

Oracle® Fusion Middleware

Administering Oracle Access Management



12c (12.2.1.4.0)

E95911-21

July 2023

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Fusion Middleware Administering Oracle Access Management, 12c (12.2.1.4.0)

E95911-21

Copyright © 2019, 2023, Oracle and/or its affiliates.

Primary Author: Panendra Puttachar

Contributors: Madhu Martin, Kamal Narayan, Vamsi Motokuru

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	ix
Documentation Accessibility	ix
Related Documents	ix
Conventions	lxi

What's New in This Guide?

Updates in October 2022 Documentation Refresh for 12c Release 2 (12.2.1.4.0)	lxii
Updates in October 2021 Documentation Refresh for 12c Release 2 (12.2.1.4.0)	lxii
Updates in April 2021 Documentation Refresh for 12c Release 2 (12.2.1.4.0)	lxiii
Updates in January 2021 Documentation Refresh for 12c Release 2 (12.2.1.4.0)	lxiv
Updates in October 2020 Documentation Refresh for 12c Release 2 (12.2.1.4.0)	lxv
Updates in July 2020 Documentation Refresh for 12c Release 2 (12.2.1.4.0)	lxv
Updates in April 2020 Documentation Refresh for 12c Release 2 (12.2.1.4.0)	lxvi
Features of Oracle Access Management 12c Release 2 (12.2.1.4.0)	lxvi
Features Not Supported in Access Manager	lxviii

Part I Introduction to Oracle Access Management

1 Introducing Oracle Access Management

1.1 Understanding Oracle Access Management Services	1-1
1.2 Understanding Oracle Access Management Access Manager	1-3
1.2.1 About Components in Access Manager	1-4
1.2.2 Understanding Access Manager Deployments	1-5
1.3 System Requirements and Certification	1-7
1.4 Understanding Oracle Access Management Installation	1-7
1.4.1 About Oracle Access Management Installation	1-7
1.4.2 About Oracle Access Management Post-Installation Tasks	1-8

2 Getting Started with Oracle Access Management

2.1	Starting and Stopping Servers in Your Deployment	2-1
2.1.1	Starting Node Manager	2-1
2.1.2	Removing OAM Server from WLS 12c defaultCoherenceCluster	2-2
2.1.3	Starting and Stopping WebLogic AdminServer	2-2
2.1.4	Starting and Stopping Managed WebLogic Servers and Access Manager Servers	2-3
2.2	About Oracle Access Management Administrators	2-4
2.3	Oracle Access Management Console and the Policy Manager Console	2-5
2.4	Understanding the Oracle Access Management Console	2-5
2.4.1	System Launch Pad	2-6
2.4.2	Access Manager Launch Pad	2-7
2.4.3	Agents Launch Pad	2-7
2.4.4	Help Desk Launch Pad	2-8
2.5	About Logging Into the Oracle Access Management Console	2-8
2.5.1	Logging Into The Oracle Access Management Console	2-9
2.5.2	Logging Into the Secure Oracle Access Management Console (HTTPS)	2-9
2.6	Using the Oracle Access Management Console	2-10
2.6.1	Logging Out of the Oracle Access Management Console	2-11
2.6.2	Accessing Online Help in the Oracle Access Management Console	2-11
2.6.3	SSO Agent Search Page	2-11
2.7	Command-Line Tools for Configuration	2-12
2.8	Logging, Auditing, Reporting, and Monitoring Performance	2-13
2.9	Configuring Oracle Access Management Login Options	2-13
2.9.1	Administering the Forgot Password URL	2-14
2.9.1.1	Setting a Forgot Password URL	2-14
2.9.1.2	Retrieving a Forgot Password URL	2-14
2.9.2	Choosing a User Login Language	2-14
2.9.2.1	User Login Language Code	2-15
2.9.2.2	Selecting A Language for Oracle Access Management Login	2-16
2.9.2.3	Language Preference Cookie	2-17
2.9.2.4	Propagating Language Preference and Application Integration	2-17
2.9.3	Understanding Persistent Login	2-18
2.9.3.1	Enabling Persistent Login	2-19
2.9.3.2	Troubleshooting Persistent Login	2-21

Part II Managing Common and System Configurations

3 Managing Common Services and Certificate Validation

3.1	Configuration Options in Oracle Access Management Console	3-1
3.2	Available Services of the Common Configuration Section	3-3
3.2.1	Enabling or Disabling Available Services	3-4
3.3	Common Settings	3-4
3.3.1	Managing Common Settings	3-5
3.4	Certificate Validation and Revocation	3-6
3.4.1	Enabling the Certificate Revocation List Functionality	3-7
3.4.2	Enabling OCSP Certificate Validation	3-8
3.4.3	Enabling CRL Distribution Point Extensions	3-9
3.4.4	Additional OCSP Configurations	3-9
3.4.4.1	Configuring Multiple OCSP Responders	3-10
3.5	WLST updateHTTPProxyConfig	3-11
3.6	WLST configureOAMOSCSPCertValidation	3-12

4 Delegating Administration

4.1	Understanding Administrator Roles	4-1
4.2	About Delegating the Identity Store	4-2
4.3	Assigning Roles Using the Administration Console	4-3
4.4	Understanding the Container Security Framework and MBeans	4-3
4.5	Using the Remote Registration Utility	4-4
4.6	About Auditing Reports	4-5

5 Managing Data Sources

5.1	Data Sources for Oracle Access Management	5-1
5.1.1	Updating OAM Configuration	5-3
5.1.2	About the Default LDAP Group	5-4
5.2	Registering and Managing User Identity Stores	5-4
5.2.1	Understanding User Identity Stores	5-5
5.2.2	About using the System Store for User Identities	5-6
5.2.2.1	Using the System Store for User Identities	5-6
5.2.3	About Using Multiple Identity Stores	5-7
5.2.3.1	Components of Oracle Access Management that use Identity Stores	5-8
5.2.4	User Identity Store Settings	5-9
5.2.5	Registering a New User Identity Store	5-15
5.2.6	Viewing or Editing a User Identity Store Registration	5-16
5.2.7	Deleting a User Identity Store Registration	5-16
5.3	Managing the Identity Directory Service User Identity Stores	5-17
5.3.1	Identity Directory Services	5-17

5.3.2	Creating an Identity Directory Service Profile	5-19
5.3.3	Editing or Deleting an Identity Directory Service Profile	5-23
5.3.4	Creating a Form-fill Application Identity Directory Service Profile	5-26
5.3.5	Understanding the Pre-Configured Identity Directory Service Profile	5-27
5.3.6	Creating an Identity Directory Service Repository	5-27
5.4	Managing Administrator Roles	5-29
5.4.1	Understanding Administrator Roles	5-29
5.4.2	Defining and Removing Administrator Roles	5-30
5.5	Managing the Policy and Session Database	5-31
5.5.1	About the Database Store for Policy, Password Management, and Sessions	5-32
5.5.2	About Database Deployment	5-32
5.5.3	Configuring a Separate Database for Access Manager Sessions	5-33
5.6	Introduction to Oracle Access Management Keystores	5-34
5.6.1	Access Manager Security Keys and the Embedded Java Keystore	5-34
5.6.2	Access Manager Keystores	5-35
5.6.3	Identity Federation Keystore	5-36
5.7	Integrating a Supported LDAP Directory with Oracle Access Manager	5-37

6 Managing Server Registration

6.1	Before You Register	6-1
6.2	Understanding OAM Server Registration and Management	6-1
6.2.1	About Individual OAM Server Registrations	6-2
6.2.2	About Communication Between OAM Servers and WebGates	6-3
6.2.3	Conditions Requiring Server Restart	6-4
6.3	Managing Individual OAM Server Registrations	6-4
6.3.1	OAM Server Registration Page	6-4
6.3.1.1	OAM Proxy Settings	6-5
6.3.2	Registering a Fresh OAM Server Instance	6-7
6.3.3	Viewing or Editing Individual OAM Server Registrations and Proxy Settings	6-8
6.3.4	Deleting an Individual Server Registration	6-8

Part III Logging, Auditing, Reporting and Monitoring Performance

7 Logging Component Event Messages

7.1	About Oracle Access Management Logging	7-1
7.2	Logging Component Event Messages	7-1
7.2.1	Component Loggers for Security Token Service and Access Manager	7-3
7.2.2	Sample Logger and Log Handler Definition	7-4
7.2.3	About Logging Levels	7-5

7.3	Configuring Logging for Access Manager	7-6
7.3.1	Modifying the Logger Level for Access Manager	7-6
7.3.2	Adding an Access Manager-Specific Logger and Log Handler	7-8
7.4	Configuring Logging for Identity Federation	7-9
7.4.1	Configuring Logging for Identity Federation	7-10
7.4.2	Defining Log Level and Log Details for Security Token Service or Identity Federation	7-10
7.5	Validating Run-time Event Logging Configuration	7-11

8 Auditing Administrative and Run-time Events

8.1	Introduction to Oracle Fusion Middleware Auditing	8-1
8.2	Oracle Access Management Auditing	8-2
8.2.1	Understanding Oracle Access Management Auditing	8-2
8.2.2	About Oracle Access Management Auditing Configuration	8-2
8.2.3	About Audit Record Storage	8-3
8.2.4	About Audit Reports and Oracle Business Intelligence Publisher	8-5
8.2.5	Oracle BI Enterprise Edition (Oracle BI EE)	8-5
8.2.6	About the Audit Log and Data	8-6
8.3	Access Manager Events You Can Audit	8-7
8.3.1	Access Manager Administrative Events You Can Audit	8-7
8.3.2	Access Manager Run-time Events You Can Audit	8-10
8.3.3	Auditing Authentication Events	8-12
8.4	Identity Federation Events You Can Audit	8-13
8.4.1	Session Management Events for Identity Federation	8-14
8.4.2	Protocol Flow Events for Identity Federation	8-14
8.4.3	Server Configuration Events for Identity Federation	8-15
8.4.4	Security Events for Identity Federation	8-15
8.5	Setting Up Auditing for Oracle Access Management	8-16
8.5.1	Setting Up the Audit Database Store	8-16
8.5.2	Preparing Oracle Business Intelligence Publisher EE	8-17
8.5.3	Using the Oracle Access Management Console for Audit Configuration	8-19
8.5.4	Adding, Viewing, or Editing Audit Settings	8-21
8.6	Validating Auditing and Reports	8-21

9 Logging WebGate Event Messages

9.1	Understanding Logging for WebGate Instances	9-1
9.1.1	About Logging, Log Levels, and Log Output	9-1
9.1.2	Log Levels	9-2
9.1.3	Log Output	9-4
9.2	About Log Configuration File Paths and Contents	9-4

9.2.1	Log Configuration File Paths and Names	9-5
9.2.2	Log Configuration File Contents	9-6
9.2.2.1	When Changes to the File Take Effect	9-6
9.2.2.2	Comments in the Log File	9-6
9.3	About Directing Log Output to a File or the System File	9-10
9.4	Structure and Parameters of the WebGate Log Configuration File	9-11
9.4.1	Structure of WebGate Log Configuration XML File Header	9-11
9.4.2	Structure of WebGate Initial Compound List	9-11
9.4.3	Parameters in the WebGate Simple List and Logging Threshold	9-12
9.4.4	Parameters in the WebGate Second Compound List and Log Handlers	9-14
9.4.5	Parameters in the WebGate List for Per-Module Logging	9-15
9.4.6	Parameters in the WebGate Filter List	9-15
9.4.7	WebGate XML Element Order	9-16
9.5	Activating and Suppressing Logging Levels	9-17
9.5.1	About Log Handler Precedence	9-17
9.6	Mandatory Log Configuration File Parameters	9-18
9.6.1	Settings in the Default Log Configuration File	9-20
9.6.2	Description of the Settings in the Default Log Configuration File	9-22
9.7	Configuring Different Threshold Levels for Different Types of Data	9-23
9.7.1	About the MODULE_CONFIG Section	9-24
9.7.1.1	Location of the Per-Module Logging Section in the Log Configuration File	9-25
9.7.1.2	List of Modules That Can Be Logged	9-25
9.7.2	Configuring a Log Level Threshold for a Function or Module	9-26
9.8	Filtering Sensitive Attributes	9-28

10 Understanding Oracle Access Management Reports

10.1	About Reports in Oracle Access Management	10-1
10.2	Accessing Oracle Access Management Reports	10-2
10.3	Supported Output Formats	10-2
10.4	Classification of Reports for Access Manager	10-3
10.4.1	Account Management Reports	10-3
10.4.2	Authentication Reports	10-3
10.4.2.1	AuthenticationFromIPByUser	10-3
10.4.2.2	AuthenticationPerIP	10-4
10.4.2.3	Authentication Statistics Report	10-4
10.4.2.4	AuthenticationStatisticsPerServer Report	10-4
10.4.3	Errors and Exceptions	10-5
10.4.3.1	All Errors and Exceptions	10-5
10.4.3.2	Authentication Failures	10-5
10.4.3.3	User Activities	10-6

10.4.3.4	Authentication History	10-6
10.4.3.5	Authorization History	10-6
10.4.3.6	Multiple Logins From Same IP	10-6
10.5	About Creating Reports Using Third-Party Software	10-7

11 Monitoring Oracle Access Management Performance and Access Manager Health

11.1	Introduction to Performance Monitoring	11-1
11.2	Monitoring Server Metrics Using Oracle Access Management Console	11-2
11.2.1	Monitoring Server Instance Performance	11-2
11.2.2	Oracle Access Manager Server Metrics	11-3
11.3	OAM Proxy Metrics and Tuning	11-6
11.3.1	OAM Proxy Metrics	11-6
11.3.2	OAM Proxy Server Tuning Parameters	11-7
11.4	Monitoring Metrics Using the DMS Console	11-8
11.4.1	Monitoring OAM Metrics	11-8
11.5	Monitoring the Health of an Access Manager Server	11-8
11.5.1	Understanding WebGate and Access Manager Communications	11-9
11.5.2	Monitoring Access Manager Server Health	11-9
11.6	Monitoring Server Health with Health Check Framework	11-10
11.6.1	Introduction to HealthCheck Framework	11-11
11.6.2	Understanding HealthCheck Test Configuration	11-11
11.6.3	Running Health Checks Using REST API	11-13
11.6.4	Configuring Scheduled Health Checks	11-14
11.6.5	Using the Health Script Evaluator	11-16

12 Monitoring Performance and Logs with Fusion Middleware Control

12.1	Introduction to Fusion Middleware Control	12-1
12.2	Logging In to and Out of Fusion Middleware Control	12-2
12.2.1	Logging In To Fusion Middleware Control	12-3
12.2.2	Logging Out of Fusion Middleware Control	12-3
12.3	Displaying Menus and Pages in Fusion Middleware Control	12-3
12.3.1	Farm Page in Fusion Middleware Control	12-4
12.3.2	Context Menus and Pages in Fusion Middleware Control	12-5
12.3.3	Displaying Context Menus and Target Details in Fusion Middleware Control	12-8
12.4	Viewing Performance in Fusion Middleware Control	12-9
12.4.1	Resulting Pages for Selected Nodes and Targets	12-9
12.4.2	Performance Overview Pages in Fusion Middleware Control	12-10
12.4.2.1	Access Manager Component Pages	12-14

12.4.3	Metrics Palette and the Performance Summary Page	12-16
12.4.4	Displaying Performance Metrics in Fusion Middleware Control	12-19
12.4.5	Displaying Component-Specific Performance Details	12-21
12.5	Managing Log Level Changes in Fusion Middleware Control	12-21
12.5.1	Dynamic Log Level Changes in Fusion Middleware Control	12-22
12.5.2	Setting Log Levels Dynamically Using Fusion Middleware Control	12-26
12.6	Managing Log File Configuration from Fusion Middleware Control	12-26
12.6.1	Log File Configuration Page in Fusion Middleware Control	12-27
12.6.2	Managing Log Files with Fusion Middleware Control	12-29
12.7	Viewing Log Messages in Fusion Middleware Control	12-31
12.7.1	About Finding, Viewing, and Exporting Log Messages	12-31
12.7.1.1	Log Messages Page in Fusion Middleware	12-31
12.7.2	Viewing Logged Messages With Fusion Middleware Control	12-35
12.8	Displaying MBeans in Fusion Middleware Control	12-37
12.8.1	Fusion Middleware Control System MBean Browser	12-37
12.8.2	Managing Mbeans	12-40

Part IV Managing Access Manager Settings and Agents

13 Configuring Access Manager Settings

13.1	Oracle Access Management Overview	13-1
13.2	Managing Load Balancing	13-1
13.2.1	About Common Load Balancing Settings	13-1
13.2.2	Managing OAM Server Load Balancing Settings	13-3
13.3	Managing Secure Error Modes	13-3
13.3.1	OAM Server Error Modes	13-3
13.3.2	Viewing or Editing OAM Server Secure Error Modes	13-6
13.4	Managing WebGate Traffic Load Balancer	13-6
13.4.1	About WebGate Traffic Load Balancer	13-6
13.4.2	Viewing or Editing WebGate Traffic Load Balancer	13-7
13.5	Managing SSO Tokens and IP Validation	13-7
13.5.1	Access Manager SSO Tokens and IP Validation Settings	13-7
13.5.2	Viewing or Editing SSO Tokens and IP Validation	13-8
13.6	Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security	13-9
13.6.1	OAM Proxy Simple and Cert Mode Transport Security	13-9
13.6.2	Configuration Settings of Common OAM Proxy Page for Secure Server Communications	13-11
13.6.3	Viewing or Editing Simple or Cert Settings for OAM Proxy	13-11
13.6.4	Configuring 64-bit WebGate in Cert Mode	13-12
13.6.5	Tuning the Simple Mode WebGate	13-12

13.7	Managing Run Time Policy Evaluation Caches	13-12
13.7.1	Settings for Run Time Policy Evaluation Caches	13-12
13.7.2	Managing Run Time Policy Evaluation Caches	13-13
13.8	Configuring Policy Cache Parameters	13-14

14 Introduction to Agents and Registration

14.1	Introduction to Policy Enforcement Agents	14-1
14.1.1	Agent Types and Runtime Processing for OAM Agents	14-1
14.1.2	About OAM WebGate Configured as a Detached Credential Collector	14-4
14.2	Introduction to Agent Registration	14-5
14.2.1	Keys and Policies Generated during Agent Registration	14-5
14.2.2	File System Changes and Artifacts for Registered Agents	14-6
14.3	OAM Remote Registration	14-7
14.3.1	Performing In-Band Remote Registration	14-8
14.3.2	Performing Out-of-Band Remote Registration	14-8
14.3.3	Updating Agent Configuration Files	14-9

15 Registering and Managing OAM Agents

15.1	Before Registering and Managing Agents	15-1
15.2	OAM Agent Registration Parameters in the Console	15-1
15.2.1	Creating OAM WebGate Page and Parameters	15-2
15.2.2	User-Defined WebGate Parameters	15-5
15.2.3	IP Address Validation for WebGates	15-11
15.2.3.1	IP Validation Exceptions List	15-12
15.2.3.2	IP Validation in Load Balanced Environments	15-12
15.3	Registering an OAM Agent Using the Console	15-14
15.4	Bulk Updates to WebGates	15-15
15.4.1	Updating Multiple WebGate Profiles	15-16
15.4.1.1	Creating a WebGate Template and Mapping WebGates to that Template	15-16
15.4.2	WLST Commands for Bulk Updates to WebGate Profiles	15-17
15.4.2.1	createWebgateTemplate	15-17
15.4.2.2	updateWebgateTemplateToWebgateMapping	15-17
15.4.2.3	updateWebgateTemplateParams	15-18
15.4.2.4	removeWebgateTemplateParams	15-19
15.4.2.5	rollbackWebgatesToPreviousState	15-20
15.4.2.6	showWebgateTemplate	15-20
15.5	Configuring and Managing Registered OAM Agents Using the Console	15-21
15.5.1	Registered OAM Agent Configuration Parameters in the Console	15-21
15.5.2	WebGate Search Controls	15-27

15.5.2.1	Searching for an OAM Agent Registration	15-28
15.5.3	Viewing or Editing an OAM Agent Registration Page in the Console	15-29
15.5.4	Deleting OAM Agent Registration Using the Console	15-30
15.6	Remote Registration Tool, Modes, and Process	15-31
15.6.1	Remote Registration Command Arguments and Modes	15-31
15.6.2	Common Elements within Remote Registration Request Templates	15-33
15.6.3	Key Use, Generation, Provisioning, and Storage	15-37
15.6.3.1	Key Use	15-38
15.6.3.2	Key Generation Process	15-38
15.6.3.3	Key Accessibility and Provisioning	15-38
15.6.3.4	Key Storage	15-39
15.7	Remote Registration Templates: OAM Agents	15-39
15.7.1	OAM Agent Parameters for Remote Registration	15-40
15.8	Performing Remote Registration for OAM Agents	15-44
15.8.1	Acquiring and Setting Up the Remote Registration Tool	15-44
15.8.2	Creating Your Remote Registration Request	15-45
15.8.3	Performing In-Band Remote Registration	15-46
15.8.4	Performing Out-of-Band Remote Registration	15-47
15.9	Remote Agent Update Modes and Templates	15-49
15.9.1	Remote Agent Update Modes	15-49
15.9.2	Remote OAM Agent Updates Template	15-49
15.10	Updating Agents Remotely	15-50
15.10.1	Updating Agent Registrations Remotely	15-50
15.10.2	Validating an Agent Registration Remotely	15-51
15.10.3	Removing an Agent Registration Remotely	15-51
15.11	Validating Remote Registration and Resource Protection	15-51
15.11.1	Validating Agent Registration using the Oracle Access Management Console	15-52
15.11.2	Verifying Authentication and Access After Remote Registration	15-52
15.12	setAllowEmptyHostIdentifier	15-54

16 Maintaining Access Manager Sessions

16.1	Introducing Access Manager Session Management	16-1
16.2	Understanding Server-Side Session Management	16-3
16.2.1	About Securing Access Manager Sessions	16-3
16.2.2	Access Manager Session Lifecycle, States, and Enforcement	16-4
16.2.2.1	Global Session Enforcement Checks for State Changes	16-5
16.2.2.2	Access Manager Session Removal	16-6
16.2.2.3	About Step-Up and Step-Down Authentication and Credentials	16-6
16.2.2.4	Optional Application-Specific Session Enforcement	16-6
16.3	Server-Side Session Enforcement Examples	16-7

16.3.1	Example 1: Single Authentication Scheme	16-7
16.3.2	Example 2: Multiple Authentication Schemes	16-8
16.4	Configuring the Server-Side Session Lifecycle	16-9
16.4.1	Global Session Lifecycle Settings	16-10
16.4.2	Polling Interval for System and Policy Configuration	16-12
16.4.3	Application-Specific Session Overrides	16-13
16.4.4	Viewing or Modifying Global Session Settings	16-14
16.4.5	Viewing or Modifying Optional Application-Specific Session Overrides	16-14
16.5	Managing Active Server-Side Sessions	16-15
16.5.1	Session Management Controls	16-15
16.5.2	Locating and Managing Active Sessions	16-17
16.6	Validating Server-Side Session Operations	16-18
16.7	Using REST APIs for CRUD Operations on a Session	16-19
16.7.1	Searching Sessions Using Session REST API	16-20
16.7.2	Updating a Session Using Session REST API	16-22
16.7.3	Deleting a Session Using Session REST API	16-22
16.7.4	Assigning a Delegated Administrator	16-24

