStoreConfig
    - TrustKeyStoreConfig
- wls

Application Defined MBeans: CertRevocationListConfig

Hide MBean Information

MBean Name oracle.sts:Location=oam_server1,type=CertRevocationListConfig

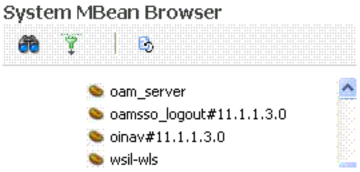
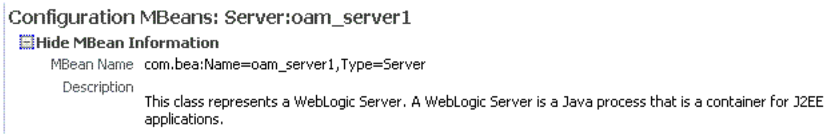
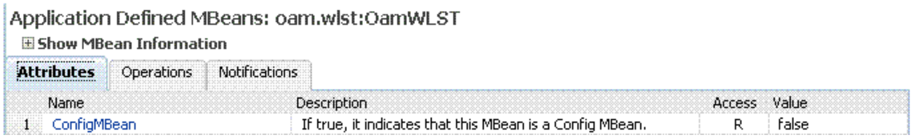
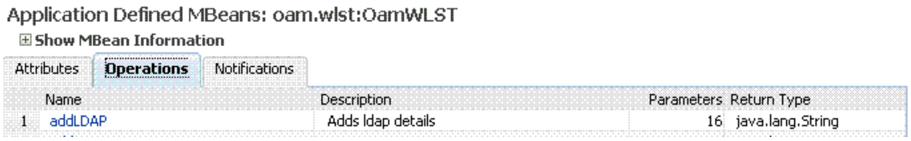
Description Information on the management interface of the MBean

Attributes | Notifications

	Name	Description	Access	Value
1	ConfigMBean	If true, it indicates that this MBean is a Config MBean.	R	true
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.chang
4	objectName	The MBean's unique JMX name	R	oracle.sts:type=Ce
5	ReadOnly	If true, it indicates that this MBean is a read only MBean.	R	true
6	RestartNeeded	Indicates whether a restart is needed.	R	false
7	SystemMBean	If true, it indicates that this MBean is a System MBean.	R	false

Table 26–13 describes the System MBean Browser and associated tab in greater details.

Table 26–13 System MBean Browser

System MBean Browser	
System MBean Browser	Expand items in this section to display Mbeans for the selected target. Under Application Defined Beans, find `oracle.oam` and `oracle.sts`.  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
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.
Controls	The following controls are available from these pages: ■ Name Link: Clicking a name on either tab displays a full description of related MBeans. ■ Apply Button: Submits and applies the selected MBean attribute value. ■ Revert Button: Restores previous MBean attribute values following a change (and before clicking Apply. ■ Return Button: Returns you to the MBean Information page. ■ Invoke Button: Invokes the selected MBean and value

26.9.2 Managing Mbeans

Fusion Middleware Control administrators can use the following procedure to view MBeans for Oracle Access Manager or Oracle Security Token Service. Additionally, you can apply values (or revert the change) and invoke MBeans.

To view, edit, or invoke MBeans for Oracle Access Manager and Oracle Security Token Service

1. Log in as described in "Logging In To Fusion Middleware Control" on page 26-3.

2. Expand the desired node and select a target. For example:

   Identity and Access
      oam_server

3. From the Oracle Access Manager context menu, select **System MBean Browser**.
4. **System MBean Browser**: Expand classes and select an MBean target to display related attributes and operations. For example: **oracle.sts** or **oracle.oam**.
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 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.

26.10 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.

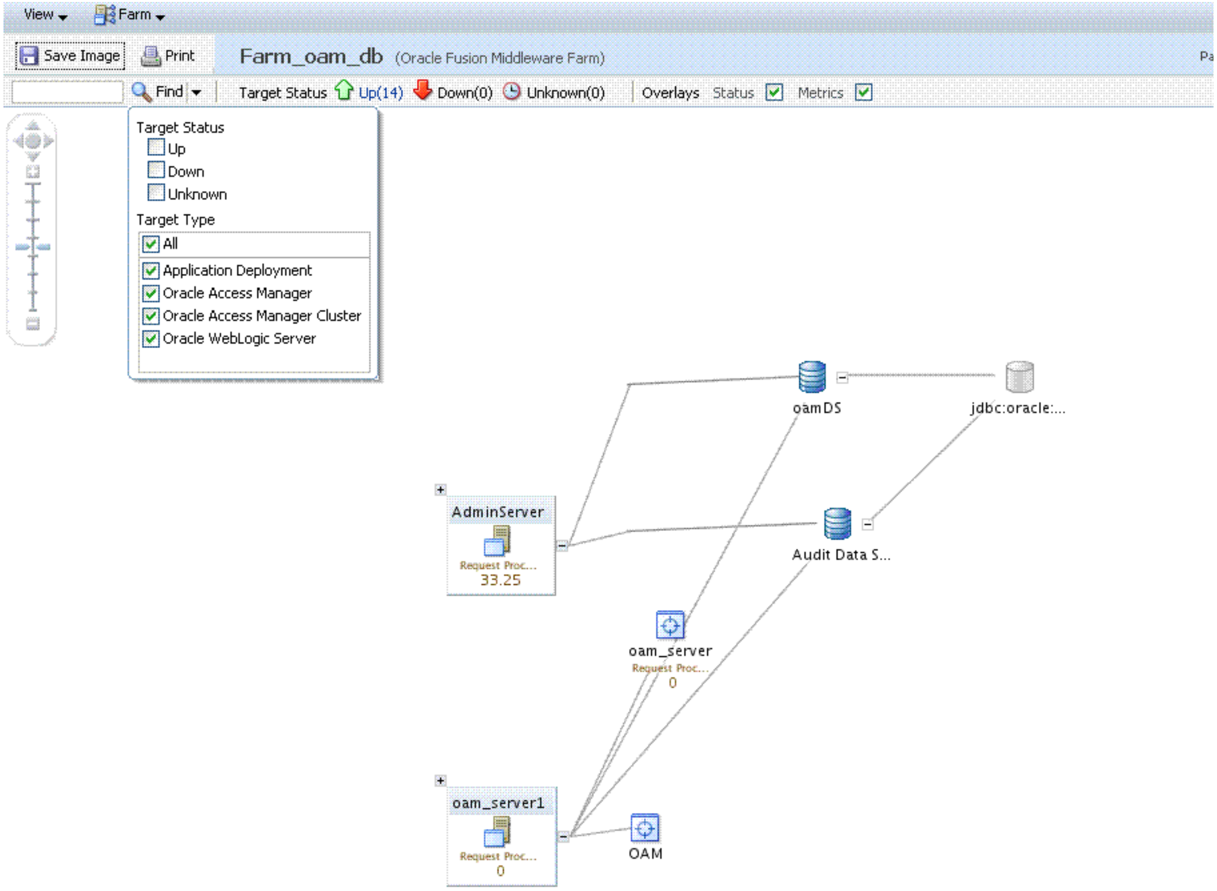
This section provides the following topics:

- About the Routing Topology
- Viewing the Routing Topology using Fusion Middleware Control

26.10.1 About the Routing Topology

Figure 26–22 shows the Farm routing topology page in Fusion Middleware Control.

Figure 26–22 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 26–14 describes the status and controls on the Farm topology page.

Table 26–14 Farm Topology

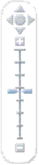
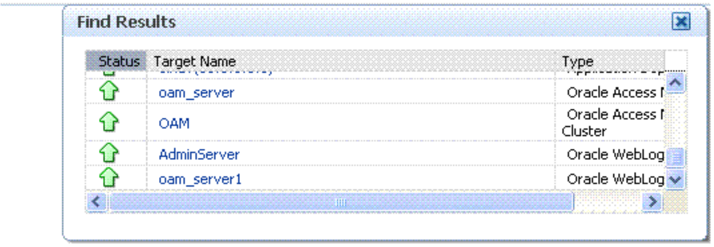
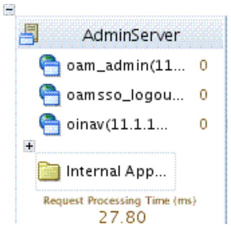
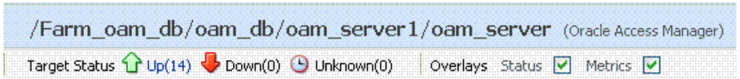
Element	Description
Save Image	Saves the image.
Print	Prints the image.
	Scales the image.

Table 26–14 (Cont.) Farm Topology

Element	Description
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 ☑

26.10.2 Viewing the 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 26-3.
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 26–14).
9. Click Print or Save, as needed.

Part VII

Using 10g Webgates with Oracle Access Manager 11g

When your enterprise includes Web server types other than Oracle HTTP Server, you can install 10g Webgates to use with Oracle Access Manager 11g.

Part VII contains the following chapters:

- Chapter 27, "Managing OAM 10g Webgates with OAM 11g"
- Chapter 28, "Configuring Apache, OHS, IHS for 10g Webgates"
- Chapter 29, "Configuring the IIS Web Server for 10g Webgates"
- Chapter 30, "Configuring the ISA Server for 10g Webgates"
- Chapter 31, "Configuring Lotus Domino Web Servers for 10g Webgates"

27 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 IAMSuiteAgent 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 IAMSuiteAgent 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 9.

27.1 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 Oracle Access Manager Console is running and get familiar with:

- Introduction to Policy Enforcement Agents on page 9-1
- Introduction to OAM 10g Agents for OAM 11g in this chapter

27.2 Introduction to OAM 10g Agents for OAM 11g

This section provides the following topics:

- About Replacing the IAMSuiteAgent with an OAM 10g Webgate
- About Legacy OAM 10g Deployments and Webgates
- About Installing Fresh OAM 10g Webgates to Use With OAM 11g

27.2.1 About Replacing the IAMSuiteAgent with an OAM 10g Webgate

As described in Chapter 9, the IAMSuiteAgent 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 IAMSuiteAgent 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 IAMSuiteAgent with an OAM 10g Webgate" on page 27-15.

27.2.2 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 Fusion Middleware Administrator's Guide for Oracle Access Manager with Oracle Security Token Service.
- Legacy 10g Webgates configured as the Identity Assertion Provider (IAP) for SSO (for applications using IAP 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 Oracle Access Manager 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

27.2.3 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 27–1 outlines these differences.

Table 27–1 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 27-4.	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 15, "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

27.3 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 Oracle Access Manager 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 *my-10g-agent1*, protecting */.../**, and declaring a public resource, /public/index.html. Your values will be different. You can use a full registration template to specify public, private, and excluded resources.

See Also:

- "Replacing the IAMSuiteAgent with an OAM 10g Webgate" on page 27-15 if needed
- Chapter 10 for more information about the remote registration tool, processing, and request files
- Chapter 9 if you prefer using the Oracle Access Manager 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:

   ```
   ORACLE_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 (oamreg.bat or oamreg.sh), set the following environment variables based on your situation (client side or server side) and information in Table 10–7:

   OAM_REG_HOME = *exploded_dir_for_RREG*.tar/rreg
   JAVA_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:

   ```
   $OAM_REG_HOME/input/OAMRequest_short.xml/
   ```

   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>
       <autoCreatePolicy>true</autoCreatePolicy>
       <primaryCookieDomain>.us.example.com</primaryCookieDomain>
       <logOutUrls>
   ```

```
      <url>/oamsso/logout.html</url>
    </logOutUrls>
</OAMRegRequest>
```

See Also: "Creating the Registration Request" on page 10-23

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/*my-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 Oracle Access Manager Console and review the new registration:

   a. From the System Configuration tab, Access Manager Settings section, expand the following nodes:

      SSO 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 System Configuration tab, Common Configuration section, 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 the time being.
6. Add resources to the application domain
7. 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 IAMSuiteAgent with an OAM 10g Webgate on page 27-15

27.4 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

27.4.1 Preparing for a Fresh 10g Webgate Installation with OAM 11g

Table 27–2 outlines the requirements that must be met before starting an OAM 10g Webgate installation.

Table 27–2 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.
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 28 ■ The APACHE22_Webgate supports v2.2 with or without SSL (and with or without reverse proxy enabled on Solaris and Linux). See also Chapter 28 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 28.

Table 27–2 (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 31, "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 29-12. 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 29, "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 30, "Configuring the ISA Server for 10g Webgates"

27.4.2 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".

27.4.3 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.

27.4.4 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".

27.4.5 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".

27.4.6 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.

27.4.7 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 27-12.
3. Return to the Webgate installation screen, click Next, and proceed to "Provisioning a 10g Webgate with OAM 11g" on page 27-4.

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 29, "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".

27.4.7.1 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".

27.4.8 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 29.

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 28, "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 29, "Configuring the IIS Web Server for 10g Webgates".
   - **ISA Web Servers**: Chapter 30, "Configuring the ISA Server for 10g Webgates"
   - **Lotus Domino Web Servers**: Chapter 31, "Configuring Lotus Domino Web Servers for 10g Webgates"

27.4.9 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.

27.4.10 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 IAMSuiteAgent, proceed to "Updating the WebLogic Server Plug-in" on page 27-18

   - **Unsuccessful**: Webgate should be uninstalled and reinstalled, as described in "Removing a 10g Webgate from the OAM 11g Deployment" on page 27-25.

27.5 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 logged out (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 15-7.

27.6 Replacing the IAMSuiteAgent 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 IAMSuiteAgent 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 IAMSuiteAgent.

To protect applications beyond containers, you can replace the IAMSuiteAgent with a 10g Webgate (to protect the same set of applications using the same application domain and policies as the pre-registered IAMSuiteAgent).

Task overview: Replacing the IAMSuiteAgent with an OAM 10g Webgate

1. Provisioning a 10g Webgate to Replace the IAMSuiteAgent
2. Installing a 10g Webgate to Replace the IAMSuiteAgent
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 IAMSuiteAgent
7. Verification

27.6.1 Provisioning a 10g Webgate to Replace the IAMSuiteAgent

Provisioning is the process of creating a Webgate registration in the Oracle Access Manager Console. The following procedure walks through provisioning using the remote registration tool, in-band mode.

See Also:

- Chapter 10 for more information about the remote registration tool, processing, and request files
- Chapter 9 if you prefer using the Oracle Access Manager 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 IAM Suite policies with the replacement Webgate, ensure that the Webgate registration is configured to use the IAMSuiteAgent Host Identifier and Preferred Host.

To reuse existing IAM Suite policies you can specify IAMSuiteAgent as the hostidentifier in the OAMReqRequest xml for the Webgate registration to set IAMSuiteAgent as the HostIdentifier and preferredHost. Alternatively, you can edit the Agent registration using the Oracle Access Manager Console.

To provision a 10g Webgate to replace the IAMSuiteAgent

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:

   ```
   ORACLE_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 10–7:

   OAM_REG_HOME = *exploded_dir_for_RREG*.tar/rreg
   JAVA_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: *10g4IAM*.xml

   b. Edit *10g4IAM*.xml to include details for your environment. For example, if you are changing from the IAMSuiteAgent to a 10g Webgate Agent your request might look like the following:

   ```
   <OAMRegRequest>
       <serverAddress>http://sample.us.oracle.com:7001</serverAddress>
       <hostIdentifier>10g4IAM</hostIdentifier>
       <agentName>10g4IAM</agentName>
       <autoCreatePolicy>false</autoCreatePolicy>
       <primaryCookieDomain>.us.example.com</primaryCookieDomain>
       <logOutUrls><url>/oamsso/logout.html</url></logOutUrls>
       ...retain defaults for remaining elements...
       ...
       ...
   </OAMRegRequest>
   ```

   See Also: "Creating the Registration Request" on page 10-23

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/10g4IAM.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 Oracle Access Manager Console and review the new registration:

   a. From the System Configuration tab, Access Manager Settings section, expand the following nodes:

      SSO 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 Server with which this Webgate is registered.

   c. **OAM Proxy Port**—From the System Configuration tab, Common Configuration section, 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".

27.6.2 Installing a 10g Webgate to Replace the IAMSuiteAgent

After provisioning you must install the 10g Webgate to replace the IAMSuiteAgent. 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 IAMSuiteAgent

Task overview: Installing the Webgate includes

1. Locating and Installing the Latest OAM 10g Webgate for OAM 11g
2. Replacing the IAMSuiteAgent: Proceed to "Updating the WebLogic Server Plug-in".

27.6.3 Updating the WebLogic Server Plug-in

After provisioning and installing the 10g Webgate to replace the IAMSuiteAgent, 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 27–1 illustrates the areas that must be changed using sample entries. Entries for your environment will be different.

Example 27–1 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 IAMSuiteAgent

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 27–1).
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

27.6.4 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 Oracle Identity Manager is integrated with OAM 11g requires the 10g Webgate Web server host name and port in the list of host identifiers for the IAMSuiteAgent.

> **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 Oracle Access Manager/Oracle Identity Manager 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 IAMSuiteAgent:

   Shared Components
     Host Identifiers
       IAMSuiteAgent

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".

27.6.5 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

27.6.5.1 About Security Providers

To complete the Oracle Access Manager 11g SSO configuration when a 10g Webgate is replacing the IAMSuiteAgent 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*.

27.6.5.2 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 IAMSuiteAgent".
- **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 27-20.

27.6.6 Disabling the IAMSuiteAgent

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 IAMSuiteAgent

1. On the computer hosting the IAMSuiteAgent, 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.

27.6.7 Verification

Oracle recommends testing your environment using the 10g Webgate to ensure that all applications that were previously protected by the IAMSuiteAgent are now protected after configuring the 10g Webgate.

See Also:

- "Validating Authentication and Authorization in an Application Domain" on page 13-50
- Chapter 14, "Validating Connectivity and Policies Using the Access Tester"

27.7 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:

27.8 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 29-27.

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 Oracle Access Manager 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 28, Chapter 29, Chapter 30, Chapter 31.

8. Restart the Web server.

9. Remove the *Webgate_install_dir* directory if it remains, especially if you plan to reinstall it.

28

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:

- About Oracle HTTP Server and Oracle Access Manager
- About Oracle Access Manager with Apache and IHS v2 Webgates
- About Apache v2 Architecture and Oracle Access Manager
- Requirements for Oracle HTTP Server, IHS, Apache v2 Web Servers
- Preparing Your Web Server
- Activating Reverse Proxy for Apache v2 and IHS v2
- Verifying httpd.conf Updates for Oracle Access Manager Webgates
- Tuning Oracle HTTP Server for Oracle Access Manager Webgates
- Tuning OHS /Apache Prefork and MPM Modules for OAM
- Starting and Stopping Oracle HTTP Server Web Servers
- Tuning Apache/IHS v2 for Oracle Access Manager Webgates
- Removing Web Server Configuration Changes After Uninstall
- Helpful Information

28.1 Prerequisites

Ensure that your Oracle Access Manager Console is running and get familiar with:

- "Introduction to Policy Enforcement Agents" on page 9-1
- "About Installing Fresh OAM 10g Webgates to Use With OAM 11g" on page 27-2

28.2 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 28-7.

28.3 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.

28.3.1 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 28-4.

28.3.2 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 28-4 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
```

28.3.3 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 28-6.

28.4 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 I-25.

- **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 28-28.

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 #lfdefs. 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 28-28.

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 I-25.

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">
```

28.5 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

28.5.1 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`.

28.5.2 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 28-19.

28.5.3 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.

28.6 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 28-3. See also "Activating Reverse Proxy for Apache v2 and IHS v2" on page 28-19.

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 28-19.
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 28-22.
5. Refer to the following topics as needed:
   - "Tuning Oracle HTTP Server for Oracle Access Manager Webgates" on page 28-25
   - "Tuning OHS /Apache Prefork and MPM Modules for OAM" on page 28-26
   - "Tuning Apache/IHS v2 for Oracle Access Manager Webgates" on page 28-28

> **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.

28.6.1 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.

28.6.1.1 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 28-6 and "Requirements for Apache v2 Web Servers" on page 28-6.

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 28-9.

28.6.1.2 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

28.6.1.3 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

28.6.1.4 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.

28.6.2 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 28-28.

> **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.

28.6.3 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.

28.6.4 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.

28.6.5 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 28-16

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 28-16.

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 I-25.

   > **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 28-3.

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 28-16 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.

28.6.6 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 28-12 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.

28.6.7 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.

28.7 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

28.7.1 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 28-3.
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 27, "Managing OAM 10g Webgates with OAM 11g", if you haven't yet done so.

28.7.2 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 28-3
2. Install the IHS v2 Web server, as described in "Preparing the IHS v2 Web Server" on page 28-8.
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 27, "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.

28.8 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 28-20. To customize httpd.conf for your Web server, see your Web server documentation.

28.8.1 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 28-28 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 27, "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 28-28.

28.8.2 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.

28.9 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"

28.10 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

28.10.1 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

28.10.2 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.)

28.10.3 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.

28.11 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
   ```

28.12 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 I-19

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.

28.13 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.

28.14 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

29

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

29.1 Prerequisites

Ensure that your Oracle Access Manager Console is running and get familiar with:

- Introduction to Policy Enforcement Agents on page 9-1
- About Installing Fresh OAM 10g Webgates to Use With OAM 11g on page 27-2

29.2 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

29.2.1 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 27-4.

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 29-4
- Webgates for IIS v6 on page 29-4

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 29-4
- Finishing 64-bit Webgate Installation on page 29-24

General Webgate Preparation and Installation Details: Refer to this chapter for IIS-specific guidelines. Refer to Chapter 27 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

29.2.1.1 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 29-5
- Updating IIS 7 Web Server Configuration on Windows 2008 on page 29-6
- Installing and Configuring Multiple 10g Webgates for a Single IIS 7 Instance on page 29-14

29.2.1.2 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 27. 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 29-24.

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)

29.2.1.3 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 29-19

29.3 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

29.3.1 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.

29.3.2 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.

29.4 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 29–1 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".

29.5 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 29-24
> - Installing and Configuring Multiple Webgates for a Single IIS 6 Instance on page 29-19
>
> If you have IIS v7, Oracle recommends the following topics:
>
> - Updating IIS 7 Web Server Configuration on Windows 2008 on page 29-6
> - Installing and Configuring Multiple 10g Webgates for a Single IIS 7 Instance on page 29-14

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 29-7
2. Ordering the ISAPI Filters on page 29-8
3. Enabling Pass-Through Functionality for POST Data on page 29-9
4. Protecting a Web Site When the Default Site is Not Setup on page 29-13

29.5.1 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.

29.5.2 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.**

29.5.3 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 10g 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

29.5.3.1 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*.

29.5.3.2 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.

29.5.3.3 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 27-6.
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".

29.5.3.3.1 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 I-14

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".

29.5.3.3.2 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.

29.5.3.4 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 Chapter 27, "Managing OAM 10g Webgates with OAM 11g".
2. Set up IIS 6.0 as described in "Setting Up IIS 6.0 Web Server in IIS 5.0 Isolation Mode" on page 29-11.
3. Install postgate.dll as described in "Installing the Postgate ISAPI Filter""Installing the Postgate ISAPI Filter".

29.5.3.4.1 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".

29.5.3.4.2 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.

29.5.4 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 29-25

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.

29.6 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 29-8
   - "Confirming Webgate Installation on IIS" on page 29-26

   See Also: "Confirming Multiple Webgate Installation" on page 29-24

29.6.1 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 27.
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 29-16
   b. "Enabling Client Certification for Multiple IIS 7 Webgates" on page 29-17
9. Repeat these steps when you install the next Webgate for the IIS instance.

29.6.2 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 29-19.

29.6.3 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"

29.6.4 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".

29.6.5 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.

29.7 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 29-14

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 29-8
   - "Confirming Webgate Installation on IIS" on page 29-26

   See Also: "Confirming Multiple Webgate Installation" on page 29-24

29.7.1 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 27.
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 29-22

   b. "Enabling SSL and Client Certification for Multiple Webgates" on page 29-23

9. Repeat these steps when you install the next Webgate for the IIS instance.

29.7.2 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 29-24.

29.7.3 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".

29.7.4 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 29-24
- "Confirming Webgate Installation on IIS" on page 29-26

29.8 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 29-7.

Before you start tasks here, be sure that you have completed Webgate installation according to information in Chapter 27. 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 29-4.

Task overview: Finishing installation of a 64-bit Webgate

1. Perform steps in "Setting Access Permissions, ISAPI filters, and Directory Security Authentication" on page 29-25.
2. Enable client certificates, if desired. See "Setting Client Certificate Authentication" on page 29-26.
3. When finished, you can:
   - Confirm operations as described in "Confirming Webgate Installation on IIS" on page 29-26
   - Implement Windows Impersonation, as described in the Oracle Fusion Middleware Integration Guide for Oracle Access Manager.

29.8.1 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 29-13.
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 29-8 to ensure that all filters have been added and are in the proper order.

29.8.2 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 29-7.

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 29-8.
7. Proceed to "Confirming Webgate Installation on IIS" on page 29-26.

29.9 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 27-25, in the chapter on installing a 10g Webgate Chapter 27.

29.10 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.

29.11 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.

30 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

30.1 Prerequisites

Ensure that your Oracle Access Manager Console is running and get familiar with:

- "Introduction to Policy Enforcement Agents" on page 9-1
- "About Oracle Access Manager and the ISA Server" on page 30-1

30.2 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 Fusion Middleware Administrator's Guide for Oracle Access Manager with Oracle Security Token Service 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 30-2
2. "Installing and Configuring Webgate for the ISA Server" on page 30-2.
3. "Configuring the ISA Server for the ISAPI Webgate" on page 30-3.
4. Perform the following tasks, as described in:
   a. "Ordering the ISAPI Filters" on page 30-6
   b. "Removing Oracle Access Manager Filters Before Webgate Uninstall on ISA Server" on page 30-7

30.3 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
```

30.4 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 30-2

Task overview: Performing Webgate configuration for ISA Server includes

1. "Installing Webgate with ISA Server" on page 30-3
2. "Changing /access Directory Permissions" on page 30-3
3. "Registering Oracle Access Manager Plug-ins as ISA Server Web Filters" on page 30-4

30.4.1 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 27 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 30-3

30.4.2 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**".

30.5 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 30-4
2. "Configuring ISA Firewall Policies for ISA Web Filters" on page 30-4

30.5.1 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".

30.5.2 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.

30.5.3 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.**

30.6 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.

30.7 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 27 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
```

31 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.

31.1 Prerequisites

Ensure that your Oracle Access Manager Console is running and get familiar with:

- "Introduction to Policy Enforcement Agents" on page 9-1
- "About Installing Fresh OAM 10g Webgates to Use With OAM 11g" on page 27-2

31.2 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.

31.3 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.

31.4 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.

31.5 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

31.6 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 27-6.
3. See "Completing the Webgate Installation" on page 31-5 and choose one of the two options discussed there.

31.6.1 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, "Co-existence Overview: OAM 11g and OSSO 10g"
- Appendix B, "Transitioning OAM 11g from a Test to a Production Environment"
- 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 for Oracle Access Manager 11g"
- Appendix F, "Introduction to Custom WLST Commands for Administrators"
- Appendix G, "Configuring OAM 11g for IPv6 Clients"
- Appendix H, "Creating Deployment-Specific Pages"
- Appendix I, "Troubleshooting"

A

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

A.1 Prerequisites

See *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management* for upgrade steps.

A.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".

A.3 Pre- and Post-Upgrade Topology and Authentication Examples

This section provides the following topics:

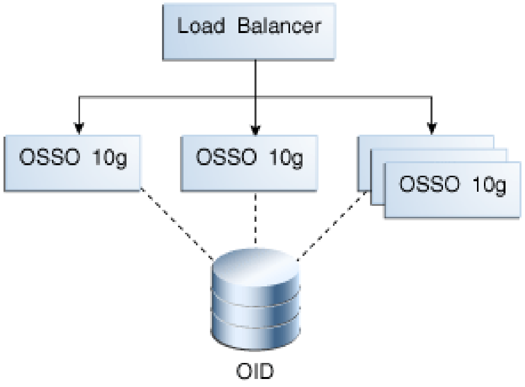
- About Pre-Upgrade OSSO 10g Topology
- About Post-Upgrade Topology and Co-existence
- Simple OSSO 10g with mod_oc4j on a Front-End Proxy Server
- Post-Upgrade: mod_wl Replaces mod_oc4j on the Proxy Server
- Post-Upgrade: No Proxy Server

See Also: "Introduction to Validating Post-Upgrade Co-Existence with OAM 11g" on page A-5

A.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 A–1.

Figure A–1 Pre-Upgrade OSSO 10g Topology

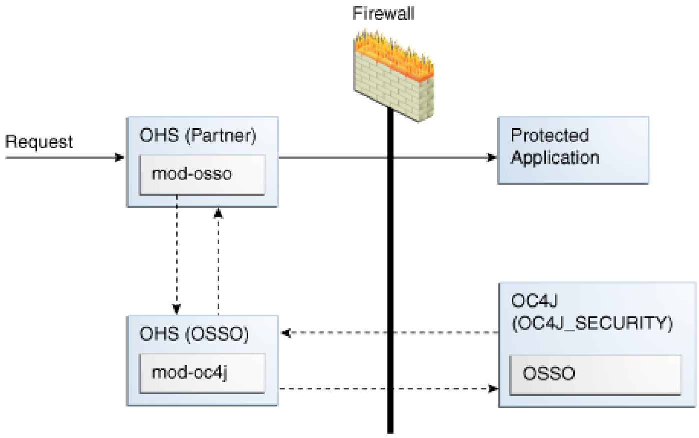


See Also: "Simple OSSO 10g with mod_oc4j on a Front-End Proxy Server"

A.3.1.1 Simple OSSO 10g with mod_oc4j on a Front-End Proxy Server

Figure A–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 A–2 Pre-Upgrade Sample OSSO 10g with Front-End Proxy



After upgrading the environment is configured to use Oracle Access Manager 11g for authentication.

A.3.2 About Post-Upgrade Topology and Co-existence

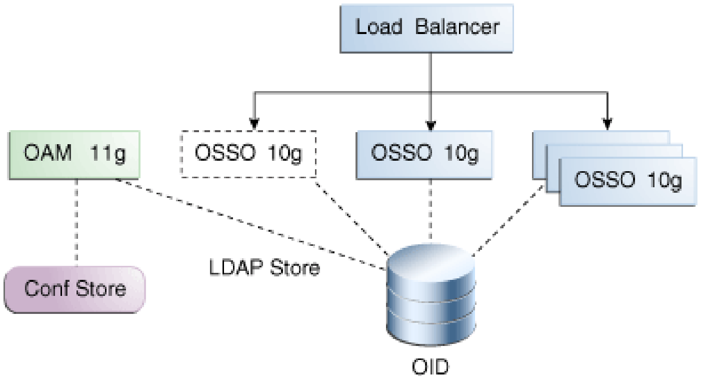
Upgrading to OAM 11g starts by installing a new OAM 11g Server and transferring the partner applications to this OAM Server.

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 A–3.

Figure A–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.

See Also: "About Single Sign-On Cookies" on page 11-14

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

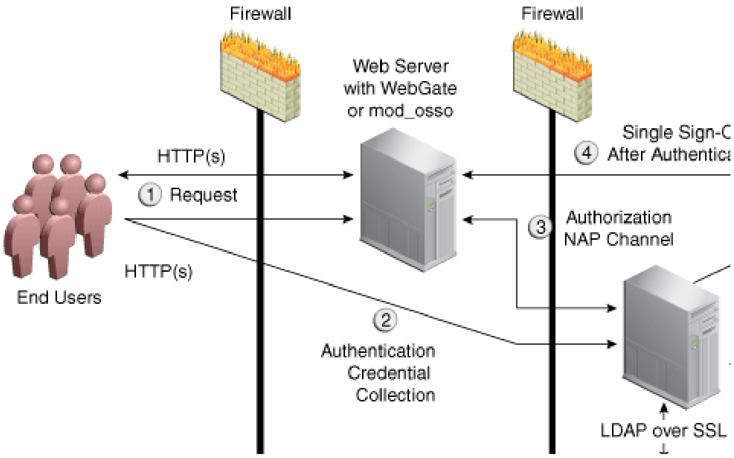
A.3.2.1 Post-Upgrade: mod_wl Replaces mod_oc4j on the Proxy Server

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 A–4 illustrates this post-upgrade topology with mod_wl replacing mod_oc4j.