Part V Implementing Multi-Data Centers

17 Understanding Multi-Data Centers

17.1	Introducing the Multi-Data Center	17-1
17.1.1	Understanding Cookies for Multi-Data Center	17-3
17.1.1.1	OAM_ID Cookie	17-3
17.1.1.2	OAMAuthn WebGate Cookie	17-4
17.1.1.3	OAM_GITO (Global Inactivity Time Out) Cookie	17-4
17.1.2	Session Adoption During Authorization	17-5
17.1.3	Session Indexing	17-6
17.1.4	Supported Multi-Data Center Topologies	17-6
17.1.4.1	The MDC Active-Active Mode	17-6
17.1.4.2	The MDC Active-Passive Mode	17-8
17.1.4.3	The MDC Active-Hot Standby Mode	17-8
17.2	Multi-Data Center Deployments	17-9
17.2.1	Session Adoption Without Re-authentication, Session Invalidation or Session Data Retrieval	17-10
17.2.2	Session Adoption Without Re-authentication But With Session Invalidation and Session Data Retrieval	17-11
17.2.3	Session Adoption Without Re-authentication and Session Invalidation But With On-demand Session Data Retrieval	17-12
17.2.4	Authentication and Authorization Requests Served By Different Data Centers	17-12

17.2.5	Logout and Session Invalidation	17-14
17.2.6	Stretch Cluster Deployments	17-15
17.3	Active-Active Multi-Data Center Topology Deployment	17-17
17.4	Load Balancing Between Access Management Components	17-19
17.5	Understanding Time Outs and Session Syncs	17-21
17.5.1	Maximum Session Constraints	17-21
17.5.2	Multi-Data Center Policy Configurations for Idle Timeout	17-22
17.5.3	Expiring Multi-Data Center Sessions	17-22
17.5.4	Session Synchronization and Multi-Data Center Fail Over	17-22
17.6	Replicating a Multi-Data Center Environment	17-25
17.6.1	Replicating Data Using the WLST	17-25
17.6.2	Syncing Data Using Automated Policy Synchronization	17-26
17.7	Multi-Data Center Recommendations	17-26
17.7.1	Using a Common Domain	17-26
17.7.2	Concerning the DCC and the OAM_GITO	17-27
17.7.3	Using an External Load Balancer	17-27
17.7.4	Honoring Maximum Sessions	17-28
17.7.5	WebGate Cookie Cannot be Refreshed During Authorization	17-28

18 Configuring Multi-Data Centers

18.1	Before Setting Up a Multi-Data Center	18-1
18.2	Primary Multi-Data Center Use Cases	18-2
18.3	Setting Up a Master and a Clone in Multi-Data Center	18-3
18.4	Adding an Additional Clone Data Center to the Existing Multi-Data Center Setup	18-7
18.5	Multi-Data Center Security Modes	18-10
18.5.1	OPEN Security Mode	18-11
18.5.2	SIMPLE Security Mode	18-12
18.5.3	CERT Security Mode	18-14

19 Synchronizing Data In A Multi-Data Center

19.1	Understanding the Multi-Data Center Synchronization	19-1
19.1.1	How Replication Works	19-2
19.1.2	Understanding the Replication Agreement	19-4
19.1.3	About Synchronizing Data Manually in a Multi-Data Center	19-4
19.2	Enabling Data Replication	19-4
19.3	Synchronizing Master and Clone Metadata	19-5
19.3.1	Synchronizing the UDM Metadata	19-5
19.4	Using REST API for Replication Agreements	19-5
19.4.1	Querying for Replication Agreement Details	19-6

19.4.2	Creating a Replication Agreement	19-7
19.4.3	Modifying a Replication Agreement	19-9
19.4.4	Deleting a Replication Agreement	19-10
19.5	Customizing Transformation Rules	19-11
19.6	Disabling Automated Policy Synchronization	19-14
19.7	Best Practices for Replication	19-15
19.7.1	Enabling Replication Logs	19-16
19.7.2	Changing the User Identifier	19-16

20 Setting Up the Multi-Data Center: A Sequence

20.1	Before You Begin	20-1
20.2	Setting Up a Multi-Data Center	20-2
20.3	Enabling Automated Policy Synchronization	20-8
20.4	Troubleshooting the Multi-Data Center Setup	20-12
20.4.1	Unauthorized Error Displayed When the Authorization Header is Correct	20-12
20.4.2	Curl Command Returns Curl: (35) SSL Connect Error	20-13
20.4.3	APS Synchronization Failed With 401-UnAuthorized Error	20-13
20.4.4	Fail to Decrypt oamkeystore Data with Cipher Key from OAM Config	20-14
20.4.5	Modifying the Polling Interval in Clone Data Centers	20-14
20.4.6	Overwriting the Existing MDC Configuration or Recovering from an Inconsistent State	20-15
20.4.7	Changing the Security Mode of Managed Servers in Working MDC Environment	20-16
20.4.8	MDC Session Adoption Issues in 11g–12c OAM Setup with SIMPLE Mode Servers	20-17
20.4.9	Request Failed When the Input Parameters Passed are Valid	20-17
20.4.10	Modifying Session Control Parameters	20-17
20.4.11	Modifying Backward Compatibility Flag	20-18
20.4.12	Disabling MDC	20-19
20.4.13	Backup Existing Artifacts in a Data Center	20-19

Part VI Managing Access Manager SSO, Policies, and Testing

21 Understanding Single Sign-On with Access Manager

21.1	Access Manager Single Sign-On Components	21-1
21.1.1	Multiple Network Domain SSO	21-4
21.1.2	Application SSO and Access Manager	21-4
21.1.3	Multiple WebLogic Server Domain SSO	21-5
21.1.4	Reverse-Proxy SSO	21-6

21.2	Access Manager Policy Model	21-6
21.3	Anatomy of an Application Domain and Policies	21-10
21.3.1	Resource Definitions for Policies	21-11
21.3.2	About Authentication Policies	21-12
21.3.3	About Authorization Policies	21-13
21.3.4	About Token Issuance Policies	21-13
21.4	Policy Conditions and Rules	21-14
21.5	Understanding SSO Cookies	21-15
21.5.1	Single Sign-On Cookies During User Login	21-15
21.5.2	Single Sign-On Server and Agent Cookies	21-16
21.5.2.1	OAM_ID cookie	21-16
21.5.2.2	OAMAuthnCookie for OAM Webgates	21-17
21.5.2.3	OAM_REQ Cookie	21-17
21.5.2.4	OAMRequestContext	21-18
21.5.2.5	DCCCtxCookie	21-18
21.5.2.6	DCCCtxCookie_COUNT	21-19
21.5.3	Support for SameSite=None Attribute in OAM Cookies	21-19
21.6	Configuring Single Sign-On with Access Manager	21-21

22 Managing Authentication and Shared Policy Components

22.1	Prerequisites to Managing Authentication and Shared Policy Components	22-1
22.2	Configuring Shared Policy Components	22-2
22.3	Managing Resource Types	22-2
22.3.1	Resource Types and Their Use	22-3
22.3.2	Resource Type Page	22-3
22.3.3	Searching for a Specific Resource Type	22-6
22.3.4	Creating a Custom Resource Type	22-6
22.4	Managing Host Identifiers	22-7
22.4.1	About Host Identifiers	22-7
22.4.1.1	Host Identifier Usage	22-8
22.4.1.2	Host Identifier Guidelines	22-9
22.4.1.3	Host Identifier Variations	22-9
22.4.2	About Virtual Web Hosting	22-10
22.4.2.1	Configuring Virtual Hosting for Non-Apache Web Servers	22-10
22.4.2.2	Associating a Webgate for Apache with Virtual Hosts, Directories, or Files	22-11
22.4.3	Host Identifier Page	22-12
22.4.4	Creating a Host Identifier	22-13
22.4.5	Searching for a Host Identifier Definition	22-14
22.4.6	Viewing or Editing a Host Identifier Definition	22-15
22.4.7	Deleting a Host Identifier Definition	22-16

22.5	Understanding Authentication Methods and Credential Collectors	22-16
22.5.1	Authentication Methods Supported by Access Manager	22-17
22.5.2	Embedded Credential Collector Versus Detached Credential Collector	22-18
22.5.3	Authentication Event Logging and Auditing	22-23
22.6	Managing Native Authentication Modules	22-24
22.6.1	Native Access Manager Authentication Modules	22-24
22.6.1.1	Native Kerberos Authentication Module	22-25
22.6.1.2	Native LDAP Authentication Modules	22-26
22.6.1.3	Native X.509 Authentication Module	22-27
22.6.2	Viewing or Editing Native Authentication Modules	22-29
22.6.3	Deleting a Native Authentication Module	22-30
22.7	Orchestrating Multi-Step Authentication with Plug-in Based Modules	22-30
22.7.1	Simple Form Versus Multi-Factor (Multi-Step) Authentication	22-31
22.7.2	Access Manager Plug-ins for Multi-Step Authentication Modules	22-33
22.7.3	Pre-populated Plug-ins for Configuring Access Manager with Multi-Step Authentication	22-45
22.7.4	Example: Leveraging SubjectAltName Extension Data and Integrating with Multiple OCSP Endpoints	22-52
22.7.5	Creating a Custom Authentication Module using Bundled Plug-ins	22-54
22.7.6	Steps and Plug-ins in Customized Step-up Authentication Module	22-55
22.7.7	Configuring Step-up Authentication	22-58
22.7.8	Configuring an HTTPToken Extractor Plug-in	22-62
22.7.9	JSON Web Token Plug-in	22-62
22.7.9.1	Understanding the JSON Web Token Plug-in	22-63
22.7.9.2	Configuring the JSON Web Token Plug-In	22-64
22.7.10	X.509 Authentication Using Extended Key Usage (EKU)	22-65
22.7.10.1	X.509 Authentication Module for EKU Validation	22-65
22.7.10.2	X509 Steps Orchestration for EKU	22-69
22.7.10.3	X509 Scheme for EKU	22-69
22.7.10.4	Protecting Resources with X509 for EKU Scheme	22-70
22.8	Deploying and Managing Individual Plug-ins for Authentication	22-70
22.8.1	About Managing Your Own Authentication Plug-ins	22-70
22.8.2	Making Custom Authentication Plug-ins Available for Use	22-71
22.8.3	Checking an Authentication Plug-in's Activation Status	22-72
22.8.4	Deleting Your Custom Authentication Plug-ins	22-72
22.8.5	Plug-ins Page	22-73
22.8.6	Plug-in Details Page	22-75
22.9	Managing Authentication Schemes	22-76
22.9.1	Authentication Schemes and Pages	22-77
22.9.1.1	Pre-configured Authentication Schemes	22-81
22.9.1.2	Credential Challenge Methods	22-86
22.9.1.3	Challenge Parameters for Authentication Schemes	22-89

22.9.2	Understanding Multi-Level and Step-Up Authentication	22-94
22.9.2.1	About Multi-Level and Step-Up Authentication	22-95
22.9.2.2	Changing Security Level of an Authentication Scheme during the Authentication Process	22-95
22.9.3	Creating an Authentication Scheme	22-96
22.9.4	Searching for an Authentication Scheme	22-97
22.9.5	Viewing, Editing, or Deleting an Authentication Scheme	22-98
22.10	Extending Authentication Schemes with Advanced Rules	22-99
22.10.1	Advanced Rules Use Cases	22-100
22.10.2	Context Data for Advanced Rules	22-101
22.11	Configuring Challenge Parameters for Encrypted Cookies	22-103
22.11.1	Challenge Parameters for Encrypted Cookies	22-103
22.11.2	Configuring Challenge Parameters for Security of Encrypted Cookies	22-105
22.11.3	Setting Challenge Parameters for Persistence of Encrypted Cookies	22-105
22.12	Configuring Authentication POST Data Handling	22-106
22.12.1	Authentication POST Data Preservation and Restoration	22-106
22.12.2	Authentication POST Data Handling	22-108
22.12.3	Post Data Size Limits	22-109
22.12.4	Configuring Authentication POST Data Handling	22-110
22.12.5	Testing POST Data Handling Configuration	22-111
22.13	Long URL Handling During Authentication	22-111
22.13.1	About Long URLs and Authentication Handling	22-111
22.13.2	Configuration Requirements for Long URL Handling	22-112
22.14	Using Application Initiated Authentication	22-114

23 Understanding Credential Collection and Login

23.1	Overview of Access Manager Credential Collection	23-1
23.1.1	Overview of the Login process with Self-Service Provisioning Applications	23-2
23.1.2	Overview of the Login Process with Access Manager-Protected Resources	23-2
23.2	Overview of the SSO Login Process with OAM Agents and ECC	23-3
23.3	Overview of the SSO Login Process with OAM Agents and DCC	23-6
23.4	Configuring OAM WebGate and Authentication Policy for DCC	23-10
23.4.1	Enabling DCC Credential Operations	23-11
23.4.2	Locating and Updating DCC Forms for Password Policy	23-12
23.4.3	Adding PasswordPolicyValidationScheme to Authentication Policy for DCC	23-12
23.4.4	Supporting Federation Flows With DCC	23-14
23.5	Tunneling from DCC to Access Manager Over Oracle Access Protocol	23-15
23.5.1	How DCC Tunneling with OAP Works	23-15
23.5.2	Configuring OAP Tunneling	23-16
23.6	Configuring a DCC WebGate for X509 Authentication	23-16
23.6.1	Configuring the WebLogic Server	23-16

23.6.1.1	Creating the Server and Trust Store	23-17
23.6.1.2	Configuring the WebLogic Server Instance	23-17
23.6.1.3	Creating the User Certificate	23-19
23.6.1.4	Adding the Root CA Certificate	23-19
23.6.2	Configuring a WebGate For DCC	23-20
23.6.3	Converting the DCC WebGate to SSL	23-21
23.6.3.1	Generating Server Certificates	23-21
23.6.3.2	Generating and Importing Client Certificates	23-22

24 Using Password Policy

24.1	Understanding Password Management	24-1
24.2	Enabling Password Management	24-2
24.3	Accessing Password Policy Configuration Page	24-3
24.3.1	Password Policy Configuration Page	24-4
24.4	Specifying Credential Collector URLs with Password Policy	24-6
24.5	Oracle-Provided Password Forms	24-10
24.6	Managing Global Password Policy	24-14
24.6.1	Defining Your Global Password Policy	24-15
24.6.2	Designating the Default Store for Your Password Policy	24-16
24.6.3	Adding Key Password Attributes to the Default Store	24-16
24.6.3.1	LDIF Files and Key Password Attributes for Password Policy	24-17
24.6.3.2	Extending the Default Store Schema with Password Policy Attributes	24-19
24.6.4	Adding an Administrator to Change User Attributes After a Password Change	24-19
24.7	Configuring Password Policy Authentication	24-20
24.7.1	Password Policy Validation Module	24-20
24.7.2	Configuring the PasswordPolicyValidationScheme	24-24
24.7.3	Adding Your PasswordPolicyValidationScheme to ECC Authentication Policy	24-26
24.7.4	Supporting DCC Authentication Schemes with Pre-Authentication Rules	24-27
24.8	Completing Password Policy Configuration	24-27
24.8.1	Setting the Error Message Mode for Password Policy Messages	24-27
24.8.2	Overriding Native LDAP Password Policy Validation	24-28
24.8.3	Disabling ECC Operation and Using DCC Exclusively	24-29
24.8.4	Testing Your Multi-Step Authentication	24-30
24.9	Configuring the PasswordManagementPlugin	24-31
24.9.1	Configuring Password Policy for PasswordManagement Service	24-31
24.9.2	Extending the LDAP Definitions	24-31
24.9.3	Configuring Password Policy Management Module	24-32
24.9.4	Setting up the Forgot Password Module	24-35
24.9.5	Configuring Forgot Password using OTP	24-35

25 Managing Policies to Protect Resources and Enable SSO

25.1	Prerequisites to Managing Policies and Protecting Resources	25-1
25.2	Introduction to Application Domain and Policy Creation	25-2
25.2.1	About Generating Application Domains and Policies Automatically	25-3
25.2.2	About Managing Application Domains and Policies Remotely	25-3
25.2.3	Creating or Managing an Application Domain and Policies	25-4
25.3	Understanding Application Domain and Policy Management	25-5
25.3.1	Application Domain Pages	25-5
25.3.2	Application Domain Summary Page	25-6
25.3.3	Resource Container in an Application Domain	25-6
25.3.4	Authentication Policy Pages	25-7
25.3.5	Authorization Policy Pages	25-9
25.3.6	Token Issuance Policy Pages	25-11
25.4	Managing Application Domains Using the Console	25-11
25.4.1	Creating a New Application Domain	25-11
25.4.2	Searching for an Existing Application Domain	25-12
25.4.3	Viewing or Editing an Application Domain	25-13
25.4.4	Deleting an Application Domain and Its Contents	25-13
25.5	Adding and Managing Policy Resource Definitions	25-14
25.5.1	Resources in an Application Domain	25-15
25.5.1.1	Resource Type in a Resource Definition	25-19
25.5.1.2	Host Identifier in a Resource Definition	25-19
25.5.1.3	Resource URL, Prefixes, and Patterns	25-20
25.5.1.4	Query String Name and Value Parameters for Resource Definitions	25-22
25.5.1.5	Literal Query Strings in Resource Definitions	25-27
25.5.1.6	Run Time Resource Evaluation	25-28
25.5.2	Defining Resources in an Application Domain	25-29
25.5.3	Searching for a Resource Definition	25-30
25.5.3.1	Search Elements and Results for Resource Definitions in an Application Domain	25-31
25.5.3.2	Searching for a Specific Resource Definition	25-32
25.5.4	Viewing, Editing, or Deleting a Resource Definition	25-32
25.6	Defining Authentication Policies for Specific Resources	25-33
25.6.1	Authentication Policy Page	25-34
25.6.1.1	Resources in an Authentication Policy	25-35
25.6.2	Creating an Authentication Policy for Specific Resources	25-36
25.6.3	Searching for an Authentication Policy	25-37
25.6.4	Viewing or Editing an Authentication Policy	25-37
25.6.5	Deleting an Authentication Policy	25-38

25.7	Defining Authorization Policies for Specific Resources	25-38
25.7.1	Authorization Policies for Specific Resources	25-39
25.7.2	Creating an Authorization Policy and Specific Resources	25-40
25.7.3	Searching for an Authorization Policy	25-41
25.7.4	Viewing or Editing an Authorization Policy and Resources	25-41
25.7.5	Deleting an Entire Authorization Policy	25-42
25.8	Configuring Success and Failure URLs for Authorization Policies	25-42
25.9	Introduction to Authorization Policy Rules and Conditions	25-43
25.9.1	About Allow or Deny Rules	25-44
25.9.2	Authorization Policy Conditions	25-44
25.9.3	About Classifying Users and Groups for Conditions	25-46
25.9.4	Guidelines for Authorization Responses Based on Conditions	25-47
25.10	Defining Authorization Policy Conditions	25-47
25.10.1	Choosing a Condition Type	25-47
25.10.1.1	Condition Window and Elements	25-48
25.10.1.2	Choosing a Condition Type	25-49
25.10.2	Defining Identity Conditions	25-50
25.10.2.1	About Identity Conditions	25-50
25.10.2.2	Specifying Identity Type Conditions	25-56
25.10.3	Defining IP4 Range Conditions	25-57
25.10.3.1	IP4 Range Condition Types	25-58
25.10.3.2	Defining IP4 Range Conditions	25-59
25.10.4	Defining Temporal Conditions	25-59
25.10.4.1	Temporal Conditions	25-60
25.10.4.2	Defining Temporal Conditions	25-61
25.10.5	Defining Attribute Conditions	25-61
25.10.5.1	Attribute-Type Conditions	25-62
25.10.5.2	Defining Attribute Type Conditions	25-65
25.10.6	Viewing, Editing, or Deleting Authorization Policy Conditions	25-66
25.11	Defining Authorization Policy Rules	25-67
25.11.1	Authorization Policy Rules	25-67
25.11.2	Expressions and Expression-Based Policy	25-69
25.11.2.1	Expression Evaluation in Authorization Rules	25-72
25.11.3	Defining Rules in an Authorization Policy	25-73
25.12	Configuring Policy Ordering	25-74
25.13	Introduction to Policy Responses for SSO	25-75
25.13.1	Authentication and Authorization Policy Responses for SSO	25-76
25.13.2	About the Policy Response Language	25-77
25.13.3	Namespace and Variable Names for Policy Responses	25-77
25.13.4	About Constructing a Policy Response for SSO	25-79
25.13.4.1	Simple Responses	25-79

25.13.4.2	Compound and Complex Responses	25-80
25.13.4.3	Multi-Valued Responses	25-81
25.13.5	About Policy Response Processing	25-81
25.13.6	Assertion Claims and Processing	25-82
25.14	Adding and Managing Policy Responses for SSO	25-82
25.14.1	Adding a Policy Response for SSO	25-82
25.14.2	Viewing, Editing, or Deleting a Policy Response for SSO	25-83
25.15	Validating Authentication and Authorization in an Application Domain	25-84
25.16	Understanding Remote Policy and Application Domain Management	25-85
25.16.1	Remote Policy Management Modes, Templates, and Flags	25-85
25.16.2	Create Policy Request Template	25-87
25.16.3	Update Policy Request Template	25-88
25.16.4	Remote Policy Management Template Elements	25-89
25.17	Managing Policies and Application Domains Remotely	25-90
25.18	Application and Application-types	25-91

26 Validating Connectivity and Policies Using the Access Tester

26.1	Prerequisites to Using the Access Tester to Validate Connectivity and Policies	26-1
26.2	Introduction to the Access Tester for Access Manager 12c	26-1
26.2.1	About OAM Agent and Server Interoperability	26-3
26.2.2	About Access Tester Security and Processing	26-5
26.2.3	About Access Tester Modes and Administrator Interactions	26-6
26.3	Installing and Starting the Access Tester	26-8
26.3.1	Installing the Access Tester	26-9
26.3.2	System Properties Supported by the Access Tester	26-9
26.3.3	Starting the Tester Without System Properties For Use in Tester Console Mode	26-11
26.3.4	Starting the Access Tester with System Properties For Use in Command Line Mode	26-11
26.3.4.1	About the Access Tester Command Line Mode	26-12
26.3.4.2	Starting the Access Tester with System Properties	26-12
26.4	Access Tester Console, Navigation, and Controls	26-13
26.4.1	Access Tester Menus and Command Buttons	26-15
26.5	Testing Connectivity and Policies from the Access Tester Console	26-16
26.5.1	Establishing a Connection Between the Access Tester and the OAM Server	26-17
26.5.1.1	Server Connection Panel in the Access Tester	26-17
26.5.1.2	Connecting the Access Tester with the OAM Server	26-19
26.5.2	Validating Resource Protection from the Access Tester Console	26-20
26.5.2.1	Protected Resource URI Panel in the Access Tester	26-20
26.5.2.2	Validating Resource Protection	26-22
26.5.3	Testing User Authentication from the Access Tester Console	26-22

26.5.3.1	User Identity Panel in the Access Tester	26-23
26.5.3.2	Testing User Credential Authentication	26-25
26.5.4	Testing User Authorization from the Access Tester Console	26-25
26.5.5	Observing Request Latency	26-26
26.6	Creating and Managing Test Cases and Scripts	26-27
26.6.1	About Test Cases and Test Scripts	26-27
26.6.2	Capturing Test Cases	26-28
26.6.3	Generating an Input Test Script	26-29
26.6.3.1	About Input Test Script	26-30
26.6.3.2	Generating an Input Test Script	26-30
26.6.4	Personalizing an Input Test Script	26-31
26.6.4.1	Test Script Control Parameters	26-31
26.6.4.2	Customizing a Test Script	26-32
26.6.5	Executing a Test Script	26-32
26.6.5.1	About Test Script Execution	26-32
26.6.5.2	Running a Test Script	26-34
26.7	Evaluating Scripts, Log File, and Statistics	26-35
26.7.1	About Evaluating Test Results	26-35
26.7.2	Saved Connection Configuration File	26-36
26.7.3	Generated Input Test Script	26-37
26.7.4	Target Output File Containing Test Run Results	26-38
26.7.5	Statistics Document	26-40
26.7.6	Execution Log	26-42