Figure A–4 mod_wl Replaces mod_oc4j on the Proxy Server



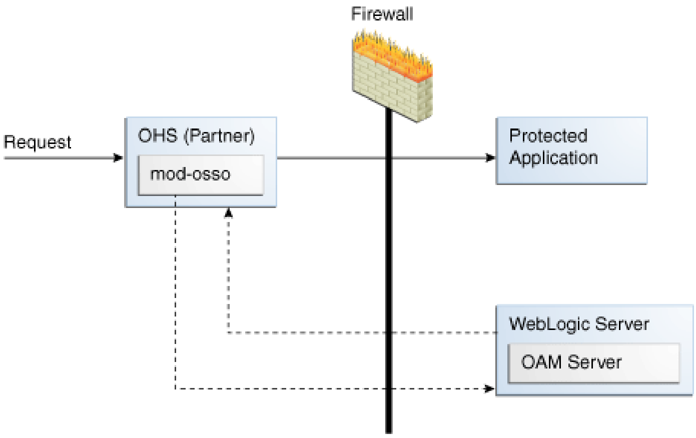
See Also: "Post-Upgrade: No Proxy Server"

A.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 A–5 illustrates this topology, which most deployments will use.

Figure A–5 Typical Topology Without Proxy Server



A.4 Introduction to Validating Post-Upgrade Co-Existence with OAM 11g

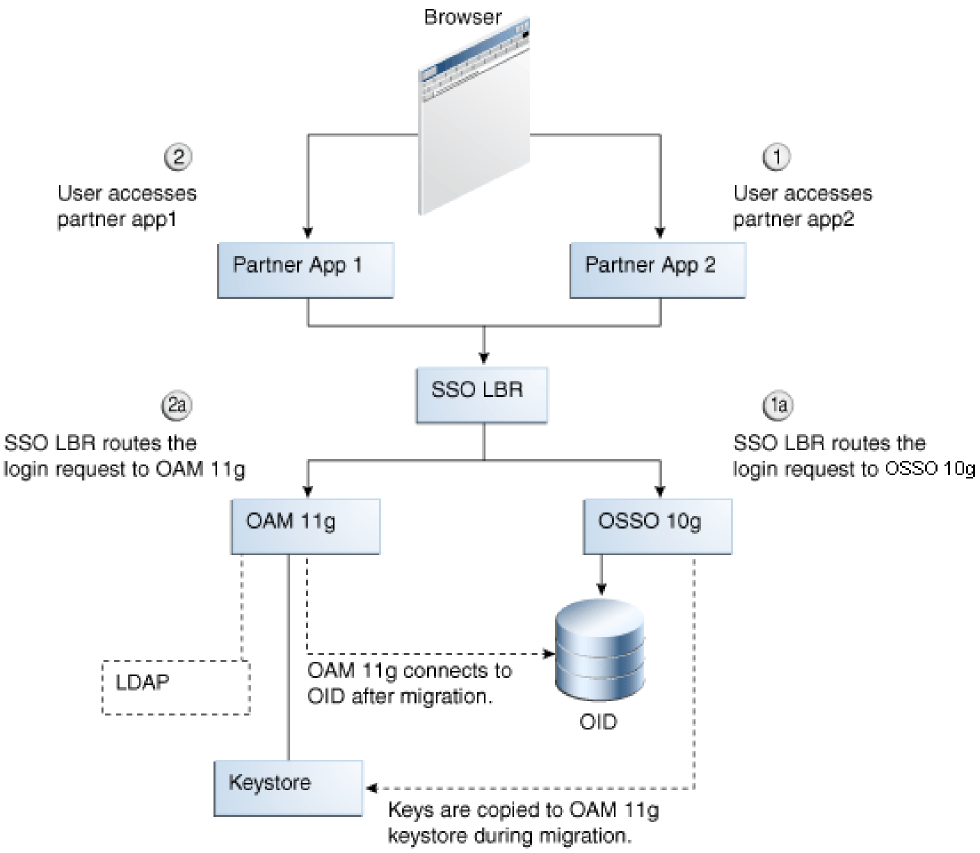
This section provides the following topics:

- About Post-Upgrade SSO
- About Post-Upgrade OSSO 10g Authentication

A.4.1 About Post-Upgrade SSO

Figure A–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 A–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.

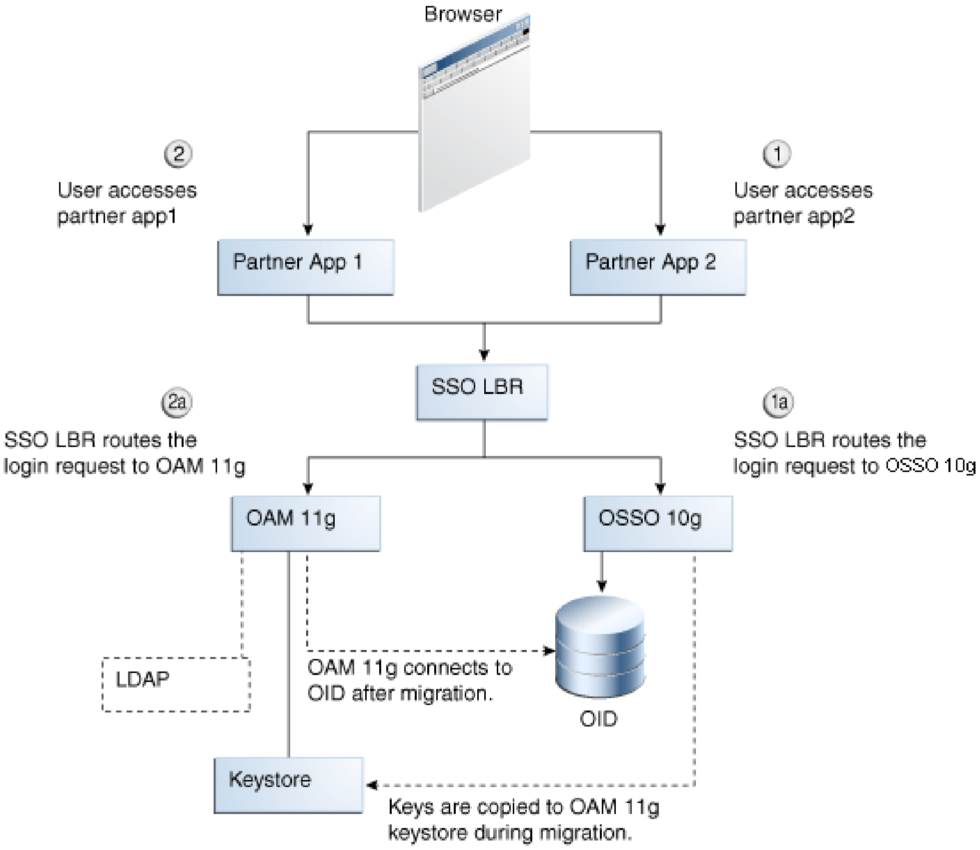
2. When the user accesses an application protected by OSSO 10g an OSSO 10g cookie is already set in the browser, and the user is not challenged for credentials and can access the application.

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.

A.4.2 About Post-Upgrade OSSO 10g Authentication

As shown in Figure A–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 A–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 displaying 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.

A.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

A.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 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 A-5

A.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 A–1 shows the applications protected by OracleAS 10g OSSO.

Table A–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

A.5.1.2 Policy Enforcement Agent Details

For each application in the OracleAS 10g SSO deployment (Table A–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 Console, you can locate each transferred OSSO Agent configuration under the System Configuration tab, Agents node in the navigation tree.

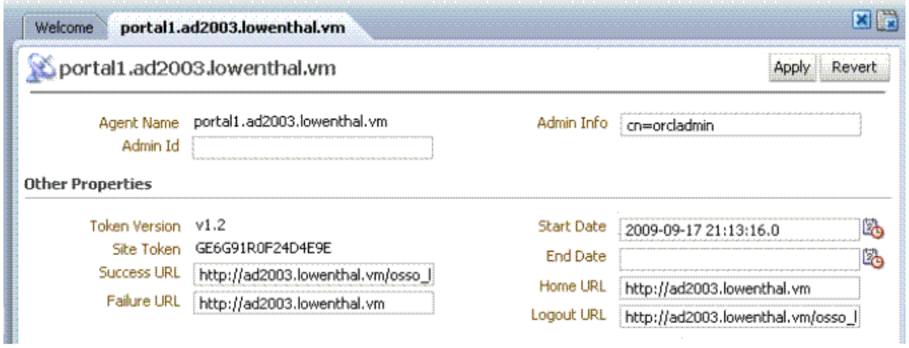
Figures in this topic illustrate the transferred OSSO Agent configurations for the applications identified in Table A–1. Each generated Agent configuration is named as the application it protects.

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 A–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 A–8 OSSO Agent Configuration Named for One Application



portal1.ad2003.lowenthal.vm

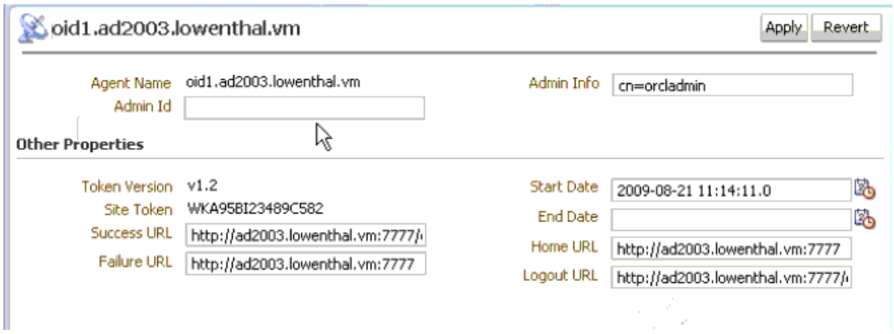
Agent Name portal1.ad2003.lowenthal.vm
Admin Id
Admin Info cn=orcladmin

Other Properties

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.vm
Logout URL http://ad2003.lowenthal.vm/osso_l

Figure A–9 shows the transferred configuration for a second application protected by OracleAS 10g SSO (OSSO).

Figure A–9 OSSO Agent Configuration Named for the Second Application



Agent Name oid1.ad2003.lowenthal.vm

Admin Id

Admin Info cn=orcladmin

Other Properties

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

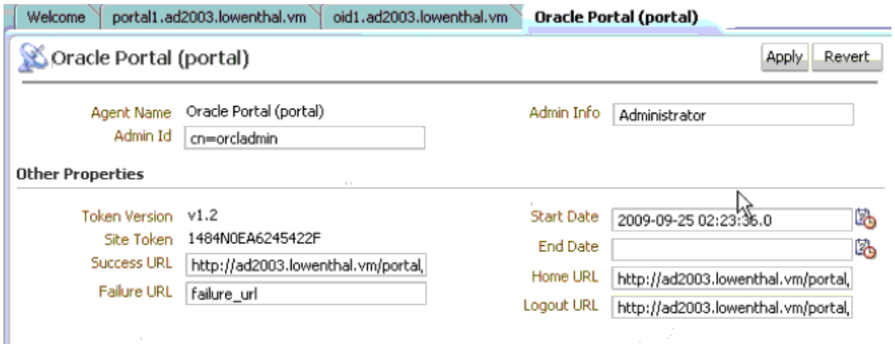
End Date

Home URL http://ad2003.lowenthal.vm:7777

Logout URL http://ad2003.lowenthal.vm:7777/

Figure A–10 shows the transferred OSSO Agent configuration for the third application protected with OracleAS 10g SSO (OSSO).

Figure A–10 OSSO Agent Configuration Named for the Third Application



Agent Name Oracle Portal (portal)

Admin Id cn=orcladmin

Admin Info Administrator

Other Properties

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

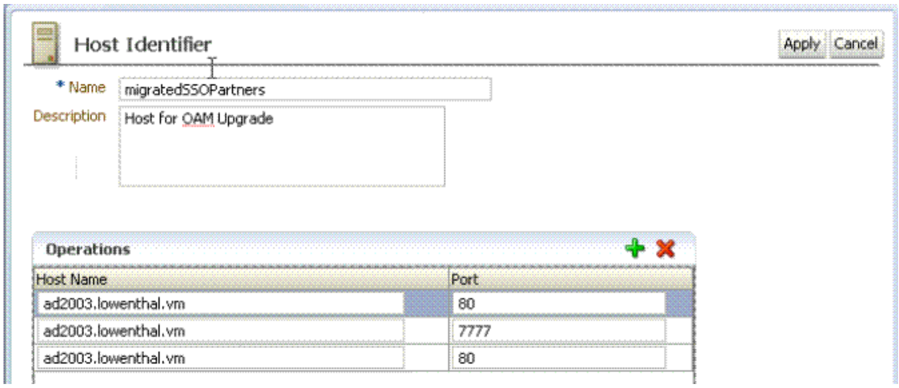
Home URL http://ad2003.lowenthal.vm/portal,

Logout URL http://ad2003.lowenthal.vm/portal,

A.5.1.3 Shared Components: Host Identifiers for migratedSSOPartners

Figure A–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 A–11 Host Identifier for migratedSSOPartners



* Name migratedSSOPartners

Description Host for OAM Upgrade

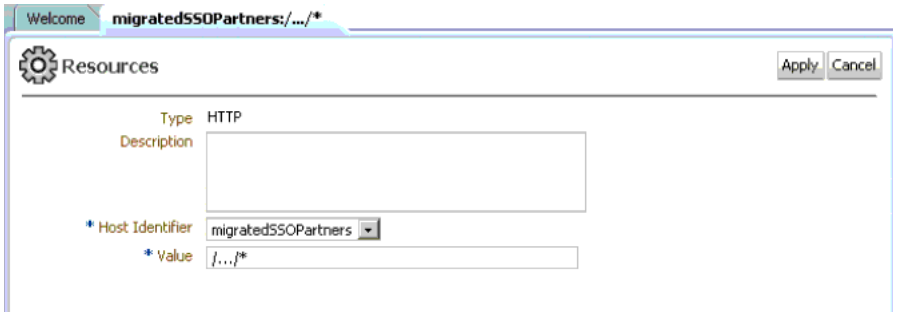
Host Name	Port
ad2003.lowenthal.vm	80
ad2003.lowenthal.vm	7777
ad2003.lowenthal.vm	80

A.5.1.4 Resources in the migratedSSOPartners Application Domain

Figure A–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 A–12 Resources in the migratedSSOPartners Application Domain



Type HTTP

Description

* Host Identifier migratedSSOPartners

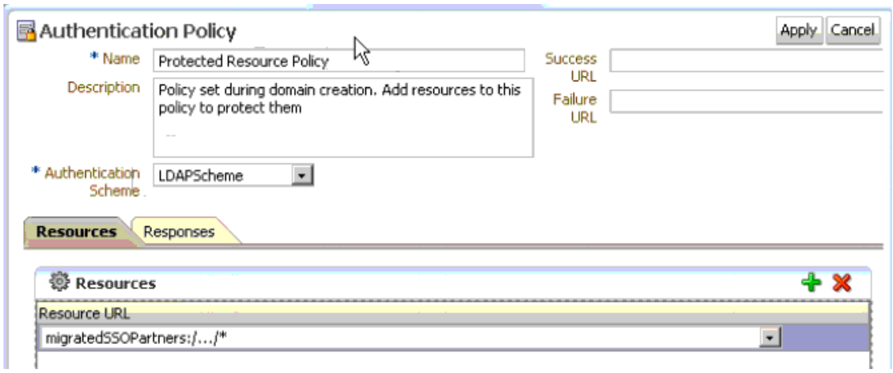
* Value /.../*

A.5.1.5 Authentication Policy in the migratedSSOPartners Application Domain

Figure A–13 illustrates the Authentication Policy, named `Protected Resource Policy`, for the Application Domain `migratedSSOPartners`.

The default OAM 11g authentication scheme is used: LDAPScheme.

Figure A–13 Authentication Policy for the Application Domain migratedSSOPartners



* Name Protected Resource Policy

Description Policy set during domain creation. Add resources to this policy to protect them

* Authentication Scheme LDAPScheme

Success URL

Failure URL

Resource URL

migratedSSOPartners:/.../*

A.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 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.

A.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.

B

Transitioning OAM 11g from a Test to a Production Environment

This chapter proides the procedure for creating an Oracle Access Manager 11g environment that is a replica of an existing, provisioned Oracle Access Manager 11g environment. You can use this approach for rolling out upgrades after testing. The information applies whether you are creating a replica of a test environment for a production environment or vice versa. The source environment is the one being replicated; the target environment is the destination.

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 to a new (or an existing) production environment.

B.1 Prerequisites

Install and configure target components, as described in the Oracle Fusion Middleware Installation Guide for Oracle Identity Management.

B.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 B–1 describes the types of deployments that customers might have within their enterprise. Deployment types might be named differently in your enterprise.

Table B–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, Oracle Access Manager 11g configuration data is stored in files while Oracle Access Manager 11g policy data is stored in a database.

On the policy configuration 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 Access Client) or an 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.

B.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 Migrating OSSO Information from One OAM Instance to Another
- About the WebLogic Scripting Tool Commands
- About Policy Conflict Resolution
- About Building a Dependency Tree for Each Application Domain

B.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 B–2.

Table B–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 B-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 B-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

B.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 B–3, " Full Replication"
- Table B–4, " Golden Template"
- Table B–5, " Delta-Replication"
- Table B–6, " Application Re-association"

Table B–3 describes full replication. By performing manual and automated tasks, you can replicate the OAM test source setup to an OAM production target.

Table B–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. Migrates the policies from the source such that both environments are identical. 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 B–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 B–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 Agents using either the Oracle Access Manager Console or remote registration commands for OAM. ■ 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 B–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 B–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 B–6 outlines requirements and tasks. By performing manual and automated tasks, you can re-associate source client applications to the target environment.

Table B–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.	

B.3.3 About Migrating OSSO Information from One OAM Instance to Another

Oracle Access Manager supports migrating partners across OAM Server instances. This is required in a GIT scenario wherein you must copy partner information from an internal to an external deployment.

When migrating a selected partner, retrieve the partner ID from the test system's oam-config.xml. For example, if the partner ID for the OSSO Agent with site name 'TEST_OSSO_AGENT2' is `998AF964144D39BC2F`, as shown here:

```
<Setting Name="998AF964144D39BC2F" Type="htf:map">
<Setting Name="AdminId" Type="xsd:string"></Setting>
<Setting Name="SiteName" Type="xsd:string">TEST_OSSO_AGENT2</Setting>
```

Then execute the following command from the WLST prompt:

```
exportSelectedPartners(pathTempOAMPartnerFile="<path where the temporary
file need to be generated>",partnersNameList="998AF964144D39BC2F")
```

B.3.4 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 (automatically) 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 B–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 B–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 >')

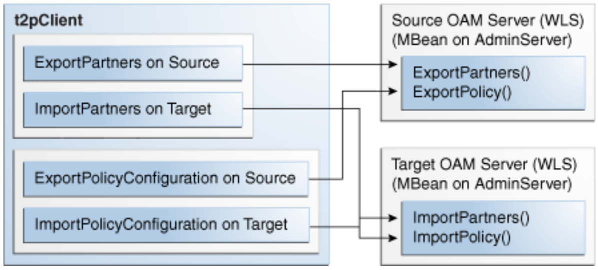
Importing decrypts the generated dump file using the key in the keystore and imports the dump file contents to the target OAM Server. You can import partners, policies, or policy differences, as described in Table B–8. Import commands are run on the target OAM Server.

Table B–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 B–1 illustrates the processing that occurs between the source and target systems.

Figure B–1 Source and Target processing



B.3.5 About Policy Conflict Resolution

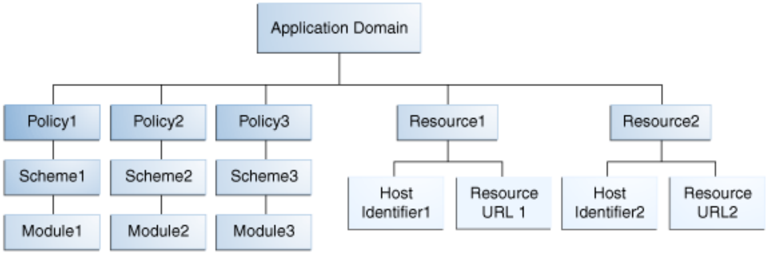
Policy conflicts are resolved automatically during processing. The source system is presumed to be the single source of truth during data migration. Any conflicts that are detected between the source system and the target system must be resolved during processing.

B.3.6 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 B–2.

Figure B–2 Dependency Tree for Each Application Domain



In the sample dependency tree shown in Figure B–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.

B.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

B.4.1 Choose the Method

Review details in "About Methods to Move from Test to Production" on page B-4 and choose the method that best suits your needs as described in:

- Table B–3, " Full Replication"
- Table B–4, " Golden Template"
- Table B–5, " Delta-Replication"
- Table B–6, " Application Re-association"

B.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 Access Clients) 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

B.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.

B.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.

B.4.5 Understanding Change Propagation

All changes are reflected in the Oracle Access Manager Console and are automatically propagated to every OAM Server in the cluster.

When you have a single OAM Server and a single Oracle Access Manager Console running on different computers, changes are propagated to the managed run-time OAM Server.

B.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.

B.5 Backup and Recovery Strategies

Oracle recommends that you back up data before transfer, and restore the backup after transfer if needed.

B.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)

B.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 B-3
- Appendix F, "Introduction to Custom WLST Commands for 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 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 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.

B.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 B-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 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 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 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.

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 15. 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 for Oracle Access Manager 11g

This appendix provides the information and steps required to ensure that OAM 11g Servers and clients (OAM Agents) can communicate securely across the Access Protocol channel. This chapter provides the following details:

- Prerequisites
- Introduction to Securing Communication Between OAM 11g Servers and Webgates
- Generating Client Keystores for OAM Tester in Cert Mode
- 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 clients (Webgates) means defining the transport security mode for the NAP (also known as the OAP) channel within the component registration page. The security level for the channel is specified as 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-13.

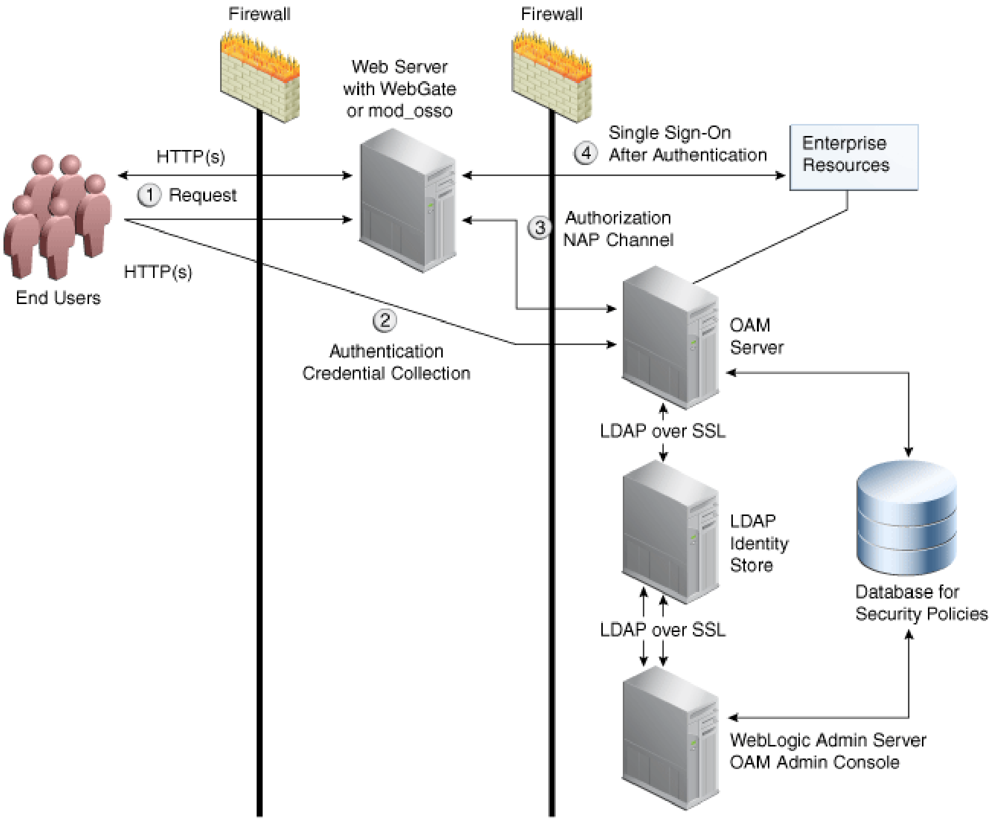
- Cert: Encrypted communication through SSL with a public key certificate issued by a trusted third-party certificate authority (CA).

  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-6.

  > **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



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.

The PEM (Privacy Enhanced Mail) format (BASE64-encoded ASCII) is preferred for private keys, digital certificates, and trusted certificate authorities (CAs). The preferred keystore format for OAM Servers is JCEKS and for OAM Clients is JKS (Java KeyStore) format. Oracle Access Manager components use X.509 digital certificates in DER (binary form of a certificate) format only.

For more information, see:

- About Certificates, Authorities, and Encryption Keys
- About Security Modes and X509Scheme Authentication
- About the Importcert Tool

E.2.1 About Certificates, Authorities, and Encryption Keys

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.

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.

See Also:

- "About Cert Mode Encryption and Files" on page E-6
- "About Simple Mode, Encryption, and Keys" on page E-13

E.2.2 About Security Modes and X509Scheme Authentication

Administrators must ensure that the OAM Server is reachable only over the transport specified in the OAM Server configuration. OAM Server configuration defines the end points for the Server and accounts for the deployment of load balancers or reverse proxies. When the OAM Server is reachable over both HTTP and HTTPS, all requests (over either transport) are accepted.

To allow the user to interact with the OAM Server (and logout) over SSL with non-X509 authentication schemes, the specified Server Port must not be configured to require CLIENT CERTS.

With the X509 authentication scheme (X509Scheme), the OAM Server SSL Port must differ from the Server Port, and must be configured to require Client Certificates. When X509Scheme is used, the X509 module is called after credential collection. X509Scheme requires the X509 challenge method and the X509 authentication module. The fully-qualified URL to the credential collector must be specified as the Challenge URL within X509Scheme. For example: https://<oam_server>:<ssl_port>/oam/CredCollectServlet/X509.

Note: If a relative Challenge URL is specified with X509Scheme, 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.

See Also: "Managing SSO Tokens and IP Validation" on page 8-4

E.2.3 About the Importcert Tool

Administrators use the Oracle-provided `importcert` tool for several different procedures related to keystores, keys, and certificates. Table E–1 provides the syntax for `importcert` commands.

Table E–1 importcert Command Syntax

Option	Description

keystore	Follow this command with the path to an existing (or new) keystore. For example:
	`/scratch/.oamkeystore`
	`or`
	`/scratch/clientKey.jks`

privatekeyfile	Follow this option with the path to your private key. For example: `/scratch/aaa_key.der`
signedcertfile	Follow this option with the path to your signed certificate. For example: `/scratch/aaa_cert.der`
alias	Follow this option with your keystore entry alias. Required with genkeystore.: `alias`

storetype	Follow this option with your keystore type. By default, the store type is JCEKS (OAM Server keystore). For example:
	Server keystore .oamkeystore, of type:
	`JCEKS`
	Client keystore/scratch/clientTrustStore.jks and /scratch/clientKey.jks can be used. Both are type:
	`JKS`

Table E–1 (Cont.) importcert Command Syntax

Option	Description
genkeystore	This flag is required for generating OAM client certificates. The client does not expose the alias and alias password parameters. However, importcert tool sets the keystore password as the alias password. Specify: `Yes or No` Yes imports the certificates in a new keystore. No imports certificates into an existing keystore.
Sample for OAM Server	`- java -cp importcert.jar oracle.security.am.common.tools.importcerts.Certificate Import -keystore <path to .oamkeystore> -privatekeyfile <path to aaa_key.der> -signedcertfile  <path to aaa_ cert.der> -alias oam.certmode -aliaspassword <password> -storetype <JCEKS> genkeystore <yes>` Enter the keystore password and alias password when prompted.
Sample for OAM Client See Also "Generating Client Keystores for OAM Tester in Cert Mode"	`- java -cp importcert.jar oracle.security.am.common.tools.importcerts.Certificate Import -keystore <path to clientkey.JKS> -privatekeyfile <path to aaa_key.der> -signedcertfile <path to aaa_cert.der> -storetype <JKS> genkeystore <yes>` Enter the keystore password when prompted.

E.3 Generating Client Keystores for OAM Tester in Cert Mode

This section is required to generate JKS keystores to be used with OAM Tester in Cert mode only. Otherwise, you can skip this section. This section describes how to use importcert commands to generate client keystores for OAM Tester in Cert mode to contain the imported trusted certificate chain.

> **See Also:** "About the Importcert Tool" on page E-4

To generate client keystores for OAM Tester in Cert mode

1. Use ImportCert tool to create JKS keystores (file name specified by -privatekeyfile and -signedcertfile). For example:

```
- java -cp importcert.jar
oracle.security.am.common.tools.importcerts.CertificateImport -keystore
<Keystore path> -privatekeyfile <Private key file> -signedcertfile  <Signed
certificate file> path -storetype <JKS> genkeystore <yes>
```

   Enter the keystore password when prompted.

2. Proceed as needed for your environment:

   - Configuring Cert Mode Communication for OAM 11g
   - Configuring Simple Mode Communication with OAM 11g

3. **Remove a Keystore**: Use the following command to remove the JKS keystore. For example:

```
keytool -delete -alias <alias> -keystore  <path to clientkey.JKS> -storetype
<JKS>
```

Enter the keystore password when prompted.

E.4 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 :
   - "Introduction to Securing Communication Between OAM 11g Servers and Webgates"
   - "About Cert Mode Encryption and Files"
2. Generating a Certificate Request and Private Key for OAM Server
3. Retrieving the OAM Keystore Alias and Password
4. Importing the Trusted, Signed Certificate Chain Into the Keystore
5. Adding Certificate Details to Access Manager Settings
6. Generating a Private Key and Certificate Request for Webgates
7. Updating Webgate to Use Certificates

E.4.1 About Cert Mode Encryption and Files

The certificate request for Webgate generates the request file aaa_req.pem, which you must send to a root CA that is trusted by the OAM Sever. The root CA returns the 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 (reserved name for Webgate key file, which cannot be changed)
- aaa_cert.pem (reserved name for Webgate certificate file, which cannot be changed)
- aaa_chain.pem (reserved name for CA Cert for Webgate side)

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. However, you cannot complete OAM deployment until the certificates are issued and installed.