27 Configuring Centralized Logout for Sessions Involving OAM WebGates

27.1	Prerequisites for the Configuration of Centralized Logout Sessions Involving OAM WebGates	27-1
27.2	Introduction to Centralized Logout for Access Manager	27-1
27.2.1	About Centralized Logout for OAM WebGates	27-2
27.2.2	About Logout Parameters for OAM WebGates	27-3
27.3	Configuring Centralized Logout for OAM WebGates	27-4
27.3.1	Configuring Centralized Logout for OAM WebGates When the ECC is Used	27-5
27.3.2	Configuring Logout When Using Detached Credential Collector-Enabled WebGate	27-6
27.4	Validating Global Sign-On and Centralized Logout	27-7
27.4.1	Confirming Global Sign-On	27-7
27.4.2	Observing Centralized Logout	27-7

Part VII Managing Oracle Access Management Identity Federation

28 Introducing Identity Federation in Oracle Access Management

28.1	Integrating Identity Federation with Access Manager	28-1
28.2	Deploying Identity Federation with Oracle Access Management	28-2
28.3	Understanding How Identity Federation Works	28-3
28.4	Using Identity Federation	28-3
28.4.1	Achieving SSO	28-4
28.4.2	Logging Out	28-4
28.4.3	Authorizing	28-4
28.4.4	Forcing Authentication	28-5
28.4.5	Indicating a Passive Identity Provider	28-5
28.4.6	User and Assertion Mapping	28-5
28.4.7	Platform Dependencies	28-5
28.5	Initiating Federation SSO	28-5
28.5.1	IdP Initiated Federation SSO Service	28-6
28.5.1.1	Multivalue Attributes in SAML Assertion	28-6
28.5.2	SP Initiated Federation SSO Service	28-7
28.5.3	Attribute Consuming Service	28-7
28.5.3.1	Elements Of Attribute Consuming Service	28-7
28.5.3.2	WLST Commands For Attribute Consuming Service	28-9
28.6	Exchanging Identity Federation Data	28-16
28.6.1	Using SAML 2.0	28-16
28.6.1.1	SAML 2.0 Bindings for SSO and Federation	28-16
28.6.1.2	SAML 2.0 Bindings for Single Logout	28-17
28.6.1.3	SAML 2.0 NameID Formats	28-17
28.6.1.4	Securing SAML 2.0 Data	28-18
28.6.1.5	OAM SAML 2.0 Supported Encryption Algorithms	28-18
28.6.1.6	Changing Default Encryption Algorithm	28-19
28.6.1.7	SAML 2.0 Service Details	28-20
28.6.2	Using SAML 1.1	28-21
28.6.2.1	SAML 1.1 Profiles for Web Browser SSO	28-21
28.6.2.2	SAML 1.1 Logout Profile	28-21
28.6.2.3	SAML 1.1 NameID Formats	28-22
28.6.2.4	About SAML 1.1 Data Security	28-22
28.6.2.5	SAML 1.1 Service Details	28-22
28.6.3	Using OpenID 2.0	28-23
28.6.3.1	OpenID 2.0 Authentication/SSO	28-23
28.6.3.2	OpenID 2.0 Logout	28-24
28.6.3.3	OpenID 2.0 NameID Format	28-24
28.6.3.4	About OpenID 2.0 Data Security	28-24
28.6.3.5	OpenID 2.0 Extensions	28-24

28.6.3.6	OpenID 2.0 Service Details	28-25
28.6.4	Using WS-Federation 1.1	28-25
28.7	Administrating Identity Federation	28-26
28.8	Enabling Identity Federation	28-27

29 Managing Identity Federation Partners

29.1	Understanding Federation And Partners	29-1
29.2	Managing Federation Partners	29-1
29.3	Administering Identity Federation As A Service Provider	29-2
29.3.1	Creating Remote Identity Provider Partners	29-3
29.3.1.1	Defining a New SAML 2.0 Identity Provider for Federation	29-6
29.3.1.2	Defining a New SAML 1.1 Identity Provider for Federation	29-6
29.3.1.3	Defining a New OpenID 2.0 Identity Providers for Federation	29-7
29.3.1.4	Enabling OpenID Simple Registration	29-9
29.3.1.5	Disabling OpenID Simple Registration	29-9
29.3.2	Managing the Remote Identity Provider Partners	29-9
29.3.2.1	Searching for Existing Identity Providers	29-9
29.3.2.2	Updating Identity Providers for Federation	29-10
29.4	Administering Identity Federation As An Identity Provider	29-11
29.4.1	Creating Remote Service Provider Partners	29-11
29.4.2	Managing the Remote Service Provider Partners	29-12
29.5	Using Attribute Mapping Profiles	29-13
29.5.1	Using the SP Attribute Mapping Profile	29-14
29.5.1.1	AWS Role Mapping Attribute in SAML Response	29-16
29.5.2	Using Attribute Value Mapping and Filtering	29-16
29.5.2.1	About Attribute Value Mapping	29-17
29.5.2.2	Configuring Attribute Value Mapping	29-17
29.5.2.3	About Attribute Value Filtering	29-20
29.5.2.4	Configuring Attribute Value Filtering	29-21
29.5.3	Using the IdP Attribute Mapping Profile	29-24
29.6	Mapping Federation Authentication Methods to Access Manager Authentication Schemes	29-25
29.6.1	Understanding Federation SSO As An IdP	29-26
29.6.2	Understanding Federation SSO As An SP	29-27
29.6.3	Configuring an Alternate Authentication Scheme	29-27
29.6.4	Using WLST For Mapping Administration	29-28
29.7	Using the Attribute Sharing Plug-in for the Attribute Query Service	29-28
29.7.1	Understanding the Plug-in and Query Service Design	29-28
29.7.1.1	Using the SP Attribute Requester	29-30
29.7.1.2	Using the IdP Attribute Responder	29-31
29.7.1.3	Using the SOAP Endpoint	29-32

29.7.2	Configuring for Attribute Sharing	29-32
29.7.2.1	NameID	29-33
29.7.2.2	NameID Format	29-34
29.7.2.3	IdP	29-34
29.7.2.4	RequestedAttributes	29-35
29.8	Using the Federation Proxy	29-35
29.9	Using WLST for Identity Federation Administration	29-36
29.10	Configuring OAM (IDP) for SAML Holder-of-Key (HoK) Profile with OCI Government Regions (SP)	29-36

30 Managing Settings for Identity Federation

30.1	Prerequisites for Settings in Federation Identity	30-1
30.2	About Federation Settings	30-1
30.3	Managing General Federation Settings	30-2
30.3.1	About Managing General Federation Settings	30-2
30.3.2	Managing General Federation Settings	30-3
30.3.2.1	Prerequisites for General Federation Settings	30-3
30.3.2.2	Setting or Modifying General Settings for Federation	30-3
30.4	Managing Proxy Settings for Federation	30-3
30.4.1	About Proxy Settings for Federation	30-4
30.4.2	Managing Proxy Settings for Identity Federation	30-4
30.4.2.1	Prerequisites for Proxy Settings for Identity Federation	30-4
30.4.2.2	Setting or Modifying Proxy Settings for Federation	30-4
30.5	Defining Keystore Settings for Federation	30-5
30.5.1	About Managing Keystore Settings for Identity Federation	30-5
30.5.2	Managing Identity Federation Encryption/Signing Keys	30-6
30.5.2.1	Task Overview: Managing Identity Federation Encryption/Signing Keys	30-6
30.5.2.2	Resetting the System (.oamkeystore) and Trust (amtruststore) Keystore Password	30-6
30.5.2.3	Adding a New Key Entry to the System Keystore (.oamkeystore)	30-7
30.6	Exporting Metadata	30-10

31 Managing Federation Schemes and Policies

31.1	Use of Identity Federation and Access Manager Together	31-1
31.2	Using Authentication Schemes and Modules for Identity Federation	31-1
31.2.1	About the FederationScheme Authentication Scheme	31-2
31.2.2	About the FederationMTScheme	31-3
31.2.3	About the FederationPlugin Authentication Module	31-3
31.2.4	Managing Authentication with Identity Federation in 11g Release 2	31-5

31.2.4.1	Prerequisites for the Authentication with Identity Federation in 11g Release 2	31-5
31.2.4.2	Viewing or Modifying FederationScheme	31-5
31.2.4.3	Viewing or Modifying FederationPlugin	31-6
31.2.4.4	Adding an Authentication Policy with FederationScheme	31-6
31.3	Using Authentication Schemes and Modules for Oracle Identity Federation	31-7
31.3.1	About Scheme OIFScheme	31-8
31.3.2	About the OIFMTLDAPPlugin Authentication Module	31-9
31.3.3	Managing Authentication with Oracle Identity Federation Release 11gR1	31-9
31.3.3.1	Prerequisites for Authentication with Oracle Identity Federation Release 11gR1	31-10
31.3.3.2	Viewing or Modifying the OIFScheme Authentication Scheme	31-10
31.3.3.3	Prerequisites for Viewing or Modifying the OIFMTLDAPPlugin Authentication	31-10
31.3.3.4	Viewing or Modifying the OIFMTLDAPPlugin Authentication	31-10
31.3.3.5	Adding an Authentication Policy with OIFScheme	31-11
31.4	Managing Access Manager Policies for Use with Identity Federation	31-11
31.4.1	About Policy Responses with Assertion Attributes for Identity Federation	31-11
31.4.2	Defining Policy Responses with Assertion Attributes for Identity Federation	31-12
31.4.2.1	Background on Conditions and Responses for Identity Federation	31-12
31.4.2.2	Prerequisites for Viewing and Configuring Policy Responses with Assertion Attributes	31-13
31.4.2.3	Viewing or Configuring Responses with Assertion Attributes	31-13
31.5	Testing Identity Federation Configuration	31-14
31.5.1	Test SP Module	31-15
31.5.1.1	Enabling or Disabling the Test SP Module	31-15
31.5.2	Accessing the Test SP Module and Performing a Federation SSO Operation	31-15
31.5.3	Troubleshooting Errors During Federation Configuration After an Upgrade	31-15
31.6	Using the Default Identity Provisioning Plug-in	31-16
31.6.1	Why Use a Provisioning Plug-in?	31-16
31.6.2	About the Default Provisioning Plug-in	31-16
31.6.3	Using the Default Provisioning Plug-in	31-17
31.6.4	Switching to a Custom Provisioning Plug-in	31-17
31.7	Configuring the Identity Provider Discovery Service	31-17
31.7.1	Configuring the Bundled IdP Discovery Service	31-18
31.7.2	Configuring Identity Federation with a Custom IdP Discovery Service	31-18
31.7.3	Disabling the use of an IdP Discovery Service	31-20
31.8	Integrating OAM Identity Provider With Microsoft Office 365 Service Provider	31-21
31.8.1	Configuring Microsoft Office 365 for OAM Integration	31-21
31.8.2	Configuring OAM for Microsoft Office 365 Integration	31-22
31.8.2.1	Configuring for Web and Non-Web Clients	31-22
31.8.2.2	Additional Configurations for Non-Web Clients	31-23

31.8.3	Verifying Federation Single Sign-On	31-24
31.8.3.1	Verifying SP-Initiated SSO	31-24
31.8.3.2	Verifying IDP-Initiated SSO	31-24
31.8.3.3	Verifying Federation with Non Web-based Clients	31-25

32 Identity Federation Use Cases

32.1	Using Test SP Application in OIF and SP	32-1
32.1.1	Enabling Test SP Engine	32-1
32.1.2	Using the Test SP Engine	32-2
32.1.2.1	Starting Federation SSO Flow	32-2
32.1.2.2	Displaying Test SP Operation Result	32-3
32.1.2.3	Diagnosing Mapping and Response Validation Issues	32-4
32.2	Using Federation Attributes for OAM Authorization and Protected Web Applications	32-9
32.2.1	Overview of Authenticating User Access to a Protected Resource	32-10
32.2.2	Prerequisites for Setting up Federation SSO	32-11
32.2.3	Prerequisites for Protected Web Application	32-13
32.2.4	Constructing Authorization Policy Using Federation Attributes	32-14
32.2.5	Injecting Federation Attributes	32-18
32.3	Key and Certificate Management and Rollover in OIF and OSTs	32-21
32.3.1	Introduction to Key and Certificate Management and Rollover	32-21
32.3.2	Generating Keys and Certificates	32-23
32.3.3	Setting New Key Entries	32-24
32.3.3.1	Creating New Key Entry in .oamkeystore	32-24
32.3.3.2	Updating OIF and OSTs Settings	32-25
32.3.4	Using New Key Entries to Update Global Settings	32-25
32.3.4.1	Updating Global OIF Settings	32-26
32.3.4.2	Updating Global OSTs Settings	32-26
32.3.4.3	Updating OSTs Settings	32-26
32.3.5	Managing Key Rollover per Partner	32-27
32.3.6	Managing OIF Key Rollover	32-28
32.3.7	Managing OSTs Key Rollover	32-29
32.4	Cryptographic Settings in Oracle Identity Federation	32-32
32.4.1	Hashing Algorithms	32-32
32.4.2	Examples on SHA-1 Signed Messages	32-32
32.4.3	Examples on SHA-256 Signed Messages	32-33
32.4.4	Configuring OIF to use SHA-1 or SHA-256 Hashing Algorithm	32-34
32.4.5	Signing Outgoing Messages	32-35
32.4.5.1	OOTB Configurations for Outgoing SAML Messages	32-35
32.4.5.2	Configuring SAML 2.0 AuthnRequest	32-36
32.4.5.3	Changing SAML 2.0 Metadata	32-38

32.4.6	Signing Incoming Messages	32-39
32.4.6.1	OOTB Boolean Settings for Incoming SAML Messages	32-39
32.4.6.2	Configuring SAML 2.0 AuthnRequest	32-40
32.4.6.3	Configuring SAML 1.1 Assertion for Incoming Messages	32-42
32.4.6.4	Configuring SAML 2.0 Assertion for Incoming Messages	32-43
32.4.6.5	Changing SAML 2.0 Metadata of Incoming Messages	32-44
32.4.7	Configuring X.509 Certificate in Outgoing Message	32-44
32.4.8	Managing SAML 2.0 Encryption	32-45
32.4.8.1	OOTB Configuration to Encrypt Outgoing SAML Messages	32-45
32.4.8.2	Encrypting Outgoing Assertion	32-46
32.4.8.3	Configuring NameID and Attributes Properties	32-46
32.4.9	Encryption Algorithm	32-47

Part VIII Managing the Adaptive Authentication Service and Oracle Mobile Authenticator

33 Introducing the Adaptive Authentication Service

33.1	About Adaptive Authentication Service	33-1
33.2	Working with the Adaptive Authentication Service	33-2
33.2.1	Understanding the One Time Password Option	33-3
33.2.1.1	About using OTP through Email or SMS	33-4
33.2.1.2	About using OTP from Oracle Mobile Authenticator	33-5
33.2.2	Understanding the Access Request (Push) Notification Option	33-5
33.2.3	Using the Oracle Mobile Authenticator with OTP And Access Request	33-7
33.3	Understanding Adaptive Authentication Service and OMA Configurations	33-8
33.4	Configuring an Adaptive Authentication Service	33-8
33.4.1	Generating a Secret Key for the Oracle Mobile Authenticator	33-8
33.4.2	Configuring Oauth Services to enable the Secret Key API	33-9
33.4.3	Configuring the Adaptive Authentication Plug-in in the Oracle Access Management Console	33-11
33.4.4	Enabling User Lockout During the Multi Factor Authentication Flow	33-14
33.4.5	Setting Credentials for UMS, iOS, and Android	33-15
33.4.6	Creating a Java KeyStore for iOS Access Request (Push) Notifications	33-16
33.4.7	Configuring Push Notifications on Mobile Device	33-17
33.4.7.1	Configuring Host Name Verifier for Android Access Request (Push) Notifications	33-17
33.4.7.2	Creating OTPRestUserGroup and Modifying OAMMFAOTP	33-18
33.4.7.3	Verifying Push Notification Settings	33-18
33.4.7.4	Creating an Authentication Policy	33-19
33.4.7.5	Connecting with Messaging Server	33-19

33.4.7.6	Setting the GCM API key within the OAM Credential Store	33-20
33.4.7.7	Installing the Google CA Files into the OAM Keystore	33-20
33.4.7.8	Creating a Webpage to Deliver the OMA Application Profile to the Mobile Device	33-21
33.4.7.9	Testing SFA through Push Notification	33-22
33.4.7.10	Troubleshooting Push Notifications	33-23
33.4.8	Configuring Access Manager for VPN in a Use Case	33-24
33.4.9	Administering a Secret Key	33-25

34 Configuring the Oracle Mobile Authenticator

34.1	Understanding Oracle Mobile Authenticator Configuration	34-1
34.2	Using the Oracle Mobile Authenticator App	34-3
34.2.1	Adding an Account to the OMA App by Scanning the QR Code	34-3
34.2.2	Adding an Account to the OMA App Using the Configuration URL	34-4
34.2.3	Adding an Account to the OMA App by Entering the Key Manually	34-4
34.2.4	Using the Oracle Mobile Authenticator App as an Authentication Method	34-5
34.3	Managing the Oracle Mobile Authenticator App	34-6
34.3.1	Switching Between Grid View and List View	34-6
34.3.2	Editing Accounts in the OMA App	34-6
34.3.3	Reordering Accounts in the OMA App	34-7
34.3.4	Deleting an Account in the OMA App	34-7
34.3.5	Enabling App Protection	34-8
34.3.6	Changing Your OMA App PIN	34-9
34.3.7	Disabling OMA App PIN Protection	34-9
34.3.8	Managing Notification History in the OMA App	34-9
34.4	Configuring the Google Authenticator App	34-10

35 Configuring TOTP-based Multi Factor Authentication in OAM

Part IX Managing the Oracle Access Management OAuth Service and OpenIDConnect

36 Understanding OAuth Services

36.1	About Oracle Access Management OAuth Services	36-1
36.2	Understanding OAuth Services Authorization for Web Clients	36-1
36.2.1	Understanding 3-Legged Authorization	36-2
36.2.2	Understanding 2-Legged Authorization	36-4

36.3	Understanding the OAuth Services Components	36-5
36.3.1	Identity Domains	36-5
36.3.2	Clients	36-5
36.4	About OAuth Tokens	36-7
36.4.1	OAuth Access Tokens	36-7
36.4.2	OAuth Refresh Tokens	36-8
36.4.3	OAuth Token Revocation	36-8

37 Configuring OAuth Services in 12c

37.1	Set-up OAuth Services	37-1
37.2	Configuring OAuth Services Settings	37-2
37.2.1	Creating an Identity Domain	37-2
37.2.1.1	Token Signing Using Third-Party Certificates	37-4
37.2.2	Enabling Consent Management	37-8
37.2.3	Creating a Resource	37-9
37.2.4	Creating a Client	37-10
37.3	Enabling User Lock Validation	37-11
37.3.1	Enabling User Lock Validation if LDAPNoPasswordValidationSchemeOAuth Exists	37-12
37.3.2	Enabling User Lock Validation if LDAPNoPasswordValidationSchemeOAuth Does Not Exist	37-12
37.4	Enabling Consent Management on MDC	37-13
37.5	Configuring OAuth in Multi-Data Centers	37-14
37.6	Optional Parameters for Consent Management in Multi-Data Centers	37-15
37.7	Error Codes and Troubleshooting Steps for Consent Management on MDC	37-17
37.8	Dynamic Client Registration	37-18
37.8.1	Enabling Dynamic Client Registration	37-18
37.8.2	Creating OAuth Client Template	37-19
37.8.3	Getting Registration Tokens	37-20
37.8.4	Registering the Client using the Registration Token	37-23
37.8.5	Reading Client Details	37-25
37.8.6	Deleting Dynamically Registered Client	37-26
37.9	SSO Session Linking for OAuth Tokens	37-26
37.10	Runtime REST APIs for OAuth 12c	37-28
37.11	Revoking OAuth Tokens	37-33
37.11.1	Revoking OAuth Tokens by OAuth Clients	37-34
37.11.2	Revoking OAuth Tokens for a User, Client and Resource Server	37-40
37.12	Configuring Client Authentication	37-43
37.12.1	Configuring Identity Domain for Client Authentication	37-44
37.12.2	Configuring the Client for Client Authentication	37-45
37.12.3	Managing Client Certificates	37-46

37.13	Configuring mTLS Client Authentication	37-46
37.13.1	About Mutual Transport Layer Security (mTLS) in OAM	37-47
37.13.2	Configuring mTLS Endpoint	37-47
37.13.3	Managing Trust Certificates for mTLS	37-48
37.13.4	Configuring Additional Options to Support mTLS	37-50
37.13.5	Sample 2-Legged mTLS Authentication Flow	37-50
37.13.6	Sample 3-Legged mTLS Authentication Flow	37-54
37.14	Proof Key for Code Exchange (PKCE) Support in OAM	37-58
37.14.1	Enabling PKCE	37-59
37.14.2	PKCE Flow for Access Token Generation	37-62
37.15	Token Exchange Support in OAM	37-66
37.15.1	Enabling Token Exchange Support	37-66

38 Understanding OpenIDConnect

38.1	About OpenIDConnect Tokens	38-1
38.1.1	OpenIDConnect ID Token	38-1
38.2	Claims	38-3
38.3	Custom Claims	38-5
38.3.1	About Custom Claims	38-5
38.3.2	Defining Custom Claims Using Templates	38-6
38.3.2.1	About valueTransformation	38-9
38.3.2.2	About valueFiltering	38-11
38.3.2.3	About defaultValue	38-12
38.3.2.4	About dynamicParams	38-12
38.3.3	Using Claims Parameter in Authorization Request	38-13
38.3.4	Examples for Template-Based Claims	38-15
38.4	OpenIDConnect Authentication Flows in Oracle Access Manager	38-20
38.4.1	Understanding Authorization Code Grant Authentication Flow	38-21
38.4.2	Understanding Implicit Grant Authentication Flow	38-26
38.4.3	Understanding OpenIDConnect UserInfo Endpoint	38-29
38.4.3.1	Retrieving UserInfo Attributes from OAM	38-29
38.4.3.2	Retrieving User Info Attributes Using Template-Based Mapping	38-33
38.4.4	Understanding OpenIDConnect Discovery Endpoint	38-36
38.4.4.1	Configuring OpenIDConnect Discovery Endpoint	38-36
38.4.4.2	Configuring OIDC Discovery Endpoint	38-38
38.4.5	Fetching Identity Domain Certificate	38-39

39 OIDC Client Integrations with Social Identity Providers

39.1	About the OpenIDConnectPlugin	39-1
------	-------------------------------	------

39.2	Authentication Module and Scheme and Policy Changes	39-5
39.3	Authentication Scheme Changes	39-6
39.4	Policy Changes	39-7
39.5	Integration with IDCS	39-7
39.6	Integration with Google	39-10
39.7	Integration with Facebook	39-11