If you choose Cert mode when registering Webgate as 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.4.2 Generating a Certificate Request and Private Key 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 -config openssl_silent_ohs11g.cnf
   ```

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".

E.4.3 Retrieving the OAM Keystore Alias and Password

Users with valid Administrator credentials can perform the following task to retrieve the alias of the certificate in the specified keystore to be used for authentication, and the password that is required to import a certificate.

To retrieve the OAM Keystore password

1. Confirm the Oracle Access Manager Console is running.
2. On the computer hosting the Oracle Access Manager 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 the Trusted, Signed Certificate Chain Into the Keystore".

E.4.4 Importing the Trusted, Signed Certificate Chain Into the Keystore

The Oracle-provided importcert tool is used to import existing private key, signed certificate (public key) files into the specified keystore format: JKS (client keystore format) or JCEKS (OAM Server keystore format; .oamkeystore for instance.).

The keystores associated with Oracle Access Manager 11g accepts only PKCS8 DER format certificates:

- If you have PEM format certificates signed by your certificate authority (CA), the following procedure describes how to convert and then import these using the `importcert` shipped with Oracle Access Manager 11g.
- If PEM format certificates are not available, create a certificate request and have it signed by your CA before beginning the following procedure.

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

To import the trusted certificate chain into the keystore

1. Locate the keytool for OAM 11g in the following path:

   ```
   $MW_HOME/jdk160_18/bin/keytool
   ```

2. Unzip importcert.zip and locate the Readme file in the following location:

   ```
   $ORACLE_IDM/oam/server/tools/importcert/README
   ```

3. **aaa_chain.pem**: Using a text editor, modify the aaa_chain.pem file to remove all data except that which is contained within the CERTIFICATE blocks, then save the file.

   ```
     ----BEGIN CERTIFICATE-----
         ...
   ```

```
    CERTIFICATE
    ...
 -----END CERTIFICATE-----
```

4. Import the trusted certificate chain using the following command with details for your environment. For example:

```
keytool -importcert -file aaa_chain.pem -trustcacerts -storepass <password>
-keystore <$ORACLE_HOME>\user_projects\domains\$DOMAIN\config\fmwconfig\
.oamkeystore -storetype JCEKS
```

5. When prompted to trust this certificate, type **`yes`**.

6. **aaa_cert.pem**:

   a. Edit aaa_certn.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. For example:

```
  ----BEGIN CERTIFICATE-----
      ...
      CERTIFICATE
      ...
   -----END CERTIFICATE-----
```

   b. Enter the following command to convert the signed certificate (aaa_cert.pem) to DER format using openSSL or any other tool. For example:

```
openssl x509 -in aaa_cert.pem -inform PEM -out aaa_cert.der -outform DER
```

7. **aaa_key.pem**:

   a. Edit aaa_key.pem 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. For example:

```
  ----BEGIN CERTIFICATE-----
      ...
      CERTIFICATE
      ...
   -----END CERTIFICATE-----
```

   b. Enter the following command to convert the private key (aaa_key.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
```

8. Import signed DER format certificates into the keystore. For example:

   a. Import aaa_key.der using the following command line arguments and details for your environment. For example:

   c:\Middleware\idm_home\oam\server\tools\importcert

```
- java -cp importcert.jar
oracle.security.am.common.tools.importcerts.CertificateImport
-keystore <> -privatekeyfile <path> -signedcertfile  <path>
-alias [ -storetype <> genkeystore <> -help]
```

> **Note:** Enter the key store password and alias password when prompted. On a Windows system, use a semicolon (;) instead of a colon (:) in the command line.

9. Proceed to "Adding Certificate Details to Access Manager Settings".

E.4.5 Adding Certificate Details to Access Manager Settings

After importing the certificates into the keystore, you must add the alias and password that you specified earlier into Access Manager settings configuration in Oracle Access Manager Console, as described here.

> **Note:** No explicit configuration is needed for Simple mode, which is provided out of the box for OAM 11g.

Prerequisites

Importing the Trusted, Signed Certificate Chain Into the Keystore

See Also:

- "Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security" on page 8-5

To add certificate details to Access Manager Settings

1. From the Oracle Access Manager Console, click the System Configuration tab.
2. From the System Configuration tab, Access Manager Settings section, open the Access Manager Settings page.
3. Expand the Access Protocol section of the page, if needed.
4. Fill in the alias and alias password details acquired in the previous procedure. For example:

   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 and Certificate Request for Webgates".

E.4.6 Generating a Private Key and Certificate Request 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 Webgate to Use Certificates".

E.4.7 Updating Webgate to Use Certificates

For all communication modes (Open, Simple, or Cert), the Agent registration should be updated from the Oracle Access Manager 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 Access Manager Settings

To update the communication mode in the Webgate Agent registration

1. From the System Configuration tab, Access Manager Settings section, expand the SSO Agents node, and expand OAM Agents.
2. On the Search page, define your criteria and open the desired agent registration, as described in "Searching for an OAM Agent Registration" on page 9-25.
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 and Certificate Request for Webgates".
5. Click Apply to submit the changes.
6. Copy your updated Webgate files as follows:

   11g Webgate:

   ObAccessClient.xml
   cwallet.sso (11g Webgate only)
   password.xml

- **From**: $IDM_DOMAIN_HOME/output/*AGENT_NAME*
- **To**: $OHS_INSTANCE_HOME/config/OHS/ohs2/webgate/config

10g Webgate: ObAccessClient.xml

- **From**: $WLS_DOMAIN_HOME/output/*AGENT_NAME*
- **To**: $*Webgate_install_dir*/oblix/lib

10g Webgate: password.xml

- **From**: $WLS_DOMAIN_HOME/output/*AGENT_NAME*
- **To**: $*Webgate_install_dir*/oblix/config

7. Copy the following files that were created when "Generating a Certificate Request and Private Key for OAM Server":

   11g Webgate:

   - From:

     aaa_key.pem: *Webgate11g_home*/webgate/ohs/tools/openssl
     aaa_cert.pem: The location where this was saved after receiving from CA
     aaa_chain.pem: The location where this was saved after receiving from CA

   - To: OHS_INSTANCE_HOME/config/OHS/ohs2/webgate/config

   10g Webgate:

   - From:

     aaa_key.pem: The location where the private key file was generated
     aaa_cert.pem: The location where this was saved after receiving from CA
     aaa_chain.pem: The location where this was saved after receiving from CA

   - To: $*Webgate_install_dir*/oblix/config

8. Restart the OAM Server and the Oracle HTTP Server instance.

E.5 Configuring Simple Mode Communication with OAM 11g

The transport security communication mode is chosen during OAM installation. In Simple mode, 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.

Note: Communication between the agent and server works when the Webgate mode matches (or is higher) than the OAM Server mode.

The agent mode can be higher but not lower. The highest level of security is Cert mode, the lowest is Open mode:

Cert mode   Simple mode   Open mode

This section provides the information you need to configure Simple mode communicat

Task overview: Configuring Simple mode communication with OAM 11g includes

1. Reviewing:
   - "About Simple Mode, Encryption, and Keys"
   - "About the Importcert Tool"
2. Retrieving the Global Passphrase for Simple Mode
3. Updating Webgate Registration for Simple Mode
4. Verifying Simple Mode Configuration

E.5.1 About Simple Mode, Encryption, and Keys

For Simple mode encryption, Oracle Access Manager includes a certificate authority with its own private key, which is installed across all Webgates and OAM Servers. During installation, the OAM server generates and saves the private-public keypair for the server. Similarly, for the OAM agent, an Oracle certificate authority is installed with the agent installation.

The installer generates a random global passphrase initially, which can be edited or viewed as needed. When an agent is registered in SIMPLE mode, the following client certificates are generated to be consumed by clients:

- aaa_key.pem: Contains private key
- aaa_cert.pem: Signed certificate
- password.xml: Contains the random global passphrase in obfuscated format

> **Note:** Changing the global passphrase requires reconfiguring all agents that are already configured in Simple mode.

E.5.2 Retrieving the Global Passphrase for Simple Mode

Oracle Access Manager generates a random global passphrase for Simple mode communication during installation. The following procedure describes how to retrieve this password.

To retrieve the random global passphrase for Simple mode communication

1. Ensure that the Oracle Access Manager Console is running.
2. On the computer hosting the Oracle Access Manager Console, locate the WebLogic Scripting Tool in the following path. For example:

   ```
   $Oracle_IDM/common/bin
   ```

   Where $Oracle_IDM represents the base Oracle Access Manager installation directory path; /common/bin is the path wherein the scripting tool is located.
3. Start the WebLogic scripting tool. For example, on a Unix system:

   ```
   ./ 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 [weblogic] :
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. View the global passphrase by entering the following command. For example:

   ```
   wls:/OAM_AC> displaySimpleModeGlobalPassphrase()
   ```

7. Proceed to Updating Webgate Registration for Simple Mode.

E.5.3 Updating 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.

See Also:

- "Registering and Managing OAM Agents Using the Console" on page 9-10

To update the Webgate registration for Simple mode

1. From the System Configuration tab, Access Manager Settings section, expand the SSO Agents node, and expand OAM Agents.
2. On the Search page, define your criteria and open the desired agent registration, as described in "Searching for an OAM Agent Registration" on page 9-25.
3. In the registration page, locate the Security options and click Simple.
4. Click Apply to submit the changes.
5. Copy the updated Webgate files as follows:

   11g Webgate:

   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/ohs2/webgate/config

   10g Webgate: ObAccessClient.xml

   - **From**: $WLS_DOMAIN_HOME/output/*AGENT_NAME*
   - **To**: $*Webgate_install_dir*/oblix/lib

   10g Webgate: password.xml

   - **From**: $WLS_DOMAIN_HOME/output/*AGENT_NAME*
   - **To**: $*Webgate_install_dir*/oblix/config

6. Copy the following files, as directed for your Webgate release:

   aaa_key.pem
   aaa_cert.pem

   11g Webgate:

   - **From**: $IDM_DOMAIN_HOME/output/*AGENT_NAME*
   - **To**: $OHS_INSTANCE_HOME/config/OHS/ohs2/webgate/webgate/config/simple

   10g Webgate:

   - From: $IDM_DOMAIN_HOME/output/*AGENT_NAME*
   - To: $*Webgate_install_dir*/oblix/config/simple

7. Restart the OAM Server and the Oracle HTTP Server instance.

E.5.4 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 Administrators

For certain administrative tasks, the WebLogic Scripting Tool (WLST) provides custom commands that can be used as an alternative to the Oracle Access Manager Console. This appendix provides an introduction to WLST commands for Administrators. Details for each command, however, are outside the scope of this book.

Sections in this appendix include:

- Prerequisites
- Introduction to WebLogic Scripting Tool Commands
- WLST Command Summary: Oracle Access Manager
- WLST Command Summary: Oracle Security Token Service
- Running WLST Commands

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

Custom WLST commands for OAM can be used for setting and managing OAM System Configuration only by Administrators.

The WebLogic Scripting Tool shares the same foundation layer with the Oracle Access Manager Console. WLST for Oracle Access Manager and Oracle Security Token Service 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 Command Summary: Oracle Access Manager

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

Table F–2 (Cont.) WLST Oracle Access Manager Commands

Use this command...	To...	Use with WLST...
deleteOamServer	Delete the named Oracle Access Manager Server configuration.	Online Offline
displayOssoAgent	Display OSSO Agent configuration details.	Online Offline
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
displayUserIdentityStore	Display a user identity store registration.	Online Offline
editUserIdentityStore	Edit a user identity store registration.	Online Offline
createUserIdentityStore	Create a user identity store registration. Note: The roleAppdAdmin is removed as a part of multi-store support. WLST is restricted and cannot set a store as the System Store.	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
exportPartners	Export the Oracle Access Manager partners from the source to the intermediate Oracle Access Manager file specified.	Online

Table F–2 (Cont.) WLST Oracle Access Manager Commands

Use this command...	To...	Use with WLST...
importPartners	Import the Oracle Access Manager partners from the intermediate Oracle Access Manager file specified.	Online
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
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
displaySimpleModeGlobalPassphrase	Displays the simple mode global passphrase in plain text from the system configuration.	Online
exportSelectedPartners	Exports selected OAM Partners to the intermediate OAM file specified.	Online
migrateArtifacts	Migrates artifacts based on the input artifact file.	Online
registerThirdPartyTAPPartner	Registers any third party as a Trusted Authentication Protocol (TAP) Partner.	Online

F.4 WLST Command Summary: Oracle Security Token Service

Use the WLST commands listed in Table F–3 to manage Oracle Security Token Service-related components.

See Also: The section on Oracle Security Token Service commands in the chapter "Infrastructure Security Custom WLST Commands" of the Oracle Fusion Middleware WebLogic Scripting Tool Command Reference.

Table F–3 WLST Commands Oracle Security Token Service

Use this command...	To...	Use with WLST...
putBooleanProperty putBooleanProperty("/stsglobal/ignoreunsupportedelements", "true")	Ignore unsupported WS-Trust elements present in the RST. Default: true Note: A value of false, returns an error if unsupported WS-Trust elements are present in the RST.	Online
Partner Commands		
getPartner	Retrieve a partner and print result.	Online
getAllRequesterPartners	Retrieve the names of Requester partners.	Online
getAllRelyingPartyPartners	Retrieve the names of all Relying Party partners.	Online
getAllIssuingAuthorityPartners	Retrieve the names of all Issuing Authority partners.	Online
isPartnerPresent	Query OSTS to determine whether or not the partner exists in the Partner store.	Online
createPartner	Create a new Partner entry.	Online
updatePartner	Update an existing Partner entry based on the provided information.	Online
deletePartner	Delete a partner entry.	Online
getPartnerUsernameTokenUsername	Retrieve the partner's username value.	Online
getPartnerUsernameTokenPassword	Retrieve the partner's password value.	Online
setPartnerUsernameTokenCredential	Set the username and password values of a partner entry.	Online
deletePartnerUsernameTokenCredential	Remove the username and password values from a partner entry.	Online
getPartnerSigningCert	Retrieve the Base64 encoded signing certificate for the partner.	Online
getPartnerEncryptionCert	Retrieve the Base64 encoded encryption certificate for the partner.	Online
setPartnerSigningCert	Upload the signing certificate to the partner entry.	Online
setPartnerEncryptionCert	Upload the encryption certificate to the partner entry.	Online
deletePartnerSigningCert	Remove the signing certificate from the partner entry.	Online Offline
deletePartnerEncryptionCert	Remove the encryption certificate from the partner entry.	Online Offline

Table F–3 (Cont.) WLST Commands Oracle Security Token Service

Use this command...	To...	Use with WLST...
getPartnerAllIdentityAttributes	Retrieve and display all Identity mapping attributes used to map a token to a requester partner.	Online Offline
getPartnerIdentityAttribute	Retrieve and display the identity mapping attribute.	Online Offline
setPartnerIdentityAttribute	Set the identity mapping attribute for a requester partner.	Online Offline
deletePartnerIdentityAttribute	Delete the identity mapping attribute for a requester partner.	Online Offline
Relying Party Partner Mapping Commands		
getAllWSPrefixAndPartnerMappings	Retrieve and display all WS Prefixes.	Online Offline
getWSPrefixAndPartnerMapping	Retrieve and display the Relying Party Partner mapped to the specified wsprefix parameter.	Online Offline
createWSPrefixAndPartnerMapping	Create a new WS Prefix mapping to a Relying Partner.	Online Offline
deleteWSPrefixAndPartnerMapping	Delete an existing WS Prefix mapping to a Relying Partner.	Online Offline
Partner Profiles Commands		
getAllPartnerProfiles	Retrieve the names of all the existing partner profiles.	Online
getPartnerProfile	Retrieve partner profile configuration data.	Online
createRequesterPartnerProfile	Create a new Requester Partner profile with default configuration data.	Online
createRelyingPartyPartnerProfile	Create a new Relying Party Partner profile with default configuration data.	Online
createIssuingAuthorityPartnerProfile	Create a new Issuing Authority Partner profile with default configuration data.	Online
deletePartnerProfile	Delete an existing partner profile.	Online
Issuance Template Commands		
getAllIssuanceTemplates	Retrieve the names of all the existing Issuance Templates.	Online Offline
getIssuanceTemplate	Retrieve configuration data of a specific Issuance Template.	Online
createIssuanceTemplate	Create a new Issuance Template with default configuration data.	Online
deleteIssuanceTemplate	Delete an existing Issuance Template.	Online Offline
Validation Template Commands		

Table F–3 (Cont.) WLST Commands Oracle Security Token Service

Use this command...	To...	Use with WLST...
getAllValidationTemplates	Retrieve the names of all the existing Validation Templates.	Online Offline
getValidationTemplate	Retrieve configuration data of a specific Validation Template.	Online Offline
createWSSValidationTemplate	Create a new WS Security Validation Template with default configuration data.	Online Offline
createWSTrustValidationTemplate	Create a new WS Trust Validation Template with default configuration data.	Online Offline
deleteValidationTemplate	Delete an existing Issuance Template.	Online Offline

F.5 Running WLST Commands

Administrators can use the following procedure as a guide for using WLST commands for Oracle Access Manager or Oracle Security Token Service operations. Included here are several operations:

> **See Also:** 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.5.1 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 ORACLE_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.

> **Note:** You can also use the "help('oamap')" and "help('oamapsso')" commands to display additional 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 chapter "Infrastructure Security Custom WLST Commands" of the Oracle Fusion Middleware WebLogic Scripting Tool Command Reference for full details.

F.5.2 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 8-8
- OAM_REQ cookie in Table 11–4, " 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-7.
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:

- Prerequisites
- Introduction to Oracle Access Manager 11g and IPv6
- Configuring IPv6: Separate Proxy for OAM 11g and Webgates

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 10g and 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

- Webgate 10g 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 WebgateWebgate10g or Webgate 11g 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

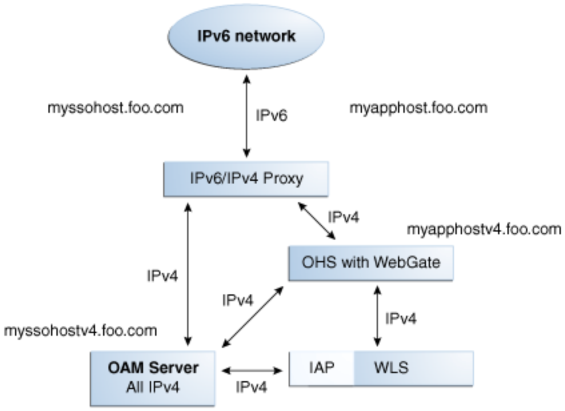
G.2.1 Configuring IPv6 with OAM 11g and Challenge Redirect

Figure G–1 illustrates configuration with a single IPv6 to IPv4 Proxy (host configured *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 Oracle Access Manager Console.

  See Also: "Single Sign-On with OAM 11g" on page 11-17

- 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 Oracle Access Manager 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.

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 10–4, " 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 of the OHS instance for the 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 Oracle Access Manager Console as shown here. See also, "About Remote Registration Request Files" on page 10-9.

a. Log in to Oracle Access Manager Console. For example:

```
    http://hostname:port/oamconsole
```

b. Click **System Configuration**, **Access Manager Settings, SSO Agents, OAM Agents**.

c. Find the agent and click its name in the Search Results table 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 Request Files"
- "About Virtual Web Hosting" on page 12-7

e. Click Apply.

4. Repeat for each Webgate and specify name of the Oracle HTTP Server Web server that is configured for this Webgate.

5. **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

Creating Deployment-Specific Pages

Oracle Application Server Single Sign-On provides a framework for integrating deployment-specific login, change password, and single sign-off pages with the single sign-on server. This means that you can tailor these pages to your UI look and feel and globalization requirements.

Oracle recommends that you use JavaServer (JSP) pages. Other Web technologies may provide inconsistent results. PLSQL pages are not supported. Sample pages are provided with the product. The Oracle Application Server Single Sign-On product ships with sample pages that are designed for testing with the Oracle Application Server.

This chapter contains the following topics:

- How the Single Sign-On Server Uses Deployment-Specific Pages
- How to Write Deployment-Specific Pages
- Page Error Codes
- Adding Globalization Support
- Guidelines for Deployment-Specific Pages
- Installing Deployment-Specific Pages
- Examples of Deployment-Specific Pages
- Adding an External Application

H.1 How the Single Sign-On Server Uses Deployment-Specific Pages

The process that enables single sign-on pages can be summarized as follows:

1. The user requests a partner application and is redirected to the single sign-on server.
2. If the user is not authenticated, the single sign-on server redirects the user to the sample login page or to a deployment-specific page. As part of the redirection, the server passes to the page the parameters contained in Table H–1 on page H-3.
3. The user submits the login page, passing the parameters contained in Table H–2 on page H-4 to the authentication URL:

   ```
   http://sso_host:sso_port/sso/auth
   ```

   or

   ```
   https://sso_host:sso_ssl_port/sso/auth
   ```

At least two of these parameters, `ssousername` and `password`, appear on the page as modifiable fields.

4. If the user's password is not set to expire soon, and the single sign-on server successfully verifies the user name and password, the server redirects the user to the success URL of the application. If authentication fails, the server redirects the user back to the login page and displays an error message.
5. If the user's password is set to expire soon, the single sign-on server presents the change password page instead of the login page. Again, if the server is configured to use a deployment-specific change password page, it redirects the user to the URL for this page, passing to the page the parameters contained in Table H–3 on page H-5.

   > **Note:** In step 5, the same conditions apply if the directory administrator forces the user to change the password, password expiration notwithstanding.

   The user submits the change password page, entering her old password, new password, and new password confirmation. The page passes the parameters contained in Table H–4 on page H-5 to the change password URL:

   ```
   http://sso_host:sso_port/sso/ChangePwdServlet
   ```

   or

   ```
   https://sso_host:sso_ssl_port/sso/ChangePwdServlet
   ```

   If an error occurs, the single sign-on server redirects the user to the change password page and displays an error message. See "Change Password Page Behavior" for a detailed discussion of conditions under which errors may occur.

   If the password change is successful, the user is redirected to the partner application URL that triggered the authentication request.
6. To finish the single sign-on session, the user clicks **Logout** in the partner application he or she is working in. This act calls application logout URLs in parallel, logging the user out from all accessed applications and ending the single sign-on session.
7. The user is redirected to the single sign-on server, which presents the single sign-off page. If the server is configured to use a deployment-specific page, it redirects the user to the URL for this page, passing to the page the parameters contained in Table H–5 on page H-6.
8. The user can click **Return** on the single sign-off page to return to the application from which logout was initiated.

   > **Note:** The change password page can be used to change a password only when the password is about to expire. The UI for Oracle Delegated Administration Services can be used for this purpose at any time. See "Change Password Page Behavior" for more about this topic.

H.1.1 Change Password Page Behavior

Users who try to log in when their passwords have expired or are about to expire experience the following server behavior:

H.1.1.1 Password Has Expired

Users are shown the password expiry screen. They must contact the directory administrator to have the password reset.

H.1.1.2 Password Is About to Expire

Users are shown an error message on the login page. They have the option of cancelling the page or changing their passwords. In either case, authentication proceeds in the same manner as it does when the change password page is not thrown.

H.1.1.3 Grace Login Is in Force

If a grace login period has been configured in the directory, users are presented the change password page after their passwords have expired. They have the option of cancelling the page or changing their passwords. In either case, the authentication sequence is the same as it is for users with valid passwords.

H.1.1.4 Force Change Password

This feature prompts users to change their password after it has been reset by an administrator. You enable force change password by setting the `pwdMustChange` attribute in the directory entry `cn=pwdpolicyentry,cn=common,cn=products,cn=OracleContext,dc=`*`default_identity_management_realm`*. You can use the command-line tool `ldapmodify` for this purpose. The value `TRUE` enables this feature. `FALSE` disables it. See the chapter about password policies inOracle Internet Directory Administrator's Guideto learn how to run the tool.

H.2 How to Write Deployment-Specific Pages

The URLs for login, change password, and single sign-off pages must accept the parameters described in the tables that follow if these pages are to function properly.

This section contains the following topics:

- Login Page Parameters
- Forgot My Password
- Change Password Page Parameters
- Single Sign-Off Page Parameters

H.2.1 Login Page Parameters

The URL for the login page must accept the parameters listed in Table H–1 on page H-3.

Table H–1 Login Page Parameters Submitted to the Page by the Single Sign-On Server

Parameter	Description
`site2pstoretoken`	Contains the authentication request token for login processing.
`ssousername`	Contains the username.
`p_error_code`	Contains the error code in the form of a string. Passed when an error occurs during authentication.
`p_cancel_url`	Contains the URL to redirect to if the user clicks **Cancel**—if such a button exists on the login page. This URL points to the home URL of the partner application from which login was initiated.

Table H–1 (Cont.) Login Page Parameters Submitted to the Page by the Single Sign-On

Parameter	Description
locale	User's language preference (optional). Must be in ISO format. For example, French is fr-fr. For more about this parameter, see "Adding Globalization Support".

The login page must pass the parameters listed in Table H–2 to the authentication URL:

```
http://sso_host:sso_port/sso/auth
```

Table H–2 Login Page Parameters Submitted by the Page to the Single Sign-On Server

Parameter	Description
site2pstoretoken	Contains the redirect URL information for login processing.
ssousername	Contains the username. Must be UTF-8 encoded.
password	Contains the password entered by the user. Must be UTF-8 encoded.
subscribername	The subscriber nickname when realms are enabled. Must be UTF-8 encoded. **Note**: This field is required on the login page only when multiple realms are enabled in the single sign-on server.
locale	User's language preference (optional). Must be in ISO format. For example, French is fr-fr. For more about this parameter, see "Adding Globalization Support".
v	Contains the page version. Recommended but optional. If the parameter is passed, its value must be v1.4.

The login page must have at least two fields: a text field with the parameter name ssousername and a password field with the parameter name password. The values are submitted to the authentication URL. The login page must also include site2pstoretoken as a hidden parameter. It must submit this parameter to the login URL.

In addition to submitting these parameters, the login page is responsible for displaying appropriate error messages, as specified by p_error_code, redirecting to p_cancel_url if the user clicks **Cancel**.

H.2.2 Forgot My Password

When building your login page, you may want to configure it with a link that enables users to reset their passwords. This URL can go either to the home page for Oracle Delegated Administration Services or to the **Forgot My Password** link within Oracle Delegated Administration Services. Users who click the **Forgot My Password** link are challenged with a question. They must successfully answer this question before their password is reset.

Oracle Delegated Administration Services is generally available on the same computer as OracleAS Single Sign-On at a URL of the following form:

```
http://sso_host:sso_port/oiddas/
```

To learn how the Forgot My Password link is used to reset passwords, see the chapter about the Oracle Internet Directory Self-Service Console in Oracle Identity Management Guide to Delegated Administration.

H.2.3 Change Password Page Parameters

The URL for the change password page must accept the parameters listed in Table H–3.

> **Note:** In a GIT deployment, when a partner logout flow requires query parameters in the p_done_url, the parameters must be URL encoded such that the Oracle Access Manager logout servlet does not interpret them as being Oracle Access Manager parameters but elements of the single p_done_url.

Table H–3 Change Password Parameters Submitted to the Page

Parameter	Description
`p_username`	Contains the user name to be displayed somewhere on the page.
`p_subscribername`	The subscriber nickname when hosting is enabled. Note: This field is required on the login page.
`p_error_code`	Contains the error code, in the form of a string, if an error occurred in the prior attempt to change the password.
`p_done_url`	Contains the URL of the appropriate page to return to after the password is saved.
`site2pstoretoken`	Contains the `site2pstoretoken` that is required by the /sso/auth login URL if the password has expired or is about to expire.
`p_pwd_is_exp`	Contains the flag value indicating whether the password has expired or is about to expire. The value can be either `WARN` or `FORCE`. See Table H–8 for the associated error codes.
`locale`	User's language preference (optional). Must be in ISO format. For example, French is `fr-fr`. For more about this parameter, see "Adding Globalization Support".

The change password page must pass the parameters listed in Table H–4 to the change password URL:

```
http://sso_host:sso_port/sso/ChangePwdServlet
```

Table H–4 Change Password Page Parameters Submitted by the Page

Parameter	Description
`p_username`	Contains the user name to be displayed somewhere on the page. Should be posted as a hidden field by the change password page. Must be UTF-8 encoded.
`p_old_password`	Contains the user's old password. Must be UTF-8 encoded.
`p_new_password`	Contains the user's new password. Must be UTF-8 encoded.
`p_new_password_confirm`	Contains the confirmation of the user's new password. Must be UTF-8 encoded.
`p_done_url`	Contains the URL of the appropriate page to return to after the password is saved.
`p_pwd_is_exp`	Contains the flag value indicating whether the password has expired or is about to expire. The value can be either `WARN` or `FORCE`. See Table H–8 for the associated error codes.
`site2pstoretoken`	Contains the redirect URL information for login processing.

Table H–4 (Cont.) Change Password Page Parameters Submitted by the Page

Parameter	Description
`p_action`	Commits changes. The values must be either `OK` (commit) or `CANCEL` (ignore).
`p_subscribername`	Contains the user name to be displayed somewhere on the page.
`p_request`	Protected URL requested by the user.
`locale`	User's language preference (optional). Must be in ISO format. Example: French is `fr-fr`. See "Adding Globalization Support".

The change password page must have at least three password fields: `p_old_password`, `p_new_password`, and `p_new_password_confirm`. The page should submit these fields to the change password URL.

The page should also submit `p_done_url` as a hidden parameter to the change password URL. In addition, it should display error messages according to the value of `p_error_code`.

H.2.4 Single Sign-Off Page Parameters

The URL for the single sign-off page must accept the parameters listed in Table H–5.

Table H–5 Parameters Submitted to the Single Sign-Off Page

Parameter	Description
`p_app_name[1. . .n]`	Contains the application name to be displayed on the page. The variable `n` stands for the number of partner applications participating in single sign-off.
`p_app_logout_url[1. . .n]`	Contains the application logout URL. The variable `n` stands for the number of partner applications participating in single sign-off.
`p_done_url`	Contains the return URL. This URL returns users to the application from which they initiated logout.
`locale`	User's language preference in ISO format. Sent only if the user does not pass the same value during login.

H.2.5 External Application Login Page Parameters

The URL for external application login pages must accept the parameters listed in Table H–6.

Table H–6 Parameters Submitted to the External Application Login Page

Parameter	Description
`ID`	The external application ID. This ID appears on the Administer External Application page. For each external application configured in OracleAS Single Sign-On, a unique ID is generated. This ID is the primary key in the external application-related tables. The external application login page must pass this ID back to the application.

Table H–6 (Cont.) Parameters Submitted to the External Application Login Page

Parameter	Description
p_app_name	Contains the application name to be displayed on the page. This is the name for the external application that was provided when the application was configured in OracleAS Single Sign-On.
extappfieldname1..9	Extra field names. Each external application can have up to nine extra fields associated with it. These fields can be visible or non-visible. Visible fields appear on the external application login page, and the user can change the default value of the field. The non-visible field values are also submitted to the external application, but the user cannot change their value. For example, for an application that has login, password, and locale fields, you can add an field named LO with a value of FR. See "Adding an External Application" on page H-14 for details.
extappfieldvalue1..9	Extra field values.
extappfielddisplay1..9	Extra field visibility (true or false). Determines if the field is visible to the user and is modifiable (true) or if the field has a fixed value (false).
mode	This parameter may or may not be passed to the custom login page. If it is passed to the login page, it must be submitted with its value set to modify. This indicates to the single sign-on application controller that the external application login page has been called from the portal to update the user's credentials in the database.
p_error_code	Contains the error code, in the form of a string, if an error occurred in the prior attempt to change the password.
done	The URL to which the response must be redirected after the user's credentials have been updated. This is used with modify mode.

This page must be able to submit the parameters shown in Table H–7 using the POST method to the external application login controller:

Table H–7 Parameters the External Application Login Page Submits to the Application

Parameter	Description
ID	The external application ID.
p_app_username	Contains the user name of the person logging in to the application.
p_app_pwd	The password that the user submits.
p_remember_credentials	Flag that indicates to the external application login controller if the application user name and password must be saved to the database.
extappfieldname1..9	Extra field names. See Table H–6 for details.
extappfieldvalue1..9	Extra field values.
extappfielddisplay1..9	Extra field visibility (true/false). See Table H–6 for details.
p_change_password	A true/false flag that indicates to the external application login controller if the mode is set to change password.

Table H–7 (Cont.) Parameters the External Application Login Page Submits to the

Parameter	Description
mode	This parameter may or may not be passed to the custom login page. If it is passed to the login page, it must be submitted with its value set to `modify`. This indicates to the single sign-on application controller that the external application login page has been called from the portal to update the user's credentials in the database.
done	The URL to which the response must be redirected after the user's credentials have been updated.

H.3 Page Error Codes

URLs for login and change password pages must accept the process errors described in the tables that follow if these pages are to function properly.

H.3.1 Login Page Error Codes

The login page must process the error codes listed in Table H–8.

Table H–8 Login Page Error Codes

Value of p_error_code	Corresponding message and description
acct_lock_err	Description: The user has committed too many login failures. Message: "Your account is locked. Please notify the system administrator."
pwd_exp_err	Description: The user's password has already expired. Message: "Your password has expired. Please contact the administrator to reset it."
null_uname_pwd_err	Description: The user left the user name field blank. Message: "You must enter a valid user name."
auth_fail_exception	Description: Authentication has failed. Message: "Authentication failed. Please try again."
null_password_err	Description: The user left the password field blank. Message: "You must enter your logon password."
sso_forced_auth	Description: The application requires authentication. Message: "The application you are trying to access requires you to sign in again even if you have signed in previously."
unexpected_exception	Description: An unexpected error occurred during authentication. Message: "An unexpected error occurred. Please try again."
unexp_err	Description: Unexpected error. "Unexpected Error. Please contact Administrator."

Table H–8 (Cont.) Login Page Error Codes

Value of p_error_code	Corresponding message and description
`internal_server_err`	Description: Internal server error report. Message: "Internal Server Error. Please contact Administrator."
`internal_server_try_again_err`	Description: Internal server error report with "try again" prompt. Message: "Internal Server Error. Please retry the operation."
`internal_server_try_later_err`	Description: Internal server error report with "try later" prompt. Message: "Internal Server Error. Please try the operation later."
`gito_err`	Description: Inactivity timeout. User must log in again. Message: "Your Single Sign_on session has expired. For your security, your session expires after some duration of inactivity. Please sign in again."
`cert_auth_err`	Description: Certificate sign-on has failed. User should check that the certificate is valid or should contact the administrator. Message: "Certificate-based sign in failed. Please ensure that you have a valid certificate or contact the administrator."
`session_exp_error`	Desscription: Single sign-on session time limit reached. Message: "Your Single Sign-On session has expired. For your security, your session expires after the specified amount of time. Please sign in again."
`userid_mismatch`	Description: The user ID presented during a forced authentication does not match the user ID in the current single sign-on session. Message: "The user name submitted for authentication does not match the user name present in the existing Single Sign-On session."

H.3.2 Post-Login Messages

The messages listed in Table H–9 appear after the user authenticates. They are processed by the login page but may appear with the password change page.

Table H–9 Post-Login Messages

Value of p_error_code	Corresponding message and description
`pwd_expiry_warn_err`	Description: The user's password will expire soon. Message: "Your password is about to expire. Please change it."

Table H–9 (Cont.) Post-Login Messages

Value of p_error_code	Corresponding message and description
`pwd_force_change_err`	Description: The user's password has expired and the user is expected to change it. Message: "You must change your password before you can continue."
`pwd_grace_login_err`	Description: The user's password has expired, but a grace period is in effect for resetting it. Message: "Your password has expired. You are now in the grace login period. Please change your password."

H.3.3 Change Password Page Error Codes

The change password page must process the error codes listed in Table H–10.

Table H–10 Change Password Page Error Codes

Value of p_error_code	Corresponding Error
`confirm_pwd_fail_txt`	The old and the new password do not match.
`null_new_pwd_err`	The user did not enter a new password.
`null_old_pwd_err`	The user did not enter the old password.
`pwd_expiry_warn_err`	The password is about to expire.
`pwd_force_change_err`	The password must be changed before the user can proceed.
`pwd_grace_login_err`	The password has expired, but a grace login is permitted.
`account_deactivated_err`	The user account is disabled.
`acct_lock_err`	The user account is locked.
`pwd_illegal_value`	The password contains an illegal value.
`pwd_in_history_err`	The password is in the password history.
`pwd_min_length_err`	The password does not meet the minimum length requirement.
`pwd_numeric`	The password does not meet the numeric character requirement.

H.3.4 Change External Application Login Page Error Codes

The external application login page must process the error codes listed in Table H–11.

Table H–11 External Application Login Page Error Codes

Value of p_error_code	Corresponding Error
`eapp_name_null`	The user ID is missing.
`eapp_pwd_null`	The password is missing.
`ext_app_not_found`	The external application cannot be identified.

H.4 Adding Globalization Support

The OracleAS Single Sign-On framework enables you to globalize deployment-specific pages to fit the needs of your deployment. When deciding what language to display the page in, you can adopt different strategies. Two strategies are presented in the following sections.

For a complete list of the language codes supported, see Appendix A in Oracle Application Server Globalization Guide at the following URL:

http://www.oracle.com/technology/documentation/index.html

From the landing page for the URL, click a link for the OracleAS Single-Sign on documentation, then click the View Library link to view the library for the appropriate release.

H.4.1 Deciding What Language to Display the Page In

This section explains how to use either the HTTP Accept-Language header or deployment page logic to choose a language to display.

H.4.1.1 Use the Accept-Language Header to Determine the Page

Browsers enable end users to decide the language (locale) they would like to view their Web content in. The browser sends the language that the user chooses to the server in the form of the HTTP Accept-Language header. The logic of the deployment-specific page must examine this header and render the page accordingly. When it receives this page, the single sign-on server takes note of the header value for Accept-Language and sends it to partner applications when it propagates the user's identity. Note that, although many partner applications enable users to override this header, the single sign-off page appears in the language established at sign-on. The net effect is a consistent session language for all partner applications.

The Accept-Language header is the preferred mechanism for determining the language preference. A major benefit of this approach is that end users have typically already set their language preference while browsing other Web sites. The result is browsing consistency between these pages and single sign-on pages.

H.4.1.2 Use Page Logic to Determine the Language

Although Oracle recommends the approach described in the preceding section, you may choose to implement globalization based on mechanisms that extend or override the language preference set in the browser. You may, for instance, do one of the following:

- Display a list of languages on the login page and allow the user to select from this list. As a convenience to the user, you can make this selection persistent by setting a persistent cookie.
- Render the page in one, fixed language. This method is appropriate when you know that the user population is monolingual.
- Obtain language preferences from a centralized application repository or a directory. A centralized store for user and system preferences and configuration data is ideal for storing language preferences.

If you use page logic to set language preferences, the page must propagate this information to the single sign-on server. The server must propagate this information to partner applications. The net result is a consistent globalization experience for the user. Your page must pass the language in ISO-639 format, using the `locale` parameter (Table H–2) in the login form. A number of sites contain a full list of ISO-639 two-letter language codes. Here is one of them:

```
http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt
```

Here is a site that contains a full list of ISO-3166 two-letter country codes:

```
http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html
```

Note: In the event that the `locale` parameter is passed to the single sign-on server (Table H–1), the parameter value is sent to mod_osso. mod_osso prefixes this value to the HTTP Accept-Language header before passing the header to partner applications.

H.4.2 Rendering the Page

Once it determines the end-user's locale, the deployment-specific page must use the corresponding translation strings to render the page. To learn how to store and retrieve these strings, see the chapter about locale awareness inOracle Application Server Globalization Guide. You may also want to consult standard documents about Java development. Here are two links:

- Java Internationalization Guide:

  ```
  http://java.sun.com/j2se/1.4.2/docs/guide/intl/index.html
  ```

- General link for Java documentation:

  ```
  http://java.sun.com/j2se/1.4.2/docs
  ```

H.5 Guidelines for Deployment-Specific Pages

When implementing deployment-specific pages, observe the following guidelines:

- Oracle recommends that login and change password pages be protected by SSL.
- The login and change password pages must code against cross-site scripting attacks.
- The login and change password pages must have auto-fill and caching set to `off`. This prevents user credentials from being saved or cached in the browser. Here is an example of the `AutoComplete` tag:

  ```
  <FORM NAME="foo" AutoComplete="off" METHOD="POST" ACTION="bar">
  ```

- Oracle recommends that you configure your login page to display a banner that warns against unauthorized access. You may, for example, want to use the following text or a variant thereof:

  ```
  Unauthorized use of this site is prohibited and may subject you to civil and
  criminal prosecution.
  ```
- Deploy the login and change password pages on the computer that hosts the single sign-on server. This makes it easier to detect false versions of these pages.

H.6 Installing Deployment-Specific Pages

Use the `policy.properties` file to install deployment-specific login and change password pages.

H.6.1 Using policy.properties to Install Login, Single Sign-Off, and Change Password Pages

You can configure login, single sign-off, and change password pages.

To install your own login, single sign-off, and change password pages:

1. Edit the parameters in *ORACLE_HOME*/sso/conf/policy.properties. Substitute the paths to your login, logout, and password change pages for the values shown in the example:

   ```
   #Deployment login page link
   loginPageUrl = /sso/pages/login.jsp
   logoutPageUrl = /sso/pages/logout.jsp

   #Deployment change password page link
   chgPasswordPageUrl = /sso/pages/password.jsp
   ```

2. Restart the single sign-on server:

   ```
   ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
   ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
   ```

H.6.2 Using policy.properties to Install Wireless Login and Change Password Pages

OracleAS Wireless has its own framework for integrating deployment-specific wireless login and change password pages. The procedure for installing these pages is similar to that used to install standard pages (section immediately preceding).

To install wireless login and change password pages:

1. Open *ORACLE_HOME*/sso/conf/policy.properties.
2. Edit or add the following parameters:

   ```
   #Wireless login page link
   wirelessLoginPageUrl = wireless_login_page_url
   wirelessChgPasswordPageUrl = change_password_page_URL
   ```

3. Restart the single sign-on server:

   ```
   ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
   ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
   ```

H.6.3 Using policy.properties to Install External Application Login Pages

You can configure login pages that are presented to users when they log in to third-party applications.

To configure login pages for third-party applications:

1. Be sure these pages accept the page parameters and page error codes discussed in "Login Page Parameters" on page H-3 and "Page Error Codes" on page H-8.
2. After configuring the login pages, edit the extAppLoginPageUrl parameter in *ORACLE_HOME*/sso/conf/policy.properties, substituting the path to your login page for the path shown in the following example:

   ```
   extAppLoginPageUrl = /sso/pages/ealogin.jsp
   ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
   ```

3. Optionally, you can configure the application page
4. Restart the single sign-on server:

   ```
   ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
   ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
   ```

H.7 Examples of Deployment-Specific Pages

The `ipassample.jar` file contains the files `login-ex.jsp`, `password-ex.jsp`, and `signoff-ex.jsp`. You may customize these to suit your deployment. If you want to use these files. Use this command to extract the file:

```
ORACLE_HOME/jdk/bin/jar -xvf ORACLE_HOME/sso/lib/ipassample.jar
```

H.7.1 Using Custom Classes

In general, customized deployment-specific pages must operate with the current versions of component classes in use by OC4J_SECURITY. If your custom application needs to use a different version of a given class, you must deploy that class in a separate OC4J instance and *not* in the OC4J_SECURITY instance.

For example, if your deployment requires the use of custom log4j classes that conflict with the versions in use by OC4J_SECURITY, start a separate OC4J_SECURITY instance that uses a local log4j `jar` file containing the custom classes.

WARNING: Replacing the classes used by OC4J_SECURITY with custom versions may render OracleAS Single Sign-On or other Oracle Application Server components unusable.

H.8 Adding an External Application

From the Single Sign-On Server Administration page, clicking the Administer External Applications link, then clicking Add External Application link takes you to the Add External Applications page. This page contains the following headings and fields:

Table H–12 External Application Login

Field	Description
Application Name	Enter a name that identifies the external application. This is the default name for the external application.
Login URL	Enter the URL to which the HTML login page for the external application is submitted for authentication. This, for example, is the login URL for Yahoo! Mail: `http://login.yahoo.com/config/login?6p4f5s403j3h0`
Username/ID Field Name	Enter the term that identifies the user name or user ID field of the HTML login form for the application. You find this term by viewing the HTML source of the form. (See the example after the steps immediately following). This field is not applicable if you are using basic authentication.
Password Field Name	Enter the term that identifies the password field of the HTML login form for the application. You find this term by viewing the HTML source of the form. (See the example after the steps immediately following). This field is not applicable if you are using basic authentication.

Table H–13 Authentication Method

Field	Description
Type of Authentication Use	Use the pulldown menu to select the form submission method for the application. This method specifies how message data is sent by the browser. You find this term by viewing the HTML source for the login form. Select one of the following three methods: POST: Posts data to the single sign-on server and submits login credentials within the body of the form. GET: Presents a page request to a server, submitting the login credentials as part of the login URL. Basic authentication: Submits the login credentials in the application URL, which is protected by HTTP basic authentication. Notes: ■ Basic authentication uses pop-up windows, which by default are blocked by Windows XP, service pack 2. If you use this service pack, make sure that you reconfigure browser settings to display the window for the single sign-on login page. Use the pop-up blocker item in the Tools menu of Internet Explorer. Other browsers and browser plugins are able to block popups. Mozilla is one of these. Make sure that these do not block the single sign-on login page. ■ If you use Internet Explorer 5.0 or a later version, basic authentication may not work with external applications. This version of Internet Explorer includes Microsoft MS04-004 Cumulative Security Update (832894). See this link for a workaround: `http://support.microsoft.com`

Table H–14 Additional Fields

Field	Description
Field Name	Enter the name of any additional fields on the HTML login form that may require user input to log in. This field is not applicable if you are using basic authentication.
Field Value	Enter a default value for a corresponding field name value, if applicable. This field is not applicable if you are using basic authentication.

To add an external application:

1. From the Administer External Applications page, select **Add External Application**.

   The Add External Applications page appears.

2. In the **External Application Login** field, enter the name of the external application and the URL to which the HTML login form is submitted. If you are using basic authentication, enter the protected URL.

3. If the application uses HTTP POST or HTTP GET authentication, in the **User Name/ID Field Name** field, enter the term that identifies the user name or user ID field of the HTML login form.

   You can find the name by viewing the HTML source of the login form.

   If the application uses the basic authentication method, the **User Name/ID Field Name** field should be empty.

4. If the application uses HTTP POST or HTTP GET authentication, in the **Password Field Name** field, enter the term that identifies the password field of the application.

   See the HTML source of the login form.

If the application uses the basic authentication method, the **Password Field Name** field should be empty.

5. In the **Additional Fields** field, enter the name and default values for any additional fields on the HTML login form that may require user input.

   If the application uses the basic authentication method, these fields should be empty.

6. Select the **Display to User** check box to allow the default value of an additional field to be changed by the user on the HTML login form.
7. Click **OK**. The new external application appears under the **Edit/Delete External Application** heading on the Administer External Applications page, along with the other external applications.
8. Click the application link to test the login.

The following example shows the source of the values that are used for Yahoo! Mail.