40 OAuth Just-In-Time (JIT) User Provisioning

40.1	Using Self-Registration for Just-in-Time User Provisioning	40-1
40.1.1	Prerequisites for Just-in-Time Provisioning by Self-Registration	40-1
40.1.2	About Self-Registration Page	40-3
40.1.3	Configuring UserSelfRegistration Authentication Module and Scheme	40-7
40.1.3.1	Just In Time UserSelfRegistration Authentication Module	40-7
40.1.3.2	JIT UserSelfRegistration Steps Orchestration	40-8
40.1.3.3	Just In Time UserSelfRegistration Authentication Scheme	40-8
40.1.4	Protecting Resources with UserSelfRegistrationScheme	40-9
40.2	Configuring Just-In-Time User Auto-Provisioning with Password Prompt	40-9
40.2.1	Prerequisites for Just-in-Time User Auto-Provisioning	40-10
40.2.2	Configuring AutoProvisioning Authentication Module and Scheme	40-11
40.2.2.1	Just-In-Time User Auto-Provisioning Authentication Module	40-11
40.2.2.2	JIT User Auto-provisioning Steps Orchestration	40-13
40.2.2.3	Just-In-Time User AutoProvisioningScheme	40-13
40.2.3	Protecting Resources with AutoProvisioningScheme	40-15
40.3	Configuring Just-In-Time User Auto-Provisioning (No Password Prompt)	40-15
40.3.1	Prerequisites for JIT User Auto-Provisioning (No Password Prompt)	40-15
40.3.2	Configuring JIT AutoProvisioning Authentication Module and Scheme (No Password Prompt)	40-17
40.3.2.1	Just-In-Time User Auto-Provisioning Authentication Module (No Password Prompt)	40-17
40.3.2.2	JIT User Auto-provisioning Steps Orchestration (No Password Prompt)	40-19
40.3.2.3	Just-In-Time User AutoProvisioningScheme (No Password Prompt)	40-19
40.3.3	Protecting Resources with AutoProvisioningScheme (No Password Prompt)	40-20

Part X Using Identity Context

41 Using Identity Context

41.1	Introducing Identity Context	41-1
41.2	Understanding Identity Context	41-2
41.3	Working With the Identity Context Service	41-4

41.3.1	Identity Context Dictionary	41-4
41.3.2	Identity Context Runtime	41-8
41.4	Identity Context API	41-10
41.5	Configuring the Identity Context Service Components	41-12
41.5.1	Configuring Oracle Fusion Middleware	41-12
41.5.2	Configuring Access Manager	41-13
41.5.2.1	Identity Assertion	41-13
41.5.2.2	Federation Attributes	41-13
41.5.2.3	Session Attributes	41-14
41.5.2.4	Identity Store Attributes	41-14
41.5.3	Configuring Web Service Security Manager	41-14
41.5.4	Configuring Oracle Entitlements Server	41-15
41.5.5	Configuring Oracle Enterprise Single Sign On	41-16
41.5.6	Configuring Secure Identity Context Propagation	41-17
41.6	Validating Identity Context	41-19

Part XI Using the OAM Snapshot Tool

42 Introduction to the OAM Snapshot Tool

43 Customizing the Paths for the OAM Snapshot Tool

43.1	Configuring OAM Snapshot Tool when the DB is on a Different Infrastructure	43-1
------	--	------

44 Performing Backup and Restore and Migration of OAM System Using Snapshot Tool

44.1	Backing Up an OAM Deployment	44-1
44.1.1	Prerequisites for Backing up an OAM Deployment	44-2
44.1.2	Creating a backup.properties file	44-2
44.1.3	Running the createSnapshot.sh script	44-8
44.1.4	Backing up Keystores and Custom Applications	44-11
44.2	Rebuilding an OAM Deployment - Domain Only	44-11
44.2.1	Installing the OAM Software in the New Infrastructure	44-11
44.2.2	Creating a restore.properties file	44-12
44.2.3	Running the applySnapshot.sh script	44-13
44.2.4	Restoring Keystores and Custom Applications	44-14
44.3	Migrating OAM Domain and Database Schemas	44-14
44.3.1	Installing and Creating a New Database	44-15

44.3.2	Installing the OAM Software in the New Infrastructure	44-15
44.3.3	Creating a restore.properties file	44-16
44.3.4	Running the applySnapshot.sh script	44-24
44.3.5	Restoring Keystores and Custom Applications	44-25

45 Using the OAM Snapshot Tool

45.1	Upgrading OAM Using Snapshot Tool	45-1
45.2	Rebuilding OAM Deployment in a Multi-Node Cluster	45-2

46 Troubleshooting

Part XII Integrating Access Manager with Other Products

47 Integrating RSA SecurID Authentication with Access Manager

47.1	Introduction to Access Manager and RSA SecurID Authentication	47-1
47.2	RSA Features Supported by Access Manager	47-1
47.3	Components Required for SecurID Authentication	47-3
47.3.1	Supported Versions and Platforms	47-3
47.3.2	Required RSA Components	47-3
47.3.2.1	RSA Authentication Manager	47-3
47.3.2.2	RSA SecurID Tokens	47-4
47.3.3	Installation and Configuration Requirements	47-4
47.4	SecurID Authentication Modes	47-5
47.4.1	Standard SecurID Authentication	47-5
47.4.2	SecurID Next Tokencode Authentication	47-6
47.4.3	SecurID New PIN Authentication	47-7
47.5	Configuring Access Manager for RSA SecurID Authentication	47-7
47.6	Running a Custom RSA Plug-in	47-11

48 Configuring Access Manager for Windows Native Authentication

48.1	Introducing Access Manager with Windows Native Authentication	48-1
48.1.1	Understanding Access Manager WNA Login and Fall Back Authentication	48-2
48.1.1.1	Successful Access Manager WNA Authentication	48-3
48.1.1.2	Access Manager WNA Fallback Authentication	48-4
48.1.2	Supported Kerberos Authentication Modules	48-5
48.2	About Preparing Your Active Directory and Kerberos Topology	48-5

48.2.1	Preparing Active Directory and Kerberos	48-8
48.3	Confirming Access Manager Operations	48-11
48.4	Enabling the Browser to Return Kerberos Tokens	48-12
48.4.1	Enabling Kerberos Tokens in Internet Explorer	48-12
48.4.2	Enabling Kerberos Tokens in Mozilla Firefox	48-12
48.4.3	Enabling Kerberos Tokens in Edge or Chrome	48-13
48.5	Integrating KerberosPlugin with Oracle Virtual Directory	48-13
48.5.1	Preparing Oracle Virtual Directory for Integration	48-13
48.5.2	Registering Oracle Virtual Directory as the Default Store for WNA	48-14
48.5.3	Setting Up Authentication with Access Manager KerberosPlugin and OVD	48-15
48.6	Integrating the KerberosPlugin with Search Failover	48-16
48.6.1	Registering Microsoft Active Directory Instances with Access Manager	48-17
48.6.2	Setting Up the KerberosPlugin for ADGCs	48-18
48.7	Configuring Access Manager for Windows Native Authentication	48-20
48.7.1	Creating the Authentication Scheme for Windows Native Authentication	48-20
48.7.2	Configuring Policies for Windows Native Authentication	48-21
48.7.3	Configuring WNA for NTLM Fallback	48-21
48.7.4	Configuring WNA Fallback to FORM-based Authentication Scheme	48-22
48.7.5	Verifying the Access Manager Configuration File	48-24
48.8	Validating WNA with Access Manager Protected Resources	48-25
48.9	Configuring WNA For Use With DCC	48-25
48.9.1	Initializing the Kerberos Protocol	48-26
48.9.2	Configuring Access Manager	48-27
48.10	Troubleshooting WNA Configuration	48-28
48.10.1	Kinit Fails	48-28
48.10.2	"An Incorrect Username or Password was Specified" Is Displayed	48-29
48.10.3	User Identity Store is Not Registered Correctly	48-29
48.10.4	Two BASIC Authentication Prompts Are Displayed	48-29

49 Integrating Microsoft SharePoint Server with Access Manager

49.1	What is Supported in This Release?	49-1
49.2	Introduction to Integrating With the SharePoint Server	49-2
49.2.1	About Windows Impersonation	49-3
49.2.2	Form Based Authentication With This Integration	49-3
49.2.3	Authentication With Windows Impersonation and SharePoint Server Integration	49-4
49.2.4	Access Manager Support for Windows Native Authentication	49-4
49.3	Integration Requirements	49-5
49.3.1	Requirements Confirmation	49-5
49.3.2	Required Access Manager Components	49-6
49.3.3	Required Microsoft Components	49-7

49.4	Preparing for Integration With SharePoint Server	49-9
49.5	Integrating With Microsoft SharePoint Server	49-11
49.5.1	Creating a New Web Application in Microsoft SharePoint Server	49-11
49.5.2	Creating a New Site Collection for Microsoft SharePoint Server	49-13
49.6	Setting Up Microsoft Windows Impersonation	49-14
49.6.1	Creating Trusted User Accounts	49-15
49.6.2	Assigning Rights to the Trusted User	49-16
49.6.3	Binding the Trusted User to Your WebGate	49-17
49.6.4	Adding an Impersonation Response to an Authorization Policy	49-18
49.6.5	Adding an Impersonation DLL to IIS	49-18
49.6.5.1	Configuring and Registering ImpersonationModule to IIS.	49-19
49.6.6	Testing Impersonation	49-21
49.6.6.1	Creating an IIS Virtual Site Not Protected by SharePoint Server	49-21
49.6.6.2	Testing Impersonation Using the Event Viewer	49-22
49.6.6.3	Testing Impersonation using a Web Page	49-22
49.6.6.4	Negative Testing for Impersonation	49-23
49.7	Completing the SharePoint Server Integration	49-23
49.7.1	Configuring IIS Security	49-23
49.8	Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider	49-24
49.8.1	About Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider	49-25
49.8.2	Installing Access Manager for Microsoft SharePoint Server Configured With LDAP Membership Provider	49-26
49.8.3	Configuring an Authentication Scheme for Use With LDAP Membership Provider	49-27
49.8.4	Integrating SharePoint Server with OAM 12c using FBA	49-28
49.8.5	Ensuring Directory Servers are Synchronized	49-30
49.8.6	Testing the Integration	49-31
49.9	Configuring Single Sign-On for Office Documents	49-31
49.10	Configuring Single Sign-off for Microsoft SharePoint Server	49-31
49.10.1	Configuring a Custom Logout URL in SharePoint Server	49-32
49.10.2	Configuring Logout in SharePoint Server With Impersonation	49-32
49.11	Setting Up Access Manager and Windows Native Authentication	49-33
49.11.1	Setting Up Access Manager WNA	49-33
49.11.2	Setting Up WNA With SharePoint Server	49-33
49.11.3	Installing Access Manager for WNA and SharePoint Server	49-34
49.11.4	Testing Your WNA Implementation	49-35
49.12	Synchronizing User Profiles Between Directories	49-36
49.13	Testing Your Integration	49-36
49.13.1	Testing the SharePoint Server Integration	49-36
49.13.2	Testing Single Sign-On for the SharePoint Server Integration	49-36

49.14	Troubleshooting	49-37
49.14.1	Internet Explorer File Downloads Over SSL Might Not Work	49-37

50 Integrating Access Manager with Outlook Web Application

50.1	What is New in This Release?	50-1
50.2	Introduction to Integration with Outlook Web Application	50-1
50.2.1	About Impersonation Provided by Microsoft Windows	50-2
50.2.2	Access Manager 12c Support for Windows Impersonation	50-2
50.2.3	Single Sign-On for Authenticated Access Manager Users into Exchange	50-3
50.2.4	Confirming Requirements	50-3
50.3	Enabling Impersonation With a Header Variable	50-3
50.3.1	Requirements for Impersonation with a Header Variable	50-4
50.3.2	Creating an Impersonator as a Trusted User	50-4
50.3.3	Assigning Rights to the Trusted User	50-5
50.3.4	Binding the Trusted User to Your WebGate	50-6
50.3.5	Adding an Impersonation Response to An Application Domain	50-7
50.3.6	Adding an Impersonation DLL to IIS	50-7
50.3.7	Testing Impersonation	50-8
50.3.7.1	Creating an IIS Virtual Site	50-8
50.3.7.2	Testing Impersonation Using the Event Viewer	50-9
50.3.7.3	Testing Impersonation using a Web Page	50-9
50.4	Setting Up Impersonation for Outlook Web Application (OWA)	50-10
50.4.1	Prerequisites to Setting Impersonation for Outlook Web Application	50-11
50.4.2	Creating a Trusted User Account for Outlook Web Application	50-11
50.4.3	Assigning Rights to the Outlook Web Application Trusted User	50-12
50.4.4	Binding the Trusted Outlook Web Application User to Your WebGate	50-12
50.4.5	Adding an Impersonation Action to an Application Domain for Outlook Web Application	50-13
50.4.6	Adding an Impersonation dll to IIS	50-14
50.4.7	Configuring IIS Security	50-14
50.4.8	Testing Impersonation for Outlook Web Application	50-15
50.4.8.1	Testing Impersonation Using the Event Viewer	50-15
50.4.8.2	Testing Impersonation using a Web Page	50-16
50.4.8.3	Conducting Negative Testing for Impersonation	50-16
50.5	Setting Up Access Manager WNA for Outlook Web Application	50-17

51 Integrating Access Manager with SAP NetWeaver Enterprise Portal

51.1	What is Supported in This Release?	51-1
51.2	Supported Versions and Platforms	51-1
51.3	Integration Architecture	51-1

51.3.1	Process Overview: Integration with SAP NetWeaver Enterprise Portal	51-2
51.4	Configuring Oracle Access Management and NetWeaver Enterprise Portal 7.0.x	51-3
51.4.1	Before You Begin Configuring OAM and NetWeaver Enterprise Portal 7.0.x	51-3
51.4.2	Configuring the Apache HTTP Server as a Proxy	51-4
51.4.3	Configuring SAP NetWeaver Enterprise Portal for External Authentication	51-5
51.4.4	Adjusting the Login Module Stacks for using Header Variables	51-6
51.4.5	Configuring Access Manager for SAP Enterprise Portal	51-7
51.5	Configuring Oracle Access Management and NetWeaver Enterprise Portal 7.4.x	51-8
51.5.1	Before You Begin Configuring OAM and NetWeaver Enterprise Portal 7.4.x	51-8
51.5.2	Configuring Access Manager for SAP NetWeaver Enterprise Portal 7.4.x	51-9
51.5.3	Configuring Apache Web Server 2.0.x or 2.2.x	51-11
51.5.4	Configuring SAP Enterprise Portal 7.4 for External Authentication	51-12
51.5.5	Adjusting the Login Module Stacks for Using Header Variables	51-13
51.6	Testing the Integration	51-13
51.7	Troubleshooting the Integration	51-14

Part XIII Appendixes

A Integrating Oracle ADF Applications with Access Manager SSO

A.1	Introducing Oracle Platform Security Services and Oracle Application Developer Framework	A-1
A.1.1	Oracle Platform Security Services Single Sign-on Framework	A-1
A.1.2	Oracle Application Developer Framework	A-2
A.2	Integrating Access Manager With Web Applications Using Oracle ADF Security and the OPSS SSO Framework	A-3
A.2.1	Sample SSO Configuration for Access Manager	A-5
A.2.2	SSO Provider Configuration Details	A-6
A.3	Configuring Centralized Logout for Oracle ADF-Coded Applications	A-7
A.3.1	Configuring Centralized Logout for ADF-Coded Applications with Access Manager	A-8
A.4	Confirming Application-Driven Authentication During Runtime	A-9

B Securing Communication

B.1	Prerequisites to Setting up a Secure Communication between OAM Servers and Webgates	B-1
B.2	Securing Communication Between OAM Servers and WebGates	B-1
B.2.1	About Certificates, Authorities, and Encryption Keys	B-4
B.2.2	About Security Modes and X509Scheme Authentication	B-4
B.2.3	The Importcert Tool	B-5
B.2.4	TLS 1.3 and TLS 1.2 Support in Oracle Access Management	B-6

B.2.5	Generating Client Keystores for OAM Tester in Cert Mode	B-15
B.3	Securing Communication between OAM Servers and WebGates using OAP over REST	B-16
B.3.1	About OAP over REST Communication	B-16
B.3.2	Configuring Load Balancer using Oracle mod_wl_proxy on OHS	B-17
B.3.3	Configuring Work Manager for OAP over REST	B-18
B.3.4	Configuring HTTP and HTTPS Communication between WebGate and Access Manager	B-19
B.3.4.1	Enabling two-way SSL for OAP over REST	B-21
B.3.5	Connection Tuning for OAP over REST	B-23
B.3.6	Troubleshooting OAP over REST	B-24
B.3.6.1	Error Performing Libcurl Operation	B-24
B.4	Enabling FIPS Mode on Oracle Access Management	B-25
B.4.1	Enabling FIPS Mode on OAM Server	B-25
B.4.2	Configuring SAML Federation for FIPS	B-26
B.4.3	Enabling FIPS Mode on OAM Clients	B-28
B.5	Configuring Cert Mode Communication for Access Manager	B-29
B.5.1	About Cert Mode Encryption and Files	B-30
B.5.2	Generating a Certificate Request and Private Key for OAM Server	B-30
B.5.3	Retrieving the .OAMKeystore password stored in UDM	B-31
B.5.4	Importing the Trusted, Signed Certificate Chain Into the Keystore	B-32
B.5.5	Adding Certificate Details to Access Manager Settings	B-34
B.5.6	Generating a Private Key and Certificate Request for WebGates	B-35
B.5.7	Supporting Two-Way SSL for CERT Mode Communication	B-35
B.5.8	Updating WebGate to Use Certificates	B-36
B.5.9	About WebGate Usage of PFS and Approved Cipher Suites for OAP Simple/ Cert Mode Communication	B-37
B.6	Configuring Simple Mode Communication with Access Manager	B-38
B.6.1	About Simple Mode, Encryption, and Keys	B-39
B.6.2	Retrieving the Global Passphrase for Simple Mode	B-39
B.6.3	Updating WebGate Registration for Simple Mode	B-40
B.6.4	Verifying SIMPLE Mode Configuration	B-41
B.7	Configuring SSL in IHSWebServer	B-42

C Troubleshooting

C.1	Introduction to Oracle Access Management Troubleshooting	C-2
C.1.1	System Analysis and Problem Scenarios	C-2
C.1.2	LDAP Server or Identity Store Issues	C-3
C.1.3	OAM Server or Host Issues	C-4
C.1.4	Agent-Side Configuration and Load Issues	C-5
C.1.5	Runtime Database (Audit or Session Data) Issues	C-5

C.1.6	Change Propagation or Activation Issues	C-6
C.1.7	Policy Store Database Issues	C-6
C.2	My Oracle Support for Additional Troubleshooting Information	C-6
C.3	Administrator Lockout	C-7
C.4	Error During Federation Configuration After Upgrade from PS1 to PS2	C-7
C.5	Oracle Access Management Console Inconsistent State	C-8
C.6	AdminServer Won't Start if the Wrong Java Path Given with WebLogic Server Installation	C-8
C.7	Agent Naming Not Unique	C-9
C.8	Application URL Requirements	C-9
C.9	Authentication Issues	C-9
C.9.1	Anonymous Authentication Issues	C-9
C.9.2	X.509Scheme and SSL Handshake Issues	C-10
C.9.2.1	Configuration Issues	C-10
C.9.2.2	Trust Issues	C-11
C.9.2.3	Certificate Validation Issues	C-11
C.9.3	X.509 Protected Resource and Single Sign Off	C-11
C.9.4	X509CredentialExtractor Certificate Validation Error	C-11
C.10	Authorization Issues	C-12
C.10.1	Authorization Condition Error	C-12
C.10.2	LDAP Search Filter Test Results	C-12
C.10.3	Authorization Header Response Names	C-13
C.11	Cannot Access Authentication LDAP or Database	C-13
C.12	Cannot Find Configuration	C-13
C.12.1	Configuration Does Not Exist ...	C-13
C.13	OAM unsupported Whole Server Migration	C-13
C.14	Could Not Find Partial Trigger	C-14
C.15	Denial of Service Attacks	C-14
C.15.1	Protecting the OAM Server from Crashing Under Load	C-15
C.15.2	Compensating for Network Latency	C-15
C.15.3	Protecting OAM Servers from a Flood of HTTP Requests	C-16
C.16	Diagnosing Initialization and Performance Issues	C-16
C.16.1	Diagnosing an Initialization Issue	C-17
C.16.2	Diagnosing a Performance Issue	C-17
C.16.3	Diagnosing Out-of-Memory Issues With a Heap Dump	C-17
C.17	Disabling Windows Challenge/Response Authentication on IIS Web Servers	C-18
C.18	Changing UserIdentityStore1 Type Can Lock Out Administrators	C-18
C.19	IIS Web Server Issues	C-18
C.19.1	Form Authentication or Pass-Through Not Working	C-19
C.19.2	IIS and General Web Component Guidelines	C-19
C.19.3	Issues with IIS v6 Web Servers	C-19

C.19.4	Page Cannot Be Displayed Error	C-20
C.19.5	Removing and Reinstalling IIS DLLs	C-21
C.20	Import and File Upload Limits	C-21
C.21	jps Logger Class Instantiation Warning is Logged on Authentication	C-22
C.22	Internationalization, Languages, and Translation	C-22
C.22.1	Automatically Generated Descriptions Are Not Translated	C-22
C.22.2	Console Looks Messy	C-22
C.22.3	Authentication Fails: Users with Non-ASCII Characters	C-22
C.22.4	Access Tester Does Not Work with Non-ASCII Agent Names	C-23
C.22.5	Locales, Languages, and Oracle Access Management Console Login Page	C-23
C.23	Login Failure for a Protected Page	C-23
C.24	OAM Metric Persistence Timer IllegalStateException: SafeCluster	C-24
C.25	Partial Cluster Failure and Intermittent Login and Logout Failures	C-24
C.26	RSA SecurID Issues and Logs	C-25
C.27	Registration Issues	C-25
C.27.1	Problem: Remote Registration Tool Failure	C-25
C.27.2	Problem: No ObAccessClient.xml File Generated	C-26
C.27.3	Problem: Partner Registration Failure	C-26
C.27.4	Problem: Remote Registration Failure in upgraded OAM12c environment	C-26
C.28	Rowkey does not have any primary key attributes Error	C-26
C.29	SELinux Issues	C-26
C.30	Session Issues	C-27
C.30.1	Session Impersonation Not Enabled by Default	C-27
C.30.2	Sessions with Oracle Access Manager 11.1.1 Integrated with Oracle Identity Federation 11.1.1	C-28
C.31	SSL versus Open Communication	C-28
C.32	Start Up Issues	C-28
C.32.1	AdminServer Startup (or Remote Registration Tool Failure) on AIX Platforms	C-28
C.33	Synchronizing OAM Server Clocks	C-29
C.34	Time delay in configuration change	C-30
C.35	Validation Errors	C-30
C.35.1	Resource not added to Authentication or Authorization Policy	C-30
C.35.2	Validation Failure - "description" attribute is not valid	C-31
C.36	Web Server Issues	C-31
C.36.1	Server Fails on an Apache Web Server	C-31
C.36.2	Apache v2 on HP-UX	C-32
C.36.3	Apache v2 Bundled with Red Hat Enterprise Linux 4	C-32
C.36.4	Apache v2 Bundled with Security-Enhanced Linux	C-32
C.36.5	Apache v2 on UNIX with the mpm_worker_module for Webgate	C-33
C.36.6	Domino Web Server Issues	C-34
C.36.7	Errors, Loss of Access, and Unpredictable Behavior	C-34

C.36.8	Known Issues for ISA Web Server	C-35
C.36.9	Oracle HTTP Server Fails to Start with LinuxThreads	C-35
C.36.10	Oracle HTTP Server Webgate Fails to Initialize On Linux Red Hat 4	C-36
C.36.11	Oracle HTTP Server Web Server Configuration File Issue	C-36
C.36.12	Issues with IIS v6 Web Servers	C-36
C.36.13	PCLOSE Error When Starting Sun Web Server	C-37
C.36.14	Removing and Reinstalling IIS DLLs	C-37
C.37	Windows Native Authentication	C-38
C.38	WLST Commands for Multi-Data Centers	C-38
C.38.1	enableMultiDataCentreMode	C-39
C.38.2	disableMultiDataCentreMode	C-40
C.38.3	addPartnerForMultiDataCentre	C-41
C.38.4	removePartnerForMultiDataCentre	C-42
C.38.5	setMultiDataCenterType	C-43
C.38.6	setMultiDataCenterWrite	C-43
C.38.7	setMultiDataCentreClusterName	C-44
C.38.8	validateMDCConfig	C-44
C.38.9	exportAccessStore	C-44
C.38.10	importAccessStore	C-45
C.39	Comparing Default Parameters and Values used in MDC Configuration for 11g and 12c	C-45
C.40	WADL Generation Does not Show Description	C-47
C.41	Safari Browser Does not Display Options Under the Configuration Tab of the OAM Console URL	C-47

List of Examples

11-1	User Login Failure	11-18
11-2	Directory Outage	11-19
37-1	Revoking All the OAuth Tokens for a User	37-41
37-2	Revoking All Refresh Tokens for a Client	37-42
37-3	Revoking Refresh Tokens for a User Based on Timestamp	37-42
38-1		38-9
38-2		38-9
38-3		38-10
38-4		38-10
38-5		38-11
38-6		38-12
38-7		38-12
38-8		38-13
38-9	Transforming Claim Value by Composing Different User Store Attributes	38-16
38-10	Transforming Users Group Allocation to Array and Filter Groups based on a Regular Expression	38-17
38-11	Mapping an OIDC Standard Claim in Userinfo to LDAP Attributes	38-17
38-12	Applying transformations and Filters to OIDC Standard Claim	38-18
38-13	Adding Custom Claims in AccessToken , IDToken and UserInfo	38-18
38-14	Adding Custom Claims as Client Configurations and Request Parameters	38-19
40-1	Sample Self-Registration Page	40-5

List of Figures

1-1	Oracle Access Management Overview	1-2
1-2	Access Manager Components and Services	1-4
1-3	Access Manager Component Distribution	1-5
2-1	Oracle Access Management Administrator Launch Pad	2-7
2-2	SSO Agent Search Page	2-12
3-1	Oracle Access Management Configuration Options	3-2
3-2	Available Services	3-3
3-3	Common Settings Page (Collapsed View)	3-5
3-4	Certificate Revocation List Dialog Box	3-8
3-5	OCSP/CDP Settings	3-9
5-1	Creating User Identity Store Registration	5-10
5-2	System Store Registration	5-14
5-3	Identity Directory Service Console Page	5-18
5-4	Create IDS Profile Page	5-20
5-5	Create IDS Repository Page	5-28
5-6	Add System Administrator Roles	5-30
6-1	OAM Server Registration Page with Proxy Tab Displayed	6-5
8-1	Audit to Database Architecture	8-4
8-2	Common Settings: Auditing Configuration	8-19
9-1	Log-Level Activation in the Default Log Configuration File	9-23
11-1	Server Processes Overview Page	11-3
11-2	OAM Server Metrics: Session Operations Monitoring Page	11-4
11-3	OAM Server Metrics: Server Operations Tab	11-5
11-4	OAM Server Metrics: WebGates Tab	11-6
11-5	OAM Metrics Table	11-8
12-1	Fusion Middleware Control (AS-Control) Deployment Architecture	12-2
12-2	OAM Farm Page in Fusion Middleware Control	12-4
12-3	Farm Navigation Tree in Fusion Middleware Control	12-5
12-4	Node Information Page in Fusion Middleware Control	12-6
12-5	Application Deployment Summary for the Selected Internal Application	12-6
12-6	Application Deployment Menu	12-7
12-7	WebLogic Server Domain Summary with Context Menu Exposed	12-8
12-8	Cluster Page	12-11
12-9	Key Metrics for Server Page	12-11
12-10	Aggregated Access Manager Component Metrics for the Cluster	12-15

12-11	Access Manager Component Metrics for a Single OAM Server Instance	12-15
12-12	Performance Summary Command	12-16
12-13	Performance Summary Page with Metric Palette	12-17
12-14	Access Manager Log Levels on the Log Configuration Tab	12-23
12-15	Log Levels for Security Token Service	12-23
12-16	Log Files Configuration Page	12-27
12-17	Typical Log Messages Page in Fusion Middleware Control	12-32
12-18	System MBean Browser and Attributes Tab	12-38
13-1	Access Manager Settings: Load Balancer	13-2
13-2	Access Manager Settings: Server Error Mode	13-3
13-3	Access Manager Settings: WebGate Traffic Load Balancer	13-6
13-4	Common Policy Evaluation Caches	13-13
15-1	Create OAM WebGate Page	15-2
15-2	Load Balanced Deployment	15-13
15-3	Expanded OAM WebGate Page with Defaults	15-22
15-4	WebGate Search Controls and Create Button	15-28
15-5	Key Generation	15-38
16-1	Global Session Details: Common Settings Page	16-10
16-2	Common Configuration: Session Management Page	16-15
17-1	Multi-Data Center System Architecture	17-2
17-2	Active-Active Deployment Mode	17-7
17-3	Active-Active Mode Failover	17-8
17-4	Multi-Data Center Deployment	17-9
17-5	Requests Served By Different Data Centers	17-13
17-6	Logout and Session Invalidation	17-15
17-7	Stretch Cluster Deployment	17-16
17-8	Traditional MDC Deployment	17-17
17-9	Active-Active Topology	17-18
17-10	Active-Active Topology Across Multiple Data Centers	17-19
17-11	Load Balancing Access Manager Components	17-20
17-12	Global Load Balancer Front Ends Local Load Balancer	17-21
19-1	Replication Flow	19-3
19-2	Starting Sequence Illustrated	19-8
19-3	Applying Custom Transformation Rules	19-12
21-1	Access Manager 12c Policy Model	21-7
21-2	Access Manager Shared Policy Components	21-7
21-3	Anatomy of Access Manager Policies	21-11

22-1	Default HTTP Resource Type Definition	22-4
22-2	Default Resource Type wl_authen	22-4
22-3	Default Resource Type TokenServiceRP Resource Type	22-5
22-4	Create Host Identifier Page	22-13
22-5	Native Kerberos Authentication Module	22-25
22-6	Native LDAP Authentication Module	22-26
22-7	Native X.509 Authentication Module	22-28
22-8	Access Manager Plug-ins for Customized Authentication Modules	22-34
22-9	Creating Custom Authentication Modules: General	22-35
22-10	Adding a Step and Associating a Plug-in	22-36
22-11	Plug-in Based Authentication Module Steps and Details	22-44
22-12	Steps Orchestration for Plug-in Based Authentication Modules	22-44
22-13	KerberosPlugin	22-46
22-14	Default KerberosPlugin Steps and Details	22-46
22-15	Default KerberosPlugin Steps and Orchestration	22-47
22-16	LDAPPlugin	22-47
22-17	Default LDAPPlugin Steps and Details	22-48
22-18	Default Orchestration of Steps for LDAPplugin	22-48
22-19	X509Plugin	22-49
22-20	X509Plugin Default Steps and Details	22-49
22-21	Default Orchestration for X509Plugin Steps	22-50
22-22	Password Policy Validation Module Plug-ins	22-51
22-23	Steps Orchestration: Password Policy Validation Plug-ins	22-51
22-24	Sample Authentication Scheme Page	22-70
22-25	Plug-ins Page	22-73
22-26	Plugin Details: Activation Status of Selected Plug-in	22-76
22-27	Default LDAPScheme Page	22-77
23-1	SSO Log-in with Embedded Credential Collector and OAM Agents	23-5
23-2	Example: Separate Resource WebGate and DCC WebGate Deployment	23-8
23-3	Combined DCC and WebGate Configuration	23-9
23-4	OAP Tunneling with DCC	23-15
23-5	Enable SSL	23-17
23-6	Keystore Configuration	23-18
23-7	Add Private Key Alias	23-18
23-8	SSL Advanced Options	23-19
23-9	New X509 Scheme	23-22
24-1	Password Policy Configuration Page	24-4

24-2	Password Policy Validation Authentication Module with Orchestrated Plug-ins	24-21
24-3	Step Orchestration for Password Policy Validation Module	24-22
24-4	Server Error Mode for Password Management	24-28
25-1	Application Domains Search Page	25-5
25-2	Example Application Domain Summary Page	25-6
25-3	Search Results for Resources in an Application Domain	25-7
25-4	Authentication Policies Tab	25-8
25-5	Authentication Policy Page: Resources and Responses	25-8
25-6	Authorization Policies Page	25-9
25-7	Individual Authorization Policy Page	25-10
25-8	Individual Authorization Policy Resources tab	25-10
25-9	Token Issuance Policies Page	25-11
25-10	Create Resource Page in the Application Domain	25-16
25-11	HTTP Resources, Query String Resource URL Controls	25-27
25-12	Resource Search within an Application Domain	25-31
25-13	Sample Authentication Policies Page in the Application Domain	25-34
25-14	Sample Individual Authentication Policy Page	25-35
25-15	Sample Individual Authorization Policy Page	25-39
25-16	Individual Authorization Policy Conditions Tab	25-45
25-17	Add Condition Window	25-48
25-18	Condition Containers on the Authorization Policy Page	25-49
25-19	Add Identities Window	25-51
25-20	Identity Condition and Details	25-52
25-21	Add Search Filter Controls	25-53
25-22	Identity Conditions: Details	25-54
25-23	IP4 Range Conditions	25-58
25-24	Temporal Condition Type Details Page	25-60
25-25	Attribute Conditions Page	25-63
25-26	Add Attribute Condition Dialog	25-63
25-27	Authorization Policy Rules Tab: Simple Mode	25-68
25-28	Rules Tab: Expression Rule Mode	25-70
25-29	Adding a Resource Prefix for Policy Ordering	25-74
25-30	Authorization Policy Response in the Console	25-76
25-31	Simple Response Samples	25-79
25-32	Complex Response Sample	25-80
26-1	OAM Agent (PEP) and OAM Server (PDP) Inter-operability	26-4
26-2	User Interactions with the Access Tester	26-7

26-3	Access Tester Console	26-13
26-4	Server Connection Panel in the Access Tester	26-17
26-5	Protected Resource URI Panel in the Access Tester	26-20
26-6	Access Tester User Identity Panel	26-23
26-7	Test Case Workflow	26-28
28-1	Available Services Page	28-28
29-1	New Identity Provider Page, Service Details Loaded from Metadata	29-3
29-2	New Identity Provider Page, Service Details entered Manually	29-4
29-3	Searching for Identity Providers	29-10
29-4	Attribute Sharing Plug-in Design	29-29
30-1	Identity Federation Service Settings Page	30-2
30-2	Keystore Settings	30-5
31-1	FederationScheme	31-2
31-2	FederationPlugin Steps	31-4
31-3	FederationPlugin Orchestration	31-4
31-4	Setting Up the Authentication Policy with FederationScheme	31-7
31-5	OIFScheme	31-8
31-6	OIFMTLDAPPlugin	31-9
31-7	Authorization Policy Response Tab	31-11
31-8	Adding a Federation Response Attribute to an AuthZ Policy	31-14
33-1	Second Factor Authentication Preferred Method Page	33-2
33-2	One Time Password Login Page	33-4
33-3	Access Request Notification Preferred Method Page	33-6
33-4	Access Request Notification Wait Screen	33-7
36-1	OAuth 3-Legged Flow Diagram	36-4
37-1	Diagram Showing the Flow for Getting Registration Token	37-21
37-2	Diagram Showing the Flow for Client Registration	37-24
37-3	Use case flow for SSO Session Linking for OAuth Tokens	37-27
40-1	Sample Authentication Scheme Page	40-9
40-2	Sample Authentication Scheme Page	40-15
40-3	Sample Authentication Scheme Page	40-20
41-1	End to End Identity Context Process	41-3
41-2	End To End Identity Context Process Components	41-4
41-3	Identity Context Process Flow	41-9
49-1	Setting up a Trusted User Account for Windows Impersonation	49-16
49-2	Configuring Rights for the Trusted User in Windows Impersonation	49-17
49-3	Verifying Event Viewer Settings	49-22

49-4	Impersonation Authentication	49-24
50-1	Setting up a Trusted User Account for Windows Impersonation	50-5
50-2	Configuring Rights for the Trusted User in Windows Impersonation	50-6
50-3	Verifying Event Viewer Settings	50-9
50-4	Impersonation Authentication	50-15
B-1	Communication Channels for OAM Servers and WebGates	B-3
B-2	mod_wl_proxy as Load Balancer	B-17

List of Tables

1-1	Access Manager Deployment Types	1-5
1-2	Oracle Access Management Post-Installation Tasks	1-8
2-1	Language Codes For Login Pages	2-15
2-2	Oracle Access Management Language Selection Methods	2-16
2-3	OAM_LANG_PREF Cookie	2-17
2-4	Application Integration for Language Preference	2-17
3-1	Configuration Options	3-2
3-2	Common Services	3-4
3-3	Common Settings	3-5
3-4	OCSP Responder Configuration Options	3-11
4-1	Roles for Delegating Administration	4-2
5-1	Data Sources for Oracle Access Management	5-1
5-2	Data Sources for Oracle Access Management Services	5-2
5-3	Components That Use Identity Stores	5-8
5-4	User Identity Store Elements	5-10
5-5	Access Manager Keys and Storage	5-35
5-6	Keystores for Access Manager and Security Token Service	5-35
6-1	Conditions Requiring Server Restart	6-4
6-2	OAM Server Instance Settings	6-5
6-3	OAM Proxy Settings for an Individual OAM Server	6-6
7-1	Logging Files	7-2
7-2	Logging Defaults	7-2
7-3	Oracle Access Management Server-Side Component Loggers	7-3
7-4	Oracle Access Management Shared-Service Engine Component Loggers	7-3
7-5	Oracle Access Management Foundation API Component Loggers	7-4
7-6	Mapping of ODL to Java Levels	7-5
7-7	Oracle Identity Federation Loggers	7-9
8-1	Oracle Business Intelligence Enterprise Edition Reports for OAM	8-6
8-2	Access Manager Administrative Audit Events	8-7
8-3	Access Manager Run-time Audit Events	8-10
8-4	Categories of Audit Events for Identity Federation	8-13
8-5	Identity Federation Session Management Events	8-14
8-6	Protocol Flow Events for Identity Federation	8-14
8-7	Server Configuration Identity Federation	8-15
8-8	Security Events for Identity Federation	8-15

8-9	Audit Configuration Elements	8-20
9-1	Logging Levels	9-2
9-2	Log Configuration File Names for Components	9-5
9-3	Log Writers	9-10
9-4	Global Parameters in the First Compound List	9-12
9-5	Factors that Determine Whether Logging Is Active	9-17
9-6	Mandatory Log Configuration File Parameters	9-18
9-7	Log Data File Configuration Parameters	9-19
9-8	ParamName Values You Can Configure for Per-Module Logging Threshold	9-25
10-1	Accounts_Locked_Out Report Fields	10-3
10-2	AuthenticationFromIPByUser Report Fields	10-4
10-3	AuthenticationPerIP Report Fields	10-4
10-4	Authentication_statistics Report Fields	10-4
10-5	AuthenticationStatisticsPerServer Report Fields	10-4
10-6	All Errors and Exceptions Report Fields	10-5
10-7	Authentication Failures Report Fields	10-5
10-8	Authentication History Report Fields	10-6
10-9	Authorization History Report Fields	10-6
10-10	Multiple Logins From Same IP Report Fields	10-6
11-1	OAM Proxy Metrics	11-6
11-2	OAM Proxy Tuning Parameters	11-7
11-3	Health Check REST API Parameters	11-14
12-1	Farm Page Sections	12-4
12-2	Resulting Pages for Selected Nodes and Targets	12-10
12-3	Summary of Performance Overviews in Fusion Middleware Control	12-12
12-4	Access Manager Component Metrics	12-15
12-5	Status and Controls on Performance Summary Pages	12-17
12-6	OAM Log Availability and Functions in Fusion Middleware Control	12-22
12-7	Log Levels Tab on Log Configuration Page	12-24
12-8	Log Files Elements	12-28
12-9	OAM Log Message Search Controls in Fusion Middleware Control	12-32
12-10	System MBean Browser	12-37
12-11	MBeans that Access Manager and Security Token Service Deploy	12-38
12-12	System MBean Browser	12-39
13-1	Access Manager Settings: Load Balancer	13-2
13-2	Server Error Mode	13-4
13-3	Error Trigger Condition, Modes, and Message Codes	13-4

13-4	External Error Codes, Trigger Conditions, and Recommended Messages	13-5
13-5	Access Manager Settings: WebGate Traffic Load Balancer	13-7
13-6	Access Manager Settings: SSO	13-8
13-7	Summary: Simple and Cert Mode	13-9
13-8	Server Common OAM Proxy Secure Communication Settings	13-11
13-9	Policy Evaluation Caches	13-13
13-10	Policy Cache Parameters	13-14
14-1	Agent Types	14-2
14-2	Agent Registration and SSO Support	14-2
14-3	Run Time Processing Overview for Access Manager	14-4
14-4	Keys and Policies Generated During Agent Registration	14-5
14-5	Artifacts Associated with Agent Registration	14-6
14-6	Copying Generated Artifacts	14-7
14-7	Remote Registration Methods	14-7
14-8	Agent Registration and Configuration Update Artifacts	14-9
15-1	Elements on Create Pages for OAM Agents	15-3
15-2	User-Defined WebGate Parameters	15-6
15-3	Elements on Expanded OAM WebGate/Access Client Registration Pages	15-23
15-4	Agent Search Controls	15-28
15-5	Environment Variables to Set within oamreg	15-31
15-6	Remote Registration Command Arguments: mode	15-31
15-7	Remote Registration Command Samples	15-32
15-8	Common Elements in Remote Registration Requests	15-33
15-9	Remote Registration Request Templates for OAM Agents	15-39
15-10	Elements in Extended OAM Agent Remote Registration Requests	15-40
15-11	Variables Required for Remote Registration	15-44
15-12	Files Returned by in-band Administrator to out-of-band Administrator	15-47
15-13	Remote Agent Update Modes and Input Files	15-49
15-14	Delta: OAM Agent Update versus Registration Request	15-49
16-1	Features Supported when the Database is Unavailable	16-2
16-2	Features Not Supported when the Database is Unavailable	16-2
16-3	Session Lifecycle States	16-5
16-4	Session Checks for State Changes	16-5
16-5	Session Removal	16-6
16-6	Application Domain-Specific Overrides	16-7
16-7	Session Content: Single Authentication Scheme	16-8
16-8	Session Outcomes: Multiple Authentication Schemes	16-8

16-9	Global Session Settings	16-11
16-10	Default Polling Interval	16-12
16-11	Application-Specific Session Timing Overrides	16-13
16-12	Session Management Controls and the Results Table	16-15
17-1	Multi-Data Center Policy Configurations for Idle Timeout	17-22
17-2	Session Synchronization and Failover Scenarios	17-23
18-1	MDC Use Cases	18-2
19-1	Replication States	19-3
19-2	Modifying Replication Agreement Properties	19-10
21-1	Summary: SSO Components	21-2
21-2	Introduction to SSO Implementations	21-3
21-3	Access Manager Global, Shared Policy Components	21-8
21-4	Access Manager Policy Components	21-10
21-5	Condition Types	21-14
21-6	SSO Cookies	21-15
22-1	Resource Type Definition	22-5
22-2	Host Identifiers Examples	22-7
22-3	Host Identifier Definitions	22-13
22-4	Comparing the DCC and ECC	22-19
22-5	Native Authentication Modules	22-24
22-6	Native Kerberos Authentication Module Definition	22-26
22-7	Native LDAP Authentication Modules Definition	22-27
22-8	X509 Authentication Module Definition	22-28
22-9	Simple Form versus Multi-Step Authentication	22-32
22-10	General tab	22-35
22-11	Add New Step Entries, Steps Results Table, and Details Section	22-36
22-12	Parameter Details for Various Plug-ins	22-37
22-13	Steps Orchestration Tab	22-45
22-14	X509 Step Details (KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT)	22-50
22-15	Steps and Plug-ins in a Customized Step-up Authentication Module	22-56
22-16	UserIdentification Step	22-66
22-17	X509CredentialExtractor Step	22-66
22-18	X509 Step Orchestration for ECU	22-69
22-19	Custom Plug-ins Actions	22-74
22-20	Plugins Status Table	22-75
22-21	Example of Plugin Details Extracted from XML Metadata File	22-76
22-22	Authentication Scheme Definition	22-77

22-23	Pre-configured Authentication Schemes	22-81
22-24	Challenge Parameters in Pre-configured Schemes	22-90
22-25	User-Defined Challenge Parameters for Authentication Schemes	22-91
22-26	Advanced Rules Attributes	22-99
22-27	Sample Advanced Rules	22-101
22-28	Request Context Data	22-101
22-29	Location Context Data	22-102
22-30	Session Context Data	22-102
22-31	User Context Data	22-103
22-32	Challenge Parameters for 11g Encrypted Cookies	22-104
22-33	Resource Webgate Support of POST Data Preservation and Restoration	22-106
22-34	Parameters Required for Authentication POST Data Handling	22-108
22-35	ECC and DCC: Long URL Handling	22-112
22-36	Parameters Required for Long URL Handling	22-113
23-1	Login Processing with Access Manager-Protected Resources	23-3
23-2	DCC Deployment Support	23-7
24-1	Password Policy Configuration Parameters	24-3
24-2	Password Policy Elements	24-4
24-3	Specifying Credential Collectors and Related Forms for Authentication	24-6
24-4	Credential Collector Password Pages	24-11
24-5	Password Management Forms and Functions	24-11
24-6	Location of Oracle-provided LDIFs for LDAP Providers	24-17
24-7	Key Password Attributes in a Password Policy	24-18
24-8	User Password Step Details	24-22
24-9	Included LDIF Schema Files	24-31
25-1	Resource Definition Elements	25-16
25-2	HTTP Resources Sample URL Values	25-19
25-3	Supported Wildcards in Resource URL Patterns (Precedence Order)	25-21
25-4	Sample Resource URLs	25-22
25-5	Pattern Matching for Requested URLs	25-24
25-6	Query String Matching: Examples	25-24
25-7	Resource Evaluation Outcomes	25-28
25-8	Search Elements for a Resource in an Application Domain	25-31
25-9	Authentication Policy Elements and Descriptions	25-35
25-10	Authorization Policy Elements and Descriptions	25-40
25-11	Authorization Policy Condition Tab	25-45
25-12	Add Condition Window Elements	25-48

25-13	Add identities Elements	25-51
25-14	Add Search Filter Elements	25-53
25-15	LDAP Search Filter Examples for Access Manager	25-55
25-16	Temporal Condition Details	25-60
25-17	Access Conditions that Require Attribute-Type Conditions	25-62
25-18	Attribute Condition Elements	25-64
25-19	Attribute Names for Request Built-ins	25-64
25-20	Attribute Names for Session Built-ins	25-64
25-21	Attribute Condition Data (Aggregation of Conditions)	25-65
25-22	Authorization Policy Rules Elements	25-69
25-23	Rule Tab in Expression Mode	25-71
25-24	Operators for Expressions in Authorization Rules	25-71
25-25	Response Elements	25-77
25-26	Namespace Request Variables for Single Sign-On	25-78
25-27	Namespace Session Variables for Single Sign-On	25-78
25-28	Namespace User Variables	25-78
25-29	Simple Responses and Descriptions	25-80
25-30	Complex Responses	25-80
25-31	Remote Policy Management Modes, Templates, and Flags	25-86
25-32	Remote Management Template Elements	25-89
33	UserIdentification Step	3
34	UserAuthentication Step	3
35	Passwordless Step	4
26-1	User Interactions: Tester Console Mode versus Command Line Mode Operations	26-8
26-2	Access Tester Supported System Properties	26-10
26-3	Access Tester Console Panels	26-14
26-4	Command Buttons in Access Tester Panels	26-14
26-5	Additional Access Tester Buttons	26-15
26-6	Access Tester Menus	26-16
26-7	Connection Panel Information	26-18
26-8	Protected Resource URI Panel Fields and Controls	26-21
26-9	Access Tester User Identity Panel Fields and Controls	26-23
26-10	Access Tester Capture Request Options	26-29
26-11	Generate Script Command	26-30
26-12	Test Script Control Parameters	26-31
26-13	Run Test Script Commands	26-33
26-14	Mismatched Results Reasons in the Statistics Document	26-36