```
<form method=post action="http://login.yahoo.com/config/login?6p4f5s403j3h0"
autocomplete=off name=a>
...
<td><input name=login size=20 maxlength=32></td>
....
<td><input name=passwd type=password size=20 maxlength=32></td>
...
<input type=checkbox name=".persistent" value="Y" >Remember my ID & password
...
</form>
```

The source provides values for the following:

- Login URL :

  `http://login.yahoo.com/config/login?6p4f5s403j3h0`

- Username/ID Field Name: `login`
- Password Field Name: `passwd`
- Type of Authentication Used: `POST`
- Field Name: `.persistent Y`
- Field Value: `[off]`

Note: If you change the host name of the AS middle tier, you must manually update the Login URL field for external applications on this middle tier. You do this on the Edit External Applications page, described in the next section.

I

Troubleshooting

This chapter provides troubleshooting tips.

- Introduction to OAM 11g Troubleshooting
- Oracle Access Manager Console Inconsistent State
- AdminServer Won't Start if the Wrong Java Path Given with WebLogic Server Installation
- Agent Naming Not Unique
- Application URL Requirements
- Authentication Issues
- Authorization Issues
- Cannot Access Authentication LDAP or Database
- Cannot Find Configuration
- Could Not Find Partial Trigger
- Denial of Service Attacks
- Deployments with Freshly Installed OAM 10g Webgates
- Disabling Windows Challenge/Response Authentication on IIS Web Servers
- Changing UserIdentityStore1 Type Can Lock Out Administrators
- IIS Web Server Issues
- jps Logger Class Instantiation Warning is Logged on Authentication
- Languages and Translation
- Login Failure for a Protected Page
- OAM Metric Persistence Timer IllegalStateException: SafeCluster
- Partial Cluster Failure and Intermittent Login and Logout Failures
- Registration Issues
- Rowkey does not have any primary key attributes Error
- SELinux Issues
- Session Issues
- SSL versus Open Communication
- Start Up Issues

- Synchronizing OAM Server Clocks
- Using Coherence
- Validation Errors
- Web Server Issues
- Windows Native Authentication

I.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

I.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

I.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 22 (alternatively, in Chapter 26).
5. Try to access Oracle Access Manager 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 Oracle Access Manager Console.
10. Correct the issue based on the requirements in your environment.

I.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 14.
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 10 (this should fail).
4. Startup OAM AdminServer.

I.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.

I.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 22 (alternatively, in Chapter 26).

I.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"

I.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 Oracle Access Manager Console to edit policies, and then observe errors in the AdminServer log file.

I.2 Oracle Access Manager Console Inconsistent State

Problem

Administrators performing updates concurrently will result in an inconsistent state within the system configuration of the Oracle Access Manager Console.

Cause

Concurrent configuration updates are not supported.

Solution

Only one administrator should be allowed to modify the system configuration at any given time.

I.3 AdminServer Won't Start if the Wrong Java Path Given with WebLogic Server Installation

WebLogic Server (wls1035_generic) installation is successful on Windows 64-bit with 32-bit Java (jdk1.6.0_24). When setup.exe is executed you must provide the path of the 64-bit java (jdk1.6.0_23) to successfully launch the install shield.

If you provide the 32-bit Java (jdk1.6.0_24) path, the install shield is not launched. However, if you execute config.cmd from \Middleware\Oracle_IDM1\common\bin, by default 32-bit Java (jdk1.6.0_24) path is used, but after successful installation Oracle Access Manager installation, you cannot start AdminServer.

On Windows host, the path to 32-bit JAVA_HOME (c:\program files (x86)\java\jdkxxx) is not correctly handled by the startWeblogic.cmd. Replacing SUN_JAVA_HOME to use the path with the shorter name (c:\progra~2\java\jdkxxxx) works fine.

On Windows, the shorter names can be seen by executing "dir /X".

Alternatively, you can set windows cmd shell variable JAVA_HOME to path with shorter name and execute startWeblogic.cmd within that. For example:

>set JAVA_HOME=c:\progra~2\java\jdkXXX

>startweblogic.cmd

I.4 Agent Naming Not Unique

A unique identifying name for each Agent registration is preferred. However:

- If the Agent Name exists, no error occurs and the registration does not fail. Instead, Oracle Access Manager creates the policies if they are not already in place.
- If the host identifier exists, the unique Agent Base URL is added to the existing host identifier and registration proceeds.

I.5 Application URL Requirements

The number of characters allowed in a URL are based on browser version.

Ensure that your applications do not use URLs that exceed the length that Oracle Access Manager and the browser can handle.

I.6 Authentication Issues

This section provides the following information:

- Anonymous Authentication Issues
- X.509 Protected Resource and Single Sign Off

I.6.1 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.

I.6.2 X.509 Protected Resource and Single Sign Off

Problem

Single Sign Off might not work after accessing the resource with X.509 authentication. When the user is logged out with the logout URL and tries to access the resource in the same browser, authentication might not occur. Instead, the user should be asked for authentication using the certificate pop up.

This can occur with any Agent type.

Solution

After executing the logout URL, click on Clear SSL State from the browser as follows, and the access the X.509-protected resource:

From the browser window, open the Tools menu, click Internet Options, choose Content, and then Clear SSL state.

I.7 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.

I.8 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.

I.9 Cannot Find Configuration

I.9.1 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.

I.10 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 Configuration tab or a host identifier node) and then you click any of other nodes in the navigation tree.

This does not block functionality.

I.11 Denial of Service Attacks

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. One common method of attack involves saturating the target (victim) machine with external communication requests, such that it cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable.

Denial of service attacks are classified into Authenticated and Unauthenticated Requests, and further classified as:

- NAP Requests
- HTTP Requests

Authenticated NAP Requests

For Authenticated NAP Requests, the OAM Server maintains a counter in the session and limits the number of retries. Despite this, after redirecting the user to an error page the user can repeat the cycle. This needlessly consumes server resources and can lead to OAM Server overloading.

> **Note:** To avoid OAM Server overloading with Authenticated NAP Requests, use relevant WebLogic overload configuration settings. These ensure that the server does not crash under load. However, this does not differentiate legitimate users from malicious users.

Authenticated HTTP Requests

You can handle a flood of HTTP Authenticated requests with a combination of WebLogic overload configuration and mod_security module settings.

Unauthenticated NAP Requests

Unauthenticated NAP Requests are handled by the WebLogic MDB pool throttling. This imits the number of NAP Requests that are forwarded to the OAM Server.

Again, this does not differentiate legitimate users from malicious ones.

Unauthenticated HTTP Requests

Configuring the mod_security module for the OHS server that front-ends the OAM Server enables rejection of malicious requests (unauthenticated HTTP Requests).

For more information, see:

- Protecting the OAM Server from Crashing Under Load
- Compensating for Network Latency
- Protecting OAM Servers from a Flood of HTTP Requests

I.11.1 Protecting the OAM Server from Crashing Under Load

If the number of requests to the OAM Server unexpectedly increases beyond what the server can handle, it could crash.

To limit the number of requests to the OAM Server:

1. In the WebLogic Console, use the Message Driven Bean pool to restrict the number of NAP requests to the OAM Server.

   MDBeans pull NAP requests from the Server queue and deliver NAP requests to the Server for processing. Limiting the number of MDBean instances helps control the number of requests that are processed at a given time.

2. In the WebLogic Console, configure the number of WebLogic worker threads that can be used (to restrict the number of requests to the OAM Server).

   MDBeans pull NAP requests from the server queue and deliver NAP requests to the Server for processing. Limiting the number of MDBean instances helps control the number of requests that are processed at a given time.

3. In the WebLogic Console, configure the number of WebLogic worker threads that can be used (to restrict the number of requests to the OAM Server).

   See the topic on Thread Management in the guide to Oracle Fusion Middleware Performance and Tuning for Oracle WebLogic Server.

4. In the WebLogic Console, specify a maximum incoming request size, complete message timeout, and set the number of file descriptors, to optimize performance as described in following topics in the Oracle Fusion Middleware Performance and Tuning for Oracle WebLogic Server:

   - Tuning Message Size
   - Tuning Complete Message Timeout
   - Tuning Number of File Descriptors

I.11.2 Compensating for Network Latency

Consider the scenario where Webgate sends an authentication request to the OAM Server. After successful credential collection and validation, the OAM Server creates the session and the relevant cookies (OAM_ID, ObSSOCookie). However, due to network latency, the response times out by the time the OAM Server sends it to the Webgate which triggers Webgate to re-send the authentication request to the Server. The OAM Server recognizes the session, then recreates the ObSSOCookie, and sends the response to the agent.

If the network latency persists, the cycle continues in an infinite loop between the Server and the Webgate. The user is neither asked to login again nor presented with an error message.

I.11.3 Protecting OAM Servers from a Flood of HTTP Requests

ModSecurity is a Web application firewall (WAF) that can be deployed as part of the existing Apache-based Web server infrastructure. This module can be plugged into the OHS Server that front-ends the OAM Server. In this way, Mod_security module protects the OAM Server from denial of service attacks.

A flexible rule engine is at the heart of ModSecurity. It implements the ModSecurity Rule Language, a specialized programming language designed to work with HTTP transaction data. A new configuration directive uses the httpd-guardian script to monitor for Denial of Service (DoS) attacks. By default httpd-guardian defends against clients that send more than 120 requests in a minute, or more than 360 requests in five minutes.

> **See Also:**
>
> http://www.modsecurity.org/documentation/modsecurity-apache/2.5.12/html-multipage/configuration-directives.html#N10689

To protect from a flood of HTTP Requests

1. Add the mod_security module to the OHS Server that front-ends the OAM Server.
2. In the OHS Server configuration, set the configuration directive to use the httpd-guardian script to monitor for Denial of Service (DoS) attacks.

   Syntax:

   ```
   SecGuardianLog |/path/to/httpd-guardian
   ```

   Example:

   ```
   SecGuardianLog |/usr/local/apache/bin/httpd-guardian
   ```

I.12 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 15, "Configuring Centralized Logout for OAM 11g"

I.12.1 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

I.12.2 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 15-7.
- Change the OAM 10g Webgate's httpd.conf to remove the following lines:

```
<LocationMatch "/oamsso/*">
Satisfy any
</LocationMatch>
```

- From the Oracle Access Manager Console, confirm that the LogoutUrls parameter (/oamsso/logout.html) is configured for this Webgate

I.13 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

I.13.1 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 22, "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.

I.13.2 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 22, "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.

I.13.3 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>
```

I.14 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.

I.15 Changing UserIdentityStore1 Type Can Lock Out Administrators

An Identity Store that is designated as the System Store should not be edited to change the store type (from Embedded LDAP to OID, for instance) nor the connection URLs.

If you do need to change the Identity Store that is designated as the System Store should not be edited to change the store type, Oracle recommends that you create a new Identity Store and then edit that registration to mark it as your System Store.

I.16 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

I.16.1 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.

I.16.2 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.

I.16.3 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
```

I.16.4 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.

I.16.5 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.

I.17 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.

I.18 Languages and Translation

This section provides the following topics:

-
-
-

I.18.1 Automatically Generated Descriptions Are Not Translated

The automatically generated Description for each component of the Policy Configuration tab is not translated. This is expected and enables Administrators to change the Description to whatever they require. Following such a change, translation by Oracle is not possible.

I.18.2 Locales, Languages, and Oracle Access Manager Console Login Page

When the browser locale is not supported, the Oracle Access Manager Console Login page shows as server locale. It should fall back to English. This is the expected behavior:

- If the client Locale is not supported, Oracle Access Manager falls back to the server locale.
- If the server locale is not supported, Oracle Access Manager falls back to English.

When users select an unsupported language and come to the Oracle Access Manager SSO page, it shows as server locale (German, for example). However, after logging in, all the pages are displayed as English.

To fall back to English

Disable the Oracle Access Manager SSO page and the original Oracle Access Manager login page also falls back to English.

I.18.3 Console Looks Messy

The Oracle Access Manager Console displays policies and resources oddly when the input Fusion Applications configuration file for remote registration is not in UTF-8 format or when the OAM Server is not started in UTF-8 locale (en_US.utf8, for instance).

Be sure to use UTF-8 encoding if creating a Fusion Applications configuration file for the remote registration tool, oamreg, to generate authentication policies and protected resources. Also, be sure to start OAM Server in UTF-8 locale machines. Otherwise, the Oracle Access Manager Console might display policies and resources oddly following successful inband registration.

I.19 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.

I.20 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 Oracle Access Manager 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 6-4.

I.21 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.

I.22 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 13-16.

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.

I.23 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.

I.24 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 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
```

I.25 Session Issues

This section provides the following details:

- Session Impersonation Not Enabled by Default
- Sessions with Oracle Access Manager with Oracle Identity Federation

I.25.1 Session Impersonation Not Enabled by Default

Session impersonation is not enabled by default. You can update the value in oam-config.xml and update the version of oam-config.xml to automatically propagate the ImpersonationConfig status to all managed servers without a restart.

To enable Session Impersonation

1. Back up *DOMAIN_HOME*/config/fmwconfig/oam-config.xml.
2. Set `ImpersonationConfig` to `true`:

```
<Setting Name="ImpersonationConfig" Type="htf:map">
     <Setting Name="EnableImpersonation" Type="xsd:boolean">false</Setting>
</Setting>
```

3. **Configuration Version:** Increment the `Version xsd:integer` as shown in the next to last line of this example (existing value (25, here) + 1):

   Example:

```
<Setting Name="Version" Type="xsd:integer">
  <Setting xmlns="http://www.w3.org/2001/XMLSchema"
    Name="NGAMConfiguration" Type="htf:map:>
  <Setting Name="ProductRelease" Type="xsd:string">11.1.1.3</Setting>
    <Setting Name="Version" Type="xsd:integer">25</Setting>
</Setting>
```

4. Save oam-config.xml.

I.25.2 Sessions with Oracle Access Manager with Oracle Identity Federation

When Oracle Access Manager 11g is integrated with Oracle Identity Federation, and you use the Oracle Access Manager Session Management function to clear the session, only the Oracle Access Manager session is cleared (not the Oracle Identity Federation session).

I.26 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.

I.27 Start Up Issues

Problem: AdminServer Startup (or Remote Registration Tool Failure) on AIX Platforms

AdminServer start up fails with following message:

```
"java.net.SocketException:
No buffer space available".
```

Configration for the number of AIX file descriptors set for the operating sytem is substantially high (ulimit) resulting in a buffer overflow that causes remote registration failure with the following message:

The ulimit value is application dependent and applies exclusively to application program data and the application stack. The default number of open files setting (2000) is typically sufficient for most applications. If the value is too low, errors might occur when opening files or establishing connections. Because this value limits the number of file descriptors that a server process might open, a value that is too low prevents optimum performance. For the AIX operating system, the default setting is 2000.

Solution

Increasing the ulimit file descriptor limits might improve performance. Increasing some of the other limits might be needed depending on your application.

1. Log in as root.
2. Perform the following steps to change the open file limit to 10,000 files:
   a. Open the command window.
   b. Locate and edit /etc/security/limits file to add the following lines to the user account on which the AdminServer process runs:

   ```
   nofiles =  10000
   nofiles_hard = 10000
   ```

   c. Save the file and restart AIX.
3. In a command window, decrease the TCP_TIMEWAIT interval with the following command to set the state to 15 seconds (which allows TCP to release closed connections faster and increases the number of available resources for open connections).

   ```
   /usr/sbin/no -o tcp_timewait =1
   ```

4. Tune the following parameters to 256k, as shown:

   ```
   no -a |grep space
         tcp_recvspace = 262144
         tcp_sendspace = 262144
         udp_recvspace = 262144
         udp_sendspace = 262144
   ```

5. Tune the following parameters as indicated here:

   ```
   no -o rfc1323=1
   no -o sb_max=4194304
   ```

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 IAMSuiteAgent agent is started before the OAM Server.

The IAMSuiteAgent 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 IAMSuiteAgent falls back and the WebLogic container handles protection (including login), if it is configured.

I.28 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.

I.29 Using Coherence

Oracle Access Manager 11g uses Oracle Coherence to replicate session states within a distributed installation. Coherence is used to communicate state changes between the Oracle Access Manager Console and OAM Servers.

Consider the following 2 distributed deployment topologies. Coherence relies on User Datagram Protocol (UDP) for cluster discovery and heartbeat. If a firewall exists between certain components of OAM 11g, then the corresponding UDP ports used by Coherence must be open. Otherwise, OAM 11g might not work correctly.

For example, the UDP ports used by Coherence must be opened as follows:

- The Oracle Access Manager Console is deployed within the intranet, and OAM Servers are deployed in the DMZ. In this case, the UDP ports used by Coherence must be opened on the firewall between the DMZ and the intranet.
- The Oracle Access Manager Console and OAM Servers are deployed in different security zones of the DMZ, with firewalls between any two adjacent zones. In this case, the UDP ports used by Coherence must be opened on the firewall between the adjacent security zones, where one or more instances of Oracle Access Manager Console and OAM Servers run.

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 7-4.

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

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 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.

I.30 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.

I.31 Web Server Issues

The following issues with Web servers may arise:

- 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

I.31.1 Server Fails on an Apache Web Server

Symptom: You are running an Apache Web server, and an OAM 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 OAM Server.

The additional instances create even more connections, which exceed the number of connections by the OAM Server.

Solution: Reduce the number of `MinSpareServers`, `MaxSpareServers`, `StartServers`, and `MaxClients` parameters.

Go to the OAM Server's configuration directory and open the `http.d` configuration file.

Recommended parameter settings:

- MinSpareServers 1
- MaxSpareServers 5
- StartServers 3
- MaxClients 5

I.31.2 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

I.31.3 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 28-28.

I.31.4 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 I-19

I.31.5 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.

I.31.6 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 31, "Configuring Lotus Domino Web Servers for 10g Webgates".

I.31.7 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.

I.31.8 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 .

I.31.9 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 2.4.19.9

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
   ```

I.31.10 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.

I.31.11 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
```

I.31.12 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
```

I.31.13 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

I.31.14 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.

I.32 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 Oracle Access Manager Console, review the identity store configuration: System Configuration, Data Sources, User Identity Store.
2. Confirm the LDAP store settings point to Active Directory.