27-1	Centralized Logout Circumstances	27-2
27-2	Logout Details After Registration (ObAccessClient.xml)	27-3
28-1	Supported SAML 2.0 NameID Formats	28-17
28-2	SAML 2.0 URLs for Identity Federation Acting As Identity Provider	28-20
28-3	SAML 2.0 URLs for Identity Federation Acting as Service Provider	28-20
28-4	Supported SAML 1.1 NameID Formats	28-22
28-5	SAML 1.1 URLs for Identity Federation Acting As Identity Provider	28-23
28-6	SAML 1.1 URL for Identity Federation Acting as Service Provider	28-23
28-7	OpenID 2.0 URLs for Identity Federation Acting As Identity Provider	28-25
28-8	OpenID 2.0 URLs for Identity Federation Acting as Service Provider	28-25
28-9	Configuring Identity Federation Settings	28-26
28-10	Implementing Identity Federation	28-27
29-1	Default Partner Profiles	29-2
29-2	Identity Provider Partner Settings	29-4
29-3	Attributes for Google OpenID Partner	29-8
29-4	Attributes for Yahoo OpenID Partner	29-8
29-5	Elements Used for IdP Provider Search	29-9
29-6	Service Provider Partner Settings	29-11
29-7	Sample SP Attribute Mappings	29-14
29-8	Attribute Mapping Value Expressions	29-15
29-9	Attribute Value Filtering Conditions	29-21
29-10	Sample IdP Attribute Mappings	29-24
29-11	Default Federation Authentication Method and Access Manager Authentication Scheme Mappings	29-25
29-12	Configuration Parameters for Attribute Sharing Plug-in	29-32
29-13	Session Attributes Accessible To Attribute Sharing Plug-in	29-33
30-1	Federation Settings in the Console	30-2
30-2	General Federation Settings	30-3
30-3	Federation Proxy Settings	30-4
30-4	Keystore Settings for Federation	30-5
31-1	FederationScheme Element Definitions	31-2
31-2	FederationPlugin Steps	31-4
31-3	Orchestration of FederationPlugin	31-5
31-4	OIFScheme Definition	31-8
31-5	IFMTLDAPPlugin Steps	31-9
31-6	Policy Response Elements	31-12
31-7	Message Attribute Mapping	31-23

31-8	Office 365 Service Provider Attribute Values	31-23
33-1	Adaptive Authentication Plugin Properties	33-11
33-2	Server Side Configuration for Adaptive Authentication Service	33-15
34-1	Location URL Parameter Definitions	34-2
34-2	Offline Configuration URL Parameters	34-3
37-1	Optional Parameters for Consent Management on MDC	37-15
37-2	Mandatory Property and Values for Creating the OAuth Client Template	37-19
37-3	Registration Token Sample Response	37-22
37-4	Registration Token Error Responses	37-22
37-5	Client Registration Error Responses	37-25
37-6	chaining_level Values and Behavior	37-36
37-7	Request Parameters	37-67
37-8	Response Parameters	37-68
38-1	Claims within the ID Token used by OpenIDConnect	38-2
38-2	Claims used by OpenIDConnect	38-3
38-3	Attributes and Values for Custom Claim Definition	38-7
38-4	Parameters used in the curl command for OpenIDConnect Authentication Flows	38-20
38-5	Authorization Code Grant Authentication flow: Parameters and Access tokens	38-23
38-6	Implicit Grant Authentication Flow: Parameters and Access tokens	38-27
38-7	scope values that are used to request Claims	38-29
38-8	Claims under each scope and the corresponding backend LDAP attribute.	38-30
38-9	Parameters to create new authentication module, UserInfoAuthModule	38-32
38-10	OIDC Standard Claims Mapping	38-34
38-11	Fetch public certificate of given Identity domain: Parameters	38-39
39-1	OpenIDConnectPlugin: Parameters for plugin configuration	39-2
39-2	UserIdentificationPlugin: Parameters to modify filters	39-6
40-1	OAuthUserSelfRegistrationPlugin Step	40-8
40-2	JIT Step Orchestration	40-8
40-3	OAuthUserSelfRegistrationPlugin Step for auto-provisioning	40-12
40-4	CredentialCollectorPlugin Step for Auto-Provisioning	40-12
40-5	JIT Step Orchestration	40-13
40-6	OAuthUserSelfRegistrationPlugin Step for auto-provisioning	40-18
40-7	JIT Step Orchestration	40-19
41-1	Identity Context Schema Attributes	41-5
41-2	Mapping Identity Context Operations	41-9
47-1	Access Manager Support for RSA Features	47-2
47-2	RSA Features Not Supported	47-2

48-1	Sample Naming	48-7
49-1	Component Requirements	49-6
49-2	Microsoft Requirements for this Integration	49-8
49-3	Create Web Application Options for Microsoft SharePoint Server	49-12
49-4	Create a Web Application to Host a Site Collection for SharePoint Server	49-14
50-1	Requirements for Impersonation with a Header Variable	50-4
51-1	Login Module Stacks for using Header Variables	51-6
51-2	Login Module Stacks for using Header Variables	51-13
A-1	addOAMSSOPProvider Command-line Arguments	A-4
B-1	importcert Command Syntax	B-5
B-2	Mandatory Configurations in WegGate User Defined Parameters Field	B-20
B-3	WebGate User Defined Configuration Parameters	B-21
B-4	PFS Cipher Suites	B-37
B-5	Supported Cipher Suites	B-38
C-1	oamMDC.properties Properties	C-39
C-2	partnerInfo.properties Properties	C-41

Preface

This guide provides information on administration and configuration tasks using Oracle Access Management.

Audience

This document is intended for Administrators who are familiar with:

- 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
- 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 that have purchased support 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.

Related Documents

Administering Oracle Access Management explains how to manage configuration and policies for Access Manager, Identity Federation and other available services. For more information, see the following documents in the Oracle Fusion Middleware 12c (12.2.1.4.0) documentation set:

- Oracle Access Management in *Release Notes for Oracle Identity Management*
- About the Oracle Identity and Access Management in *Installing and Configuring Oracle Identity and Access Management*—Explains how to use the Oracle

Universal Installer and the WebLogic Configuration Wizard for initial Access Manager 12c deployment.

- Configuring Oracle HTTP Server WebGate for Oracle Access Manager in *Installing WebGates for Oracle Access Manager*
- Developing a Custom User Provisioning Plug-in in *Developing Applications with Oracle Access Management*—Explains how to write custom applications and plug-ins to functions programmatically, to create custom Access Clients that protect non-Web-based resources.
- Introduction to Upgrading Oracle Identity and Access Management to 12c (12.2.1.4.0) in *Upgrading Oracle Identity and Access Management*
- Introduction in *Tuning Performance*
- Introduction to Oracle Fusion Middleware in *Administering Oracle Fusion Middleware*—Describes how to manage a secure Oracle Fusion Middleware environment, including how to change ports, deploy applications, and how to back up and recover Oracle Fusion Middleware.
- Understanding an Enterprise Deployment in *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management*—For a step-by-step guide to deployment.
- Introduction to High Availability in *High Availability Guide*—For high availability conceptual information as well as administration and configuration procedures for Administrators, developers, and others whose role is to deploy and manage Oracle Fusion Middleware with high availability requirements.
- Access Manager WLST Commands in *WebLogic Scripting Tool Command Reference for Identity and Access Management*—Provides details on customized Identity and Access Management WLST commands.
- Overview of Web Services Administration in *Administering 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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in This Guide?

This section summarizes the new features and significant changes in *Administering Oracle Access Management 12c (12.2.1.4.0)*

Follow the pointers into this guide to get more information about the features and how to use them.

- [Updates in October 2022 Documentation Refresh for 12c Release 2 \(12.2.1.4.0\)](#)
- [Updates in October 2021 Documentation Refresh for 12c Release 2 \(12.2.1.4.0\)](#)
- [Updates in April 2021 Documentation Refresh for 12c Release 2 \(12.2.1.4.0\)](#)
- [Updates in January 2021 Documentation Refresh for 12c Release 2 \(12.2.1.4.0\)](#)
- [Updates in October 2020 Documentation Refresh for 12c Release 2 \(12.2.1.4.0\)](#)
- [Updates in July 2020 Documentation Refresh for 12c Release 2 \(12.2.1.4.0\)](#)
- [Updates in April 2020 Documentation Refresh for 12c Release 2 \(12.2.1.4.0\)](#)
- **[Features of Oracle Access Management 12c Release 2 \(12.2.1.4.0\)](#)**
- [Features Not Supported in Access Manager](#)

Updates in October 2022 Documentation Refresh for 12c Release 2 (12.2.1.4.0)

This revision of Oracle® Fusion Middleware Administering Oracle Access Management contains feature updates and addresses bug fixes.

- **Federation partners support certificates using RSASSA-PSS signature algorithms**
Includes support for the Signature Algorithm SHA256-RSA-MGF1.

 **Note:**

This update is dependent on OWSM patch 34566592.

For details, see [Configuring RSA OAEP Key Transport Digest and MGF Digest](#).

Updates in October 2021 Documentation Refresh for 12c Release 2 (12.2.1.4.0)

This revision of Oracle® Fusion Middleware Administering Oracle Access Management contains feature updates and addresses bug fixes.

- **OAM SAML 2.0 Supported Encryption Algorithms**

OAM supports AES-GCM encryption modes.

For details, see [OAM SAML 2.0 Supported Encryption Algorithms](#) and [Changing Default Encryption Algorithm](#)
- **Two-way SSL for OAP over REST Communication.**

You can enable mutual authentication for OAP over REST between WebGate and OAM Server, therefore ensuring that the Server communicates with authentic clients.

For details, see [Enabling two-way SSL for OAP over REST](#)
- **TOTP-based Multi Factor Authentication in OAM**

You can configure MFA using the `configureMFA` command with `config-utility.jar`

For details, see [Configuring TOTP-based Multi Factor Authentication in OAM](#)
- **Token Signing Using Third-Party Certificates**

Access tokens can be signed using a self-signed key pair generated out-of-the-box. In this release, OAM extends the support to allow signing of access tokens using third-party key pairs.

For details, see [Token Signing Using Third-Party Certificates](#)
- **Mutual-TLS (mTLS) Client Authentication in OAM**

In TLS authentication, the server confirms its identity by producing a certificate (public key), which is then verified by the TLS verification process. In mTLS (mutual-TLS), along with the server, the client's identity is also verified. The TLS handshake is utilized to validate the client's possession of the private key corresponding to the public key in the certificate and to validate the corresponding certificate chain.

For details, see [Configuring Client Authentication](#) and [Configuring mTLS Client Authentication](#)
- **Custom Claims**

OAM extends the ability to define the custom claims using templates that can be configured at client or domain level. The custom claims can be included in all the access tokens, ID tokens and userinfo. You can also perform value transformation as well as value filtering of the custom claim.

For details, see [Custom Claims](#)
- **OAuth Access Token Maximum Size**

Default OAuth access token length limit has been increased to 7500. This value can be overridden using the OAuth Identity domain custom parameter: `accessTokenMaxLength`.
- **OAuth Client Update - Support for PATCH Request**

Introduces support for `PATCH` request during modification of OAuth clients. With `PATCH` operation, OAM appends existing scopes with values from the request. Similar behavior is provided for `redirect_uris`, `grant types`, and `custom attributes`. The existing `PUT` operation replaces the contents of OAuth client parameters with the values from the request.

Updates in April 2021 Documentation Refresh for 12c Release 2 (12.2.1.4.0)

This revision of Oracle® Fusion Middleware Administering Oracle Access Management contains feature updates and addresses bug fixes.

- **Session Management Optimizations**
Session Management Engine has been optimized and tuned to provide improved system performance under load.

Also see, Database Tuning for Oracle Access Management in the *Tuning Performance Guide*
- **OAuth Refresh Token Management**
OAuth Token Management capabilities have been enhanced with the ability to invalidate Refresh Tokens.

For details, see [Revoking OAuth Tokens](#)
- **12c WebGates for Apache and IIS Web Servers**
WebGates for IIS and Apache Web Servers are made available in this release.

For details, see [Installing and Configuring IIS 12c WebGate for OAM](#) in *Installing WebGates for Oracle Access Manager*
- **Support for TLS 1.3 & FIPS 140-2**
This release is compliant with the latest FIPS and TLS standards and versions.

For details, see [Enabling FIPS Mode on OAM Server](#) and [TLS 1.3 and TLS 1.2 Support in Oracle Access Management](#)

Updates in January 2021 Documentation Refresh for 12c Release 2 (12.2.1.4.0)

This revision of Oracle® Fusion Middleware Administering Oracle Access Management contains feature updates and addresses bug fixes.

- **Recommendation to Use Embedded Credential Collector (ECC) in OAM 12c**
It is recommended that you use ECC for the new features introduced in OAM 12c. Some of the new features introduced in OAM 12c do not support DCC. For example, OpenIDConnect with DCC is not supported.

For additional details, see Doc ID 2634863.1 at <https://support.oracle.com>.

Also see, [Overview of Access Manager Credential Collection](#) and [Embedded Credential Collector Versus Detached Credential Collector](#)
- **Proof Key for Code Exchange (PKCE) Support in OAM**
Introduces PKCE support in the existing OAM OAuth Authorization Code Grant Flow. It can be used to enhance the security of the existing 3-legged OAuth, mitigating possible authorization code interception attacks. You can enable PKCE at the domain level or just for a specific client.

 **Note:**

OAM validates `code_challenge` for Base64 with padding. To fix this and ensure RFC compliant behavior for the `code_challenge` (Base64 without padding as described in <https://tools.ietf.org/html/rfc7636#appendix-A>), you must download and apply the OAM Patch **32406872**. For details, see the note [How To Enable OAuth Proof Key For Code Exchange \(PKCE\) in Oracle Access Manager \(OAM\) 12.2.1.4 \(Doc ID 2755209.1\)](#) at <https://support.oracle.com>.

OAM will support only the RFC compliant behavior in all the subsequent releases.

For more information, see [Proof Key for Code Exchange \(PKCE\) Support in OAM](#).

- **Keep the OAUTH_TOKEN Response Unset**

OAM provides an option to not set the `OAUTH_TOKEN` cookie or header when SSO Session Linking is enabled. You must set the challenge parameter `IS_OAUTH_TOKEN_RESPONSE_SET` to `false`.

 **Note:**

If `IS_OAUTH_TOKEN_RESPONSE_SET` is not configured, or set to `true` then the `OAUTH_TOKEN` cookie/header is set.

Updates in October 2020 Documentation Refresh for 12c Release 2 (12.2.1.4.0)

This revision of Oracle® Fusion Middleware Administering Oracle Access Management contains feature updates and addresses bug fixes.

- **Support for AWS Role Mapping Attribute in SAML Response**

Introduces a new function that can be configured in SP Attribute Profile for supporting the AWS role mapping attribute in SAML response.

For details, see [AWS Role Mapping Attribute in SAML Response](#)

- **Support for Attribute Value Mapping and Filters in OAM Federation**

OAM federation supported Attribute Name Mapping. It extends the support for Attribute Value Mapping and Attribute Filtering features.

For details, see [Using Attribute Value Mapping and Filtering](#)

Updates in July 2020 Documentation Refresh for 12c Release 2 (12.2.1.4.0)

This revision of Oracle® Fusion Middleware Administering Oracle Access Management contains feature updates and addresses bug fixes.

- **Support for SameSite=None Attribute in OAM Cookies**
OAM adds SameSite=None attribute to all the cookies set by WebGate and OAM Server.
For details, see [Support for SameSite=None Attribute in OAM Cookies](#)
- **X.509 Authentication with Extended Key Usage (EKU)**
In X.509 authentication flows, Extended Key Usage (EKU) certification extension check can be added optionally to ensure that the usage of the certificate is allowed.
For details, see [X.509 Authentication Using Extended Key Usage \(EKU\)](#).

Updates in April 2020 Documentation Refresh for 12c Release 2 (12.2.1.4.0)

This revision of Oracle® Fusion Middleware Administering Oracle Access Management contains feature updates and addresses bug fixes.

- **OAuth Consent Management**
Provides capability for managing user consents, persisting user consents and providing mechanism to revoke them across DataCenters. Consent revocation capability is provided for both Administrators as well as individual users.
For details, see [Enabling Consent Management](#) and [Enabling Consent Management on MDC](#)
- **OAuth Just-In-Time (JIT) User Linking and Creation**
Provides capability to provision users automatically. The idToken as received from IDP has user attributes. These user attributes can have values like userId, user name, first name, last name, email address, and so on, which could be used for linking users to entries in the local id store or create them, if they do not exist.
For details, see [OAuth Just-In-Time \(JIT\) User Provisioning](#)
- **OAM Snapshot Tool**
Provides tooling to create a snapshot of the OAM Domain with all its configurations, persist it, and use it for creating fully functional OAM Domain clones.
For details, see [Using the OAM Snapshot Tool](#)

Features of Oracle Access Management 12c Release 2 (12.2.1.4.0)

Oracle Access Management 12c (12.2.1.4.0) includes the following features:

- **Passwordless Login**
Passwordless authentication allows you to bypass the standard web form based authentication when using a mobile device. For details, see [Using Passwordless Authentication with OAM](#).
- **Dynamic Client Registration**

Dynamic client Registration (DCR) provides a way for the native mobile apps (Android) to dynamically register as clients with the OAuth Server (OAM). For details, see [Dynamic Client Registration](#).

- **OAP over REST**

Oracle Access Protocol (OAP) over REST enables the use of HTTP(S) infrastructure to route and load balance requests. Changing the transport mechanism between WebGate and server has a beneficial impact on reducing operational cost for hybrid deployments where some components are On-Premise and others have moved to cloud. For details, see [Securing Communication between OAM Servers and WebGates using OAP over REST](#)

- **WebGate using PFS and Approved Cipher Suites for OAP Simple/Cert Mode Communication**

When the Simple/Cert Mode communication occurs, WebGate ensure that valid and approved cipher suites defined by the admin are used. For details, see [About WebGate Usage of PFS and Approved Cipher Suites for OAP Simple/Cert Mode Communication in Administering Oracle Access Management](#)

- **HealthCheck Framework**

HealthCheck Framework enables health check on servers. These checks can be performed using REST API or by scheduling periodic checks on the server. Each schedule can be associated with a specified set of tests to be run. For details, see [Monitoring Server Health with Health Check Framework](#)

- **Modified UserInfo Response**

The format of the UserInfo response for OAuth flows is modified with the following changes:

- Additional new parameters `guid` and `sub` are included in the response.
- The parameters `Profile`, `Email`, `Address`, and `Phone` are returned directly under the root tag instead of separate containers for each of the parameters.
- The parameters `email_verified` and `phone_number_verified` are returned as booleans.

For example,

```
{
  "guid": "6C9CF210194A11E99FB45DDD0C60B95A",
  "sub": "weblogic",
  "family_name": "weblogic",
  "preferred_username": "weblogic",
  "updated_at": "1548740667872",
  "email_verified": false,
  "phone_number_verified": false
}
```

To retrieve the user info attributes in the older format (see the following example), set the custom attribute `UserInfoScopeCont` to `true` at the domain level.

Sample UserInfo response format when the custom attribute `UserInfoScopeCont` is set,

```
{
  "profile": {
```

```

        "guid": "6C9CF210194A11E99FB45DDD0C60B95A",
        "sub": "weblogic",
        "family_name": "weblogic",
        "preferred_username": "weblogic",
        "updated_at": "1548743708100"
    },
    "email": {
        "email_verified": false
    },
    "address": {},
    "phone": {
        "phone_number_verified": false
    }
}

```

- **Policy Cache Resiliency**

Improved resilience of the managed servers with the ability to read, validate and replace policy cache in a small step within the server, and delegation of cache building to the Admin Server. Introduced distribution of policy cache from Admin to manage servers with write once and read many times and reducing contention between multiple OAM server's policy cache present in a cluster.

Policy cache can be fine-tuned using parameters. For details, see [Configuring Policy Cache Parameters](#).

Features Not Supported in Access Manager

This section provides a list of features that are not supported in Access Manager releases.

- [Features Not Supported in Access Manager 12.2.1.4.0](#)
- [Features Not Supported in Access Manager 12.2.1.3.0](#)

Features Not Supported in Access Manager 12.2.1.4.0

The unsupported features are the same as in 12.2.1.3.0 release.

Features Not Supported in Access Manager 12.2.1.3.0

The following table lists the features that will be unsupported from OAM 12.2.1.3.0 and provides the migration path:

Unsupported Features in OAM 12.2.1.3.0	Description	Migration Path
10g OSSO server co-existence	OAM 12c server does not support co-existence with the OSSO servers	Upgrade from OSSO to OAM 11g R2PS3 and then upgrade to OAM 12c.
OpenSSO server co-existence	OAM 12c server does not support co-existence with the OpenSSO server.	Upgrade to OAM 11gR2PS3 and then upgrade to OAM 12c.

Unsupported Features in OAM 12.2.1.3.0	Description	Migration Path
OAM 10g server co-existence	OAM 12c server does not support co-existence with OAM 10g server.	Migrate to OAM 12c server.
OpenSSO agents	OpenSSO agents are not supported in the OAM 12c release.	Migrate to supported 12c agents. OAM 11g and 12c WebGates and Accessgates are supported in OAM 12.2.1.3.0
mod_osso	OAM 12c does not support mod OSSO (OSSO Agent Proxy) agents.	Migrate to 12c WebGate agents and upgrade to OAM 12c.
OAM10g WebGate	OAM 12c server does not support OAM 10 WebGates.	Migrate to OAM11g R2PS3 or OAM 12c WebGates Upgrade the server to OAM 12c.
IDMConfigTool	OAM 12c does not support the following commands and attributes: <ul style="list-style-type: none"> • prepareIDStore= FUSION • prepareIDStore= OAM • configPolicyStore • configOVD • disableOVDAccessConfig • postProvConfig • validate: All options are not supported • ovdConfigUpgrade • upgradeOIMTo11gWebgate • POLICYSTORE_SHARES_IDSTORE • SPLIT_DOMAIN 	
IAMSuiteAgent	OAM 12c does not support IAMSuiteAgent. Till R2PS3, IAMSuiteAgent was the OOB agent protecting the OAM console. From 12c PS3 onwards, this is done using default OOB Login page. As per EDG (Enterprise Development Guide), it is recommended to protect OAM console using a webgate agent.	

Unsupported Features in OAM 12.2.1.3.0	Description	Migration Path
Oracle Mobile Security Suite (OMSS)	OAM 12c does not support OMSS.	It is recommended to use OpenID Connect. For details, see OIDC Client Integrations with Social Identity Providers .
Security Token Service (STS)	OAM 12c does not support STS.	It is recommended to use OAuth. For details, see Understanding OAuth Services .

 **Note:**

There is no 12c version of Oracle Adaptive Access Manager (OAAM), continue to use OAAM 11g with OAM 12c.

In 12c, for mobile and social login usecases, we recommend customers to use standard OAuth. We are deprecating proprietary way of achieving these use cases so that the customers can move to a more standards-based approach that would allow better interoperability. The following services are deprecated in 12c:

- Mobile and Social Services
- Mobile OAuth Service
- Security Token Service
- Access Portal Service

Part I

Introduction to Oracle Access Management

The Oracle Access Management, an enterprise-level security platform, includes Oracle Access Management Access Manager (Access Manager) and many incorporated services including (but not limited to) Identity Federation and Identity Context.

This section contains information on the available services as well as instructions on how to login and start using the Oracle Access Management Console. It contains the following chapters:

- [Introducing Oracle Access Management](#)
- [Getting Started with Oracle Access Management](#)

1

Introducing Oracle Access Management

Oracle Access Management provides an enterprise-level security platform, which comprises Oracle Access Manager and many incorporated services including (but not limited to) Identity Federation and Identity Context

The following topics provide a high-level overview of the Oracle Access Management architecture and services:

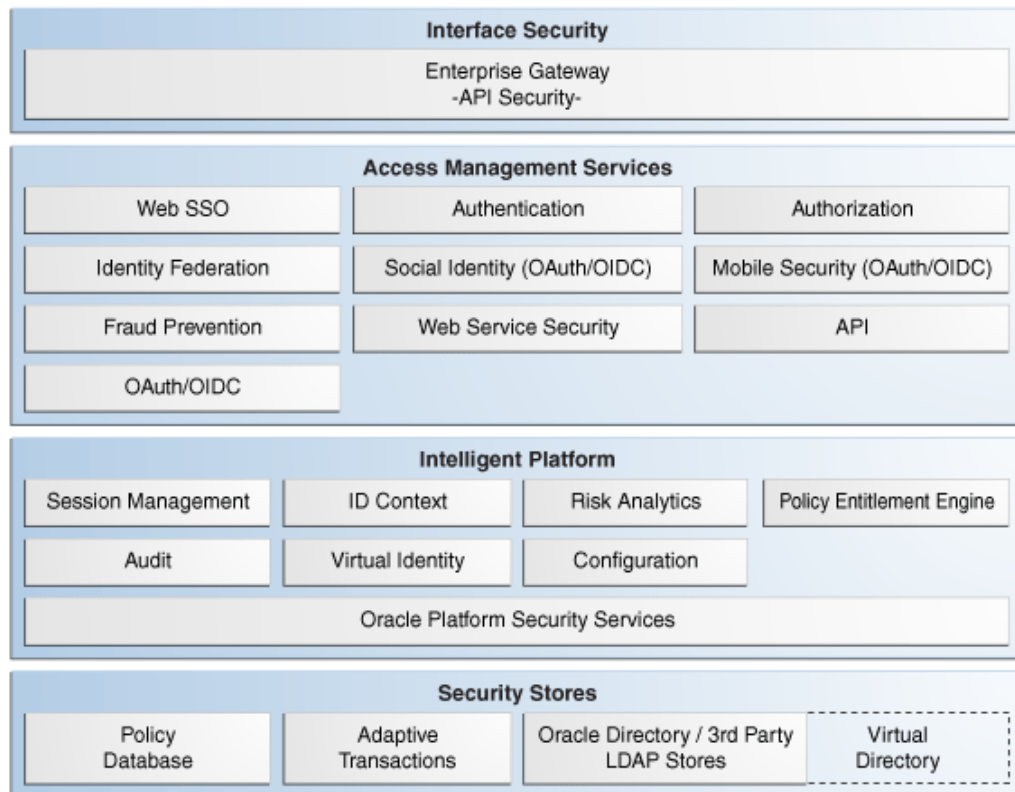
- [Understanding Oracle Access Management Services](#)
- [Understanding Oracle Access Management Access Manager](#)
- [System Requirements and Certification](#)
- [Understanding Oracle Access Management Installation](#)

1.1 Understanding Oracle Access Management Services

Oracle Access Management is a Java, Enterprise Edition (Java EE)-based enterprise-level security application that provides a full range of Web-perimeter security functions and Web single sign-on services including identity context, authentication and authorization; policy administration; testing; logging; auditing; and more.

It leverages shared platform services including session management, Identity Context, risk analytics, and auditing, and provides restricted access to confidential information. Many existing access technologies in the Oracle Identity Management stack converge in the Oracle Access Management stack as illustrated in [Figure 1-1](#).

Figure 1-1 Oracle Access Management Overview



Oracle Access Management includes these services.

- Oracle Access Management Access Manager (Access Manager) is described in "[Understanding Oracle Access Management Access Manager](#)" and the following parts of this guide.
 - [Managing Common and System Configurations](#)
 - [Logging, Auditing, Reporting and Monitoring Performance](#)
 - [Managing Access Manager Settings and Agents](#)
 - [Managing Access Manager SSO, Policies, and Testing](#)
 - [Integrating Access Manager with Other Products](#)
- Oracle Access Management Identity Federation (Identity Federation) provides cross-domain single sign-on support using open federation protocol standards such as SAML and OpenID. This Identity Federation service includes a streamlined user interface and administration experience. For more information, see the chapters listed in [Managing Oracle Access Management Identity Federation](#)
- The Adaptive Authentication Service is a One Time Password Authenticator that provides multifactor authentication in addition to the standard user name and password type authentication. It provides a framework for adding a custom second factor authentication processor that accepts a PIN from a user. For more information, see the chapters listed in [Managing the Adaptive Authentication Service and Oracle Mobile Authenticator](#)

- OAuth Services allows organizations to implement the open OAuth 2.0 Web authorization protocol in an Access Manager environment. OAuth Services enables a client to access resources protected by Access Manager that belong to another resource owner. An OAuth client can be an application or service created and controlled by your organization, or it can be an application or service created and controlled by another organization that requires access to resources protected by Access Manager. For more information, see the chapters listed in [Managing the Oracle Access Management OAuth Service and OpenIDConnect](#)
- Identity Context provides context-aware security policy management that enables Administrators to control the level of security imposed in an application delivery environment through security frameworks provided by Oracle Identity Management. For more information, see the chapters listed in [Using Identity Context](#).

1.2 Understanding Oracle Access Management Access Manager

Oracle Access Management Access Manager (Access Manager) is the former (standalone) product named Oracle Access Manager. Access Manager, it provides the Oracle Fusion Middleware single sign-on (SSO) solution. It operates independently or with the Access Manager Authentication Provider.

Access Manager SSO allows users and groups to access multiple applications after authentication, eliminating the need for multiple sign-on requests. To enable SSO, a Web server, Application Server, or any third-party application must be protected by a WebGate that is registered as an agent with Access Manager. Administrators then define authentication and authorization policies to protect the resource. To enforce these authentication policies, the agent acts as a filter for HTTP requests.

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 any Web applications currently using Oracle ADF Security and the OPSS SSO Framework with Access Manager. (See [Integrating Oracle ADF Applications with Access Manager SSO](#).) The following sections contain more details on Access Manager.

- [About Components in Access Manager](#)
- [Understanding Access Manager Deployments](#)

See Also:

Authentication Basics in *Securing Applications with Oracle Platform Security Services*

1.2.1 About Components in Access Manager

Access Manager sits on an instance of Oracle WebLogic Server and is part of the Oracle Fusion Middleware Access Management architecture.

Figure 1-2 illustrates the primary Access Manager components and services. The Protocol Compatibility Framework interfaces with OAM WebGates, and custom Access Clients created using the Access Manager Software Developer Kit (SDK).



Note:

This section does not illustrate or discuss all Access Manager components.

Figure 1-2 Access Manager Components and Services

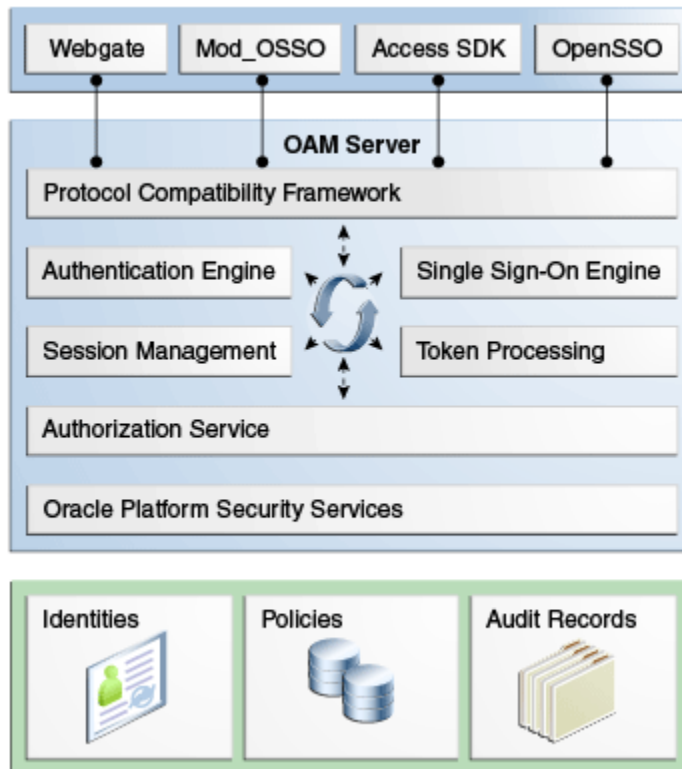
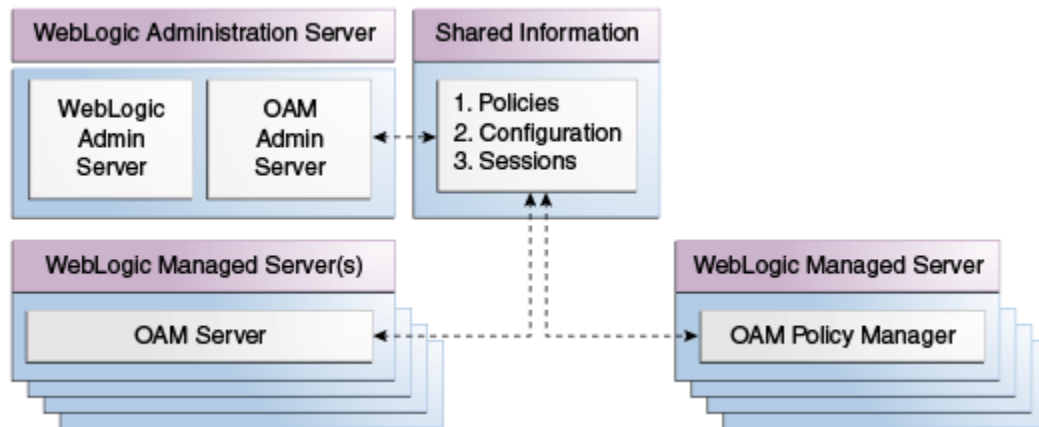


Figure 1-3 illustrates the distribution of Access Manager components.

Figure 1-3 Access Manager Component Distribution



The Oracle Access Management Console resides on the Oracle WebLogic Administration Server (referred to as AdminServer). WebLogic Managed Servers hosting OAM runtime instances are known as OAM Servers. Information shared between the two includes:

- Agent and server configuration data
- Access Manager policies
- Session data (shared among all OAM Servers)

Policy Manager Console can optionally be deployed on the WebLogic Managed Servers. See [Oracle Access Management Console and the Policy Manager Console](#) for details.

1.2.2 Understanding Access Manager Deployments

Your enterprise may have more than one Oracle Access Manager deployments. Irrespective of the deployment size, the configuration wizard installs various components in a newly created WebLogic Server domain.

[Table 1-1](#) describes the types of deployments in which Access Manager might be installed by your enterprise.

Table 1-1 Access Manager Deployment Types

Deployment Type	Description
Development Deployment	Ideally a <i>sandbox</i> -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 of Access Manager in your deployment, you create a new WebLogic Server domain (or extend an existing domain). Regardless of the deployment size or type, in a new WebLogic Server domain, the following components are installed using the Oracle Fusion Middleware Configuration Wizard.

- WebLogic Administration Server

 **Note:**

In an existing WebLogic Server domain, the WebLogic Administration Server is already installed and operational.

- Oracle Access Management Console deployed on the WebLogic Administration Server
- A WebLogic Managed Server for Oracle Access Management services
- Application deployed on the Managed Server

 **See Also:**

Understanding Oracle WebLogic Server Domains in *Understanding Domain Configuration for Oracle WebLogic Server*

Once the domain is configured, additional details are defined for OAM Servers, Database Schemas, (optional) WebLogic Managed Servers and clusters, and the following store types:

- **Policy Store:** The default policy store is file-based for development and demonstration purposes, and is not supported in production environments. All policy operations and configurations are performed directly on the database configured as the policy store in production environments.

 **See Also:**

["Managing the Policy and Session Database "](#)

- **Identity Store:** The default Embedded LDAP data store is set as the primary user identity store for Access Manager.

 **See Also:**

["Registering and Managing User Identity Stores "](#)

- **Keystore:** A Java keystore is configured 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.

 **See Also:**

["Managing the Policy and Session Database "](#)

1.3 System Requirements and Certification

Ensure that your environment meets the system requirements such as hardware and software, minimum disk space, memory, required system libraries, packages, or patches before performing any installation.

Refer to the system requirements and certification documentation on Oracle Technology Network (OTN) for information about hardware and software requirements, platforms, databases, and other information.

The system requirements document covers information such as hardware and software requirements, minimum disk space and memory requirements, and required system libraries, packages, or patches:

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html>

The certification document covers supported installation types, platforms, operating systems, databases, JDKs, and third-party products:

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

1.4 Understanding Oracle Access Management Installation

Using the Oracle Fusion Middleware Configuration Wizard deploy components for a new domain and perform post-installation tasks.

The following sections contain information and links regarding Access Manager installation and post-installation tasks.

- [About Oracle Access Management Installation](#)
- [About Oracle Access Management Post-Installation Tasks](#)

1.4.1 About Oracle Access Management Installation

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.

You can access the *Oracle Fusion Middleware Supported System Configurations* document by searching the Oracle Technology Network (OTN) Web site using the document name, or click the link below.

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

Using the Oracle Fusion Middleware Configuration Wizard, the following components are deployed for a new domain:

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

- An application deployed on the Managed Server

See About the Oracle Identity and Access Management Installation in *Installing and Configuring Oracle Identity and Access Management* for details on installation.

1.4.2 About Oracle Access Management Post-Installation Tasks

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.

During initial deployment, the WebLogic Administrator userID and password are set for use when signing in to both the Oracle Access Management and WebLogic Server Administration Console. A different Administrator can be assigned for Oracle Access Management, as described in "[About Oracle Access Management Administrators](#)". Administrators can log in and use the Oracle Access Management Console for the post-installation tasks documented in [Table 1-2](#).

Table 1-2 Oracle Access Management Post-Installation Tasks

Service	Requirements
Access Manager	Enable Access Manager Service Register: <ul style="list-style-type: none"> • Data sources • OAM server instances • Agents for Access Manager • Application domains and policies that protect resources Configure: <ul style="list-style-type: none"> • Common settings, including session-timing • Certificate validation • Common password policy Configure Access Manager settings
Identity Federation	<ul style="list-style-type: none"> • Enable Identity Federation Service • Configure federation settings • Register identity provider and service provider partners

2

Getting Started with Oracle Access Management

Start your servers and log into the Oracle Access Management Console, before you start working with Oracle Access Management.

Oracle Access Management must have already been deployed. See *Installing the Oracle Identity and Access Management Software* in *Installing and Configuring Oracle Identity and Access Management*.

The following topics describe how to start and stop servers when you work with Oracle Access Management:

- [Starting and Stopping Servers in Your Deployment](#)
- [About Oracle Access Management Administrators](#)
- [Oracle Access Management Console and the Policy Manager Console](#)
- [Understanding the Oracle Access Management Console](#)
- [About Logging Into the Oracle Access Management Console](#)
- [Using the Oracle Access Management Console](#)
- [Command-Line Tools for Configuration](#)
- [Logging, Auditing, Reporting, and Monitoring Performance](#)
- [Configuring Oracle Access Management Login Options](#)

2.1 Starting and Stopping Servers in Your Deployment

The Oracle Access Management Console is deployed on the WebLogic Administration Server (AdminServer). Therefore, Oracle Access Management Administrators can access it only when the AdminServer is running. If the Oracle Access Management Console is protected by a WebGate, the OAM Server must also be running and the node Manager must be started before the other servers.

The following sections have more details:

- [Starting Node Manager](#)
- [Removing OAM Server from WLS 12c defaultCoherenceCluster](#)
- [Starting and Stopping WebLogic AdminServer](#)
- [Starting and Stopping Managed WebLogic Servers and Access Manager Servers](#)

2.1.1 Starting Node Manager

You can use the Node Manager, a Java utility, to perform common operations tasks for a Managed Server, regardless of its location with respect to its Administration Server. Node

Manager must run before you can start and stop the WebLogic AdminServer, or WebLogic managed servers hosting OAM Servers.

After you complete the configuration, you need to make sure that the Node Manager runs the `startNodeManager.sh` script. Oracle WebLogic Administration Server does not do this automatically.

```
$WLS_HOME/server/bin/startNodeManager.sh
```

See the *Oracle WebLogic Server Administrator Guide*.

To start Node Manager:

1. Change to your `$WLS_HOME/server/bin` directory.
2. Enable Start Scripts: Run `setNMProps` to start the stack and instruct Node Manager to enable the use of start scripts (`StartScriptEnabled=true`).

```
./setNMProps.sh
```

3. Start Node Manager.

```
./startNodeManager.sh
```

2.1.2 Removing OAM Server from WLS 12c defaultCoherenceCluster

Exclude all OAM clusters (including policy manager and OAM runtime server) from the default WebLogic Server 12c coherence cluster using the WebLogic Server Administration Console.

In OAM 12.2.1.3.0, server-side session management uses database and does not require coherence cluster to be established. In some environments, warnings and errors are observed due to default coherence cluster initialized by WebLogic. To avoid or fix these errors, exclude all OAM clusters from default WLS coherence cluster using the following procedure.

1. Login to the WebLogic Server Administration Console.
2. In the left pane of the console, expand **Environment** and select **Coherence Clusters**.
The Summary of Coherence Clusters page displays the Coherence cluster configurations that have been created in this domain.
3. Click **defaultCoherenceCluster** and select the **Members** tab.
4. From **Servers** and **Clusters**, deselect all OAM clusters (including policy manager and OAM runtime server).
5. Click **Save**.

2.1.3 Starting and Stopping WebLogic AdminServer

Starting the WebLogic AdminServer the first time can take 12-15 minutes or more. This process must not be interrupted or terminated as policy data might be corrupted.

The following procedure describes starting and stopping the WebLogic AdminServer using the scripts located in your `$DOMAIN_HOME/bin` directory.

- **Unix:** `startWebLogic.sh` or `stopWebLogic.sh`
- **Windows:** `startWebLogic.cmd` or `stopWebLogic.cmd`

⚠ WARNING:

If startWebLogic.cmd (Windows) or startWebLogic.sh (Linux) is stopped for any reason (whether accidentally or because of a system crash or reboot), policy data might be corrupted. This would require removal and recreation of the domain and running the RCU again to recreate the OAM schema.

To start and stop WebLogic AdminServer:

1. Navigate to your `$DOMAIN_HOME/bin`.
2. Start AdminServer:
 - **Unix:** `./startWebLogic.sh`
 - **Windows:** `run startWebLogic.cmd`
3. Stop AdminServer:
 - **Unix:** `./stopWebLogic.sh`
 - **Windows:** `run stopWebLogic.cmd`

2.1.4 Starting and Stopping Managed WebLogic Servers and Access Manager Servers

You can perform all start and stop operations for managed WebLogic Servers hosting Oracle Access Management Servers (OAM Servers) from either a command prompt, the Oracle WebLogic Server Administration Console, the OAM Policy Manager Console, or the Oracle Enterprise Manager Fusion Middleware Control.

When using the command line scripts (located in the `$DOMAIN_HOME/bin` directory), the Managed Server name and the AdminServer URL are required as input.

The Unix system scripts are `startManagedWebLogic.sh` and `stopManagedWebLogic.sh`, and the Windows system scripts are `startManagedWebLogic.cmd` and `stopManagedWebLogic.cmd`.

To start and stop Managed WebLogic servers and Access Manager servers:

1. Navigate to `$DOMAIN_HOME/bin`.
2. Start OAM Server:
 - **Unix:** `./startManagedWebLogic.sh MANAGED_SERVER_NAME ADMIN_SERVER_URL`
 - **Windows:** `run startManagedWebLogic.cmd MANAGED_SERVER_NAME ADMIN_SERVER_URL`

If the managed server is named `oam_server1` and the AdminServer URL is `http://examplewlsadminhost.example.com:7001`, the start command run on a Unix system would be:

```
startManagedWebLogic.sh oam_server1 http://examplewlsadminhost.example.com:7001
```

3. Stop OAM Server:
 - **Unix:** `./stopManagedWebLogic.sh MANAGED_SERVER_NAME ADMIN_SERVER_URL`
 - **Windows:** `run stopManagedWebLogic.cmd MANAGED_SERVER_NAME ADMIN_SERVER_URL`

If the managed server is named `oam_server1` and the AdminServer URL is `http://examplewlsadminhost.example.com:7001`, the stop command run on a Unix system would be:

```
stopManagedWebLogic.sh oam_server1 http://  
examplewlsadminhost.example.com:7001
```

2.2 About Oracle Access Management Administrators

A single default LDAP group, the WebLogic Server `Administrators` group, is set in the Default User Identity Store (Embedded LDAP) designated as the System Store. The LDAP group, when assigned to a specified user, grants full system and policy configuration privileges.

Specifying a different LDAP group prohibits WebLogic Administrators from logging in to Oracle Access Management Console or from using administrative command-line tools.



Note:

Unless explicitly stated, the term Administrator in this guide refers to the Oracle Access Management System Administrator.

During initial deployment with the Oracle Fusion Middleware Configuration Wizard, the System Administrator userID and password are set. These credentials grant access to the:

- Oracle Access Management Console to register and manage system configurations, security elements, and policies.
See [Oracle Access Management Console and the Policy Manager Console](#) for details.
- 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. See the *Administering Oracle Fusion Middleware* for more information.
- Custom Administrative command-line tools (including the WebLogic Scripting Tool and Remote Registration Tool) provide an alternative to the Oracle Access Management Console for a specific set of functions.

See [Command-Line Tools for Configuration](#).

Initially, a System Administrator user must log in to the Oracle Access Management 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 Management administration and a different set for WebLogic administration.

See [Understanding Administrator Roles](#).

2.3 Oracle Access Management Console and the Policy Manager Console

Oracle Access Management allows for two console interfaces: **Oracle Access Management Console**, the full-featured graphical interface deployed on the WebLogic AdminServer and **Policy Manager Console**, the interface that can be deployed on one or more WebLogic Managed Servers.

- **Oracle Access Management Console:** The Oracle Access Management Console can be accessed at:

`http://wlsadminhost.example.com:7001/oamconsole`

See [Understanding the Oracle Access Management Console](#) .

- **Policy Manager Console:** It does not contain the full functionality available in the Oracle Access Management Console deployed on the AdminServer. The Policy Manager Console has only the policy administration functionality of the familiar Oracle Access Management Console. It is deployed when more capacity is needed to support many delegated administrative users of Access Manager policies.

`http://wlsadminhost.example.com:14150/access`



Note:

REST endpoints, WLST and the RREG servlet are available only on AdminServer.

2.4 Understanding the Oracle Access Management Console

The Oracle Access Management Console is a Web-based program that provides function controls for system and policy configuration.

This console displays a Launch Pad and subsequent pages based on the Administration Role to which a user is assigned a successful login. It is divided into Launch Pads and page-level tabs with forms and controls.



Note:

Admin operations are not supported on OAM console when DB is down. See [Table 16-2](#)

Any clicked shortcut appears as a named tab next to the Launch Pad. Each page is displayed only once. No warning is issued if you attempt to open the same page multiple times. The tab of the active page is white. Only the active page is visible and generally provides a work space where you can add, view, or modify related settings. Up to ten pages (tabs) can be open simultaneously. You can see named tabs for each page and click the tab to access a page that is concealed. See the following sections for details on the new Launch Pads.

- [System Launch Pad](#)

- [Access Manager Launch Pad](#)
- [Agents Launch Pad](#)
- [Help Desk Launch Pad](#)



Note:

The Oracle Access Management Console is designed for optimal display at a resolution of 1024x768.

2.4.1 System Launch Pad

The System Launch Pad will display when the user name of the Oracle Access Management System Administrator is entered.

See [About Oracle Access Management Administrators](#).

This role has access to all functions and features of the Console including policy creation, system configuration, and services settings (including Access Manager, Security Token Service, Identity Federation, Access Portal, and so on).

When the System Administrator is logged in, access is granted to five Launch Pads:

1. **Application Security** contains the functions generally associated with Oracle Access Manager and single sign-on (SSO). From this Launch Pad, click the appropriate link to gain access to agent registration, policy and policy objects creation, session management, password policy, authentication modules, and plug-ins.
2. **Federation** contains functions associated with Identity Federation (including links to configure and manage Identity, and Service Providers), the Security Token Service, Social Identity, OAuth Services and the Access Portal Service.



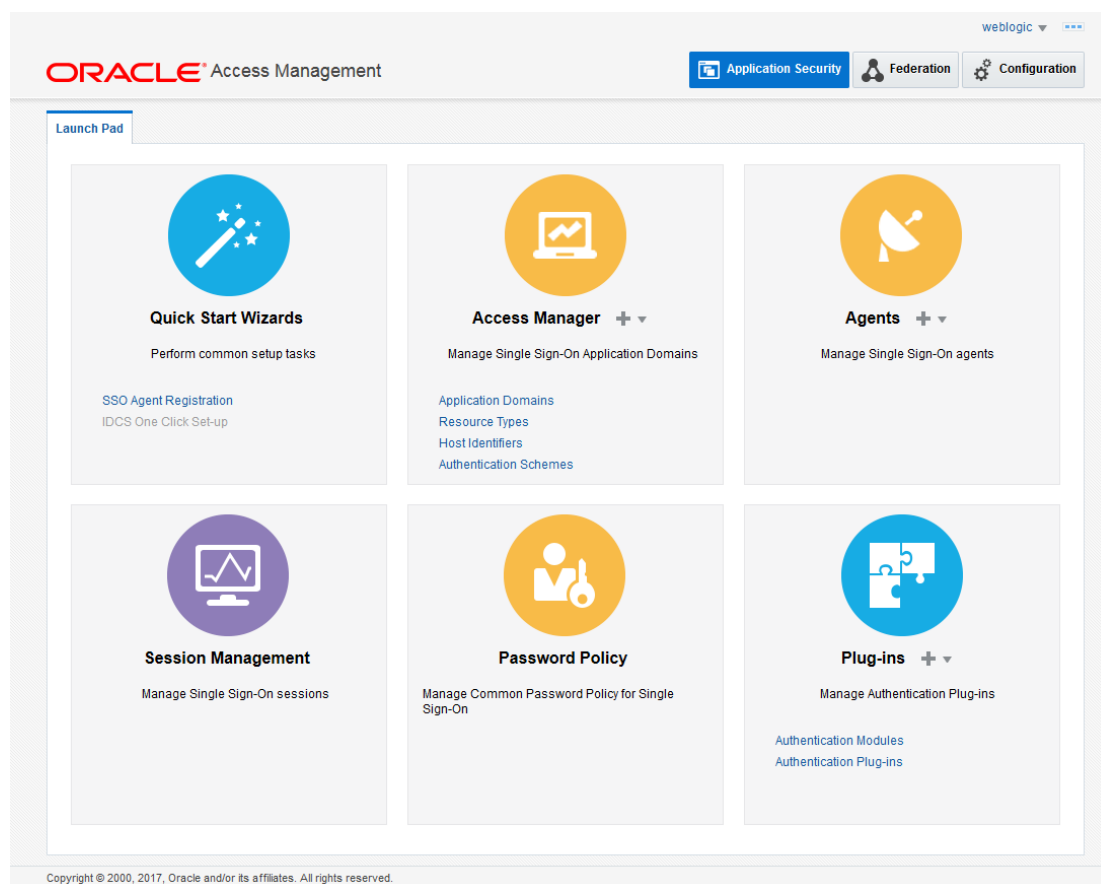
Note:

Some of these services are disabled by default and would need to be enabled under the Configuration Launch Pad.

3. **Configuration** contains panels for managing the Oracle Access Management system settings. This includes enabling and disabling available Access services, configuring user identity stores and settings, certificate validation, server instances, and granting administrative permissions.

[Figure 2-1](#) shows the Oracle Access Management System Administrator Console with the Application Security Launch Pad displayed. This is the default login view. Note the four disabled tabs on the top right of the screenshot which, when clicked, will display the other Launch Pads visible by the System Administrator.

Figure 2-1 Oracle Access Management Administrator Launch Pad



2.4.2 Access Manager Launch Pad

The Oracle Access Manager Launch Pad and subsequent functionality are displayed when the user name entered is assigned to the Application Administrator (appadminuser) Role.

See [Understanding Administrator Roles](#).

This role has access to all functions and features of the Console that includes policy object creation and policy management. When the Application Administrator is logged in, access to the Launch Pads is limited to Access Manager and Automated Policy Synchronization (APS).

2.4.3 Agents Launch Pad

The Agents Launch Pad and the subsequent functionality are displayed when the user name entered is assigned to the Oracle Access Management Agent Administrator Role.

See [Understanding Administrator Roles](#).

This role has access to all functions and features of the Console that include management and configuration of SSO Agents.

2.4.4 Help Desk Launch Pad

The Help Desk Launch Pad and the subsequent functionality are displayed when the user name entered is assigned to the Oracle Access Management Help Desk Administrator Role.

See [Understanding Administrator Roles](#).

Users with this role lands on the `http://wlsadminhost.example.com:7001/oamconsole/faces/helpdesk.jspx` page after logging in. The System Administrator can access this console directly by entering the URL in the browser. Any one without the Help Desk Administrator role cannot access this page.

2.5 About Logging Into the Oracle Access Management Console

When accessing the Oracle Access Management Console, the WebLogic Server (AdminServer) host and port must be specified in the URL.

Let's assume the following sample URL, `https://wlsadminhost.example.com:7001/oamconsole`. In this URL, the following is true.

- 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
- `wlsadminhost.example.com` refers to fully-qualified domain name of the computer hosting the Oracle Access Management Console (AdminServer)
- `7001` refers to the designated bind port for the Oracle Access Management Console, which is the same as the bind port used for AdminServer (the WebLogic Server Administration Console)
- `/oamconsole/` refers to the Oracle Access Management Console Log In page

Note:

If you specify an OAM Server host and port (as you would to access the ODSM console), the AdminServer redirects to the managed server which produces a '404 Not Found' error.

When navigating to the `/oamconsole` URL, the default Oracle Access Management Console login page is displayed. The following sections have details on logging into the Oracle Access Management Console.

- [Logging Into The Oracle Access Management Console](#)
- [Logging Into the Secure Oracle Access Management Console \(HTTPS\)](#)

 **Note:**

Ensure that you use the correct administrative credential to log in. Initially, the LDAP group for the Oracle Access Management Console 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.

2.5.1 Logging Into The Oracle Access Management Console

With appropriate administrative credentials, you can log into the Oracle Access Management Console.

Use this procedure to log in to the Oracle Access Management Console.

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

```
https://hostname:admin_server_port/oamconsole/
```

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

Username: *Admin_login_id*

Password: *Admin_password*

Language: English

See [Choosing a User Login Language](#).

3. Click the Login button.
 - **Successful:** The Oracle Access Management Console Welcome page is displayed.
 - **Not Successful:**
See [Administrator Lockout](#).

 **See Also:**

[About Oracle Access Management Administrators](#).

2.5.2 Logging Into the Secure Oracle Access Management Console (HTTPS)

After enabling SSL on the Adminserver and OAM Managed Server, or after configuring administration port (HTTPS), you can add the CA cert to the libOVD keystore. This allows logging in without connection issues.

To go into the Secure Oracle Access Management Console (HTTPS):

1. Change to the directory that contains the Demoidentity.jks.

```
$ cd $MIDDLEWARE_HOME/wlserver_10.3/server/lib/
```

2. Export the CA certificate from the Weblogic keystore using the following commands.

The `-list` command prints the contents of the keystore for reference. `DemoidentityKeyStorePassPhrase` is the default password for the keystore `Demoidentity.jks`.

```
$ keytool -list -keystore DemoIdentity.jks
-storepass DemoIdentityKeyStorePassPhrase

$ keytool -exportcert -keystore DemoIdentity.jks
-storepass DemoIdentityKeyStorePassPhrase -alias demoidentity
-file ~/demoidentity.cer
```

3. Import the Weblogic CA certificate to the libOVD keystore.

```
cd $DOMAIN_HOME/config/fmwconfig/ovd/default

mkdir keystores

cd keystores

$ keytool -importcert -keystore adapters.jks -storepass New_Password
-alias demoidentity -file ~/demoidentity.cer
```

4. Print the contents of the keystore to verify the import.

```
$ keytool -list -keystore ./adapters.jks -storepass New_Password
```

5. Add the password for the imported keystore to `trustStorePassword` in the `server.os_xml` file.

```
vim server.os_xml
server.os_xml: <keystore>keystores/adapters.jks</keystore>
server.os_xml: <trustStore>keystores/adapters.jks</trustStore>

<trustStore>keystores/adapters.jks</trustStore>
<trustStorePassword>New_Password</trustStorePassword>
```

6. Change the value of `ADMIN_URL` in `startManagedServer.sh` to point to the SSL port of the Weblogic server.
7. Restart both Adminserver and OAM Managed Server.
8. Log in as documented in [Logging Into The Oracle Access Management Console](#).

2.6 Using the Oracle Access Management Console

Log on to the Oracle Access Management console to perform the common console functionality like accessing SSO Agent search page to search specific elements, accessing online help, and logging out of Oracle Access Management console.

The following topics describe common console functionality:

- [Logging Out of the Oracle Access Management Console](#)
- [Accessing Online Help in the Oracle Access Management Console](#)
- [SSO Agent Search Page](#)

2.6.1 Logging Out of the Oracle Access Management Console

The Sign Out link appears in the upper-right corner of the Oracle Access Management Console. Click the Sign Out link to conclude your session. Oracle recommends that you also close the browser window after signing out.

To sign out of the Oracle Access Management Console:

1. Expand the drop down list under the name of the user that is logged in and select Sign Out.
2. Close the browser window.

2.6.2 Accessing Online Help in the Oracle Access Management Console

At any time while using the Oracle Access Management Console, you can click the Help link located in the drop down menu under the user name at the top of the Launch Pad page to get more information. Online Help topics link to information in an online version of this book. Generally, topics that are displayed by selecting Help in the Oracle Access Management Console appear in only English and Japanese languages. Online Help is not translated into the 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 access online help in the Oracle Access Management Console:

1. From the Oracle Access Management Console, click a tab.
2. Click Help in the drop down menu under the user name in the upper-right corner.
3. Review the page that appears in a new window and select one of the following links:
 - **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**—Expand Contents in the left Help pane to see all the help topics and the topics in the online manual.
 - **Search**—Click this link to enter your search criteria in the help search window.
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**—Sends the page through email.

2.6.3 SSO Agent Search Page

The Oracle Access Management Console provides search controls for specific elements such as Agents, Application Domains, and Resources.

Figure 2-2 is a screen shot of a Search page used for SSO Agent searches.

Figure 2-2 SSO Agent Search Page

The screenshot shows the Oracle Access Management console interface. At the top, there are navigation tabs for 'Application Security', 'Federation', 'Mobile Security', and 'Configuration'. Below this, the 'Search SSO Agents' page is displayed. It features a search form with the following fields: 'Name', 'Preferred Host', 'State' (a dropdown menu currently set to 'Enabled'), 'Primary Server', and 'Secondary Server'. There is a 'Create Webgate' button and 'Search' and 'Reset' buttons. Below the search form is a 'Search Results' section with a table that currently displays 'No data to display'. The table has columns for 'Row', 'Name', 'Preferred Host', 'State', 'Primary Server', and 'Secondary Server'. Above the table are action buttons: 'Create', 'Duplicate', 'Edit', 'Delete', and 'Detach'. At the bottom of the page, there is a copyright notice: 'Copyright © 2000, 2017, Oracle and/or its affiliates. All rights reserved.'

Search pages differ depending on the entity you are trying to find. In all searches, you can leave a field blank to display everything or use a wildcard (*) character if you do not know the exact name you seek. Some search controls include the ability to save your search criteria. From the search results table, you can choose an item to open for viewing or editing.

**Note:**

The search tool is case insensitive.

2.7 Command-Line Tools for Configuration

Several command-line tools are available to perform various tasks using the keyboard rather than the Oracle Access Management Console.

After using these commands, the configurations will be available in the console.

- Remote registration tool, `oamreg`, enables remote registration of Agents, and creation of default Application Domains.

**See Also:**

[OAM Agent Registration Parameters in the Console.](#)

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

 **See Also:**

Customization Commands in the *WLS Command Reference for WebLogic Server*.

2.8 Logging, Auditing, Reporting, and Monitoring Performance

Logging is the mechanism by which components and services write messages to a log file to capture critical component events, process, and state information. Auditing refers to the process of collecting for review specific information related to administrative, authentication, and run-time events. Access Manager provides a restricted-use license for Oracle BI Publisher and easy-to-use reporting packages. Monitoring performance refers to observing (viewing) performance metrics to make yourself aware of the state specific components.

Logging is the mechanism by which components write messages to a file. These messages can be logged at different levels of granularity. Oracle Access Management components use the same logging infrastructure and guidelines as any other component in Oracle Fusion Middleware 12c. Administrators can monitor performance and log messages for Access Manager using Oracle Fusion Middleware Control.

In Oracle Fusion Middleware, auditing provides a measure of accountability and answers to the “who has done what and when” types of questions. Oracle Access Management uses the Oracle Fusion Middleware Common Audit Framework to support auditing for a large number of user authentication and authorization run-time events, and administrative events (changes to the system). The Oracle Fusion Middleware Common Audit Framework provides uniform logging and exception handling and diagnostics for all audit events.

 **See Also:**

[Logging, Auditing, Reporting and Monitoring Performance](#).

Tuning Performance

2.9 Configuring Oracle Access Management Login Options

Oracle Access Management allows you to configure user login options like choosing a user login language, configuring forgot password URL, and many more.

The following topics describe how to configure Oracle Access Management user login options:

- [Administering the Forgot Password URL](#)
- [Choosing a User Login Language](#)
- [Understanding Persistent Login](#)

2.9.1 Administering the Forgot Password URL

When a user clicks the Forgot Password link on the Oracle Access Management login page, the user is taken to an Oracle Access Management Forgot Password page where a new password can be set in the case of a forgotten one.

The following sections contain procedures for administering the Forgot Password URL.

- [Setting a Forgot Password URL](#)
- [Retrieving a Forgot Password URL](#)

2.9.1.1 Setting a Forgot Password URL

You can set a new Forgot Password URL.

Run the following command:

```
curl --user weblogic:password
-w "%{http_code}"
-i -H
"Content-Type:application/json"
-H "Accept: */*"
-X PUT -d
'{"forgotPasswordURL":"http://oam-host:7777/identity/faces/
forgotpassword"}'
```

```
http://host:7001/oam/admin/api/v1/configurationService/forgotPassword
```

If successful, the "Forgot Password URL configured successfully" message is displayed in the output. If there is already a URL set for Forgot Password, running the command overwrites the previous Forgot Password URL.

2.9.1.2 Retrieving a Forgot Password URL

You can retrieve the Forgot Password URL.

Run the following command:

```
curl --user weblogic:password
-w "%{http_code}" \
-i \
```

```
http://host:7001/oam/admin/api/v1/configurationService/forgotPassword
```

2.9.2 Choosing a User Login Language

Topics relevant to user language selection in OAM include:

- [User Login Language Code](#)
- [Selecting A Language for Oracle Access Management Login](#)
- [Language Preference Cookie](#)
- [Propagating Language Preference and Application Integration](#)

2.9.2.1 User Login Language Code

Oracle Access Management supports language selection through a drop down list of languages on the login form combined with use of the OAM_LANG_PREF language preference cookie.

[Table 2-1](#) lists the supported languages and applicable language codes.

The **Language** column refers to languages supported by the Login Pages and the **Administrators** column refers to languages supported by the Oracle Access Management Console. If the language is supported by the Login Page, simply change the browser's language and users should see a translated page.

Table 2-1 Language Codes For Login Pages

Language Code	Language	Administrators
ar	Arabic	
cs	Czech	
da	Danish	
de	German	German
el	Greek	
en	English	English
es	Spanish	Spanish
fi	Finnish	
fr	French	French
fr-CA	Canadian French	
he	Hebrew	
hr	Croatian	
hu	Hungarian	
it	Italian	Italian
ja	Japanese	Japanese
ko	Korean	Korean
nl	Dutch	
no	Norwegian	
pl	Polish	
pt-BR	Brazilian Portuguese	Brazilian Portuguese
pt	Portuguese	
ro	Romanian	
ru	Russian	
sk	Slovak	
sv	Swedish	
th	Thai	
tr	Turkish	
zh-CN	Simplified Chinese	Simplified Chinese

Table 2-1 (Cont.) Language Codes For Login Pages

Language Code	Language	Administrators
zh-TW	Traditional Chinese	Traditional Chinese

To accomplish a very specific login experience, implement a custom login page using the customization facilities in Oracle Access Management as described in *Developing Applications with Oracle Access Management*.

2.9.2.2 Selecting A Language for Oracle Access Management Login

Oracle Access Management provides the language selection methods.

[Table 2-2](#) The order of these items in the table illustrate the preference order.

You can use the `configOAMLoginPagePref` WebLogic Scripting Tool (WLST) command to configure the login page language preferences.

See the WebLogic Scripting Tool Command Reference for Identity and Access Management for information regarding this WLST command.

Table 2-2 Oracle Access Management Language Selection Methods

Method	Description
Server Override	Allows the OAM Server to determine the language. It is intended to support scenarios where the User Agent cannot reliably indicate its language preference(s) or where the administrator needs to override other selection mechanisms for operational reasons.
Preference Cookie	A domain cookie (similar to <code>ORA_FUSION_PREFS</code>) that contains the user's language preferences. It is intended to allow lang preferences maintained by an application(s) personalization facilities to be used. Note: Multiple DNS domain support for the Preference Cookie is a limitation today. The solution will include Resource Webgates using the OAM Front-Channel protocol in combination with local resource cookie enhancements to manage preference cookie semantics across DNS domains. See Also: " Language Preference Cookie "
Browser Language	Allows User Agents (Browsers, REST Clients, HTTP Clients) to specify the user's language preference via an HTTP Accept-Language header.
Default Language	Used if Oracle Access Management cannot determine the user's language preference based on the specified selection mechanisms.

Language preferences are disabled until explicitly enabled. By default, the login form does not include the list of language values until the application locales are specified.



Note:

Language Selection is only available in the ECC login page; it is not currently available in the DCC login page.

2.9.2.3 Language Preference Cookie

The language preference cookie, OAM_LANG_PREF is a domain scoped cookies described in [Table 2-3](#).

Table 2-3 OAM_LANG_PREF Cookie

Parameters	Description
Name	OAM_LANG_PREF
Domain	Domain-scoped cookie
Path	/
Value	[<i>Cookie version</i>] [<i>separator</i>] [<i>UTF-8 BASE64(name-value pairs)</i>] For example: v1.0~kqhkiG9w0BAQQFADCB0TELM
ExpirationTime	Persistent Session (default) – Specified in OAM configuration
Secure Flag	Yes
preferredLanguage	BCP47/RFC4647. Specifically, the value space should conform to what is formally called the "language priority list".
defaultLanguageMarker	true (reconcile cookie with application maintained preferences) false (read from cookie).
Cookie Lifecycle	Oracle Access Management and other applications can perform create, read, update, and delete operations.

2.9.2.4 Propagating Language Preference and Application Integration

Oracle Access Management will propagate the language selected by the user to applications.

For more details, see [Table 2-4](#).

Table 2-4 Application Integration for Language Preference

Method	Description
HTTP Accept-Language Header	This enables application to integration without code change. This is a major advantage over the other options. We can expect this to be good for most applications that respond to the browser locale setting. This is the standard practice in internationalizing a Web application. We expect this to be able to become the standard option for all ADF based products, as well as any application that responds to browser locale. Note: OAM Agents ensure that the Accept-Language reflects the language selected. Also, ServletFilters could be used to make this happen.

Table 2-4 (Cont.) Application Integration for Language Preference

Method	Description
Access Manager Policy Response	Access Manager stores the language selection in the attribute langPref in the session namespace. For instance: <code>\$session.langPref</code> . This attribute can be passed to downstream applications using an HTTP Header and/or Cookie through the Access Manager Policy Response. The name of the Header and/or Cookie is a deployment time assignment.
Preference Cookie	When the language selected during login differs from the value stored in the Preference Cookie, Oracle Access Management will update the "preferredLanguage" parameter in the Preference Cookie with the newly selected language and set the defaultLanguageMarker parameter to "false".
IdentityContext	The language preference can be propagated as a custom claim in the IdentityContext. Select "oracle:idm:claims:session:attributes" as the claim name and then specify the session attribute using the following notation: <code>"preferredLanguage=\$session.langPref"</code> . The claim will be created with the name of "oracle:idm:claims:session:attributes:preferredLanguage" and value equal to the session's langPref attribute.

2.9.3 Understanding Persistent Login

With Access Manager, a user needs to re-authenticate after a period of session inactivity defined by the Idle Timeout parameter (default is 15 minutes) and once the session expires, due to the value of the Session Lifetime parameter (default is 8 hours). The Persistent Login functionality offers administrators the option to skip user re-authentication for a considerably longer period of time should the user opt in - allowing a user two weeks or a month significantly improves convenience. Persistent Login (sometimes referred to as Remember Me or Keep Me Signed In) can be enabled or disabled with the period of time being configurable. It is disabled by default. Persistent Login is enabled in the `oam-config.xml` global configuration file. The appropriate Application Domain must also explicitly allow Persistent Login. When enabled globally, the user login page will have a Keep Me Signed In checkbox and, when checked, the user receives an RMTOKEN. Once the user's session expires or times out, a user with an RMTOKEN will not be challenged if the resource is in the Application Domain that allows Persistent Login and if its authentication level is adequate. If the user tries to access a resource in an Application Domain that has not opted in, the user will be challenged for credentials even if the authentication level is adequate. (If the user does not opt in when logging in, reauthentication will be prompted after a session expiration or inactive timeout.)



Note:

If the Application Domain 'Session Idle Timeout' is specified, Persistent Login cannot be enabled.

The following behaviors are pertinent to the Persistent Login functionality.

- If enabled for the user logged in to Access Manager from a device browser, closing and reopening the browser does not require reauthentication within the defined Persistent Login time period
- Session activities will be reflected in the Audit data.
- When the time period expires, the end user is asked to authenticate again.
- When attempting to access applications from a different device (or even a different process/browser in the same device), the end user will be asked to authenticate again.
- When the user clicks log out, the OAM_RM token is deleted and they user must log in again. Session termination by an administrator will have the same effect.
- As the OAM_RM token is based on credentials entered at the time of token creation, any event that changes the password status will invalidate the token and force the user to re-authenticate. This includes:
 - Password expiration
 - Password reset by administrator
 - Password changed by the user on a different device
 - User deleted or locked by the administrator
- To address a stolen device scenario, the administrator can terminate all sessions for all devices/browsers of a user. The user will need to re-authenticate but has the option to enable Persistent Login on the login page
- Application triggered re-authentication forces the user to re-authenticate even if Persistent Login is enabled as the application is intentionally challenging the user before doing a sensitive operation.
- When a user navigates from an application which allows Persistent Login to one that does not, although the user is logged in automatically, the application which does not allow Persistent Login will challenge the user to enter credentials.
- Persistent Login is not available in application triggered login pages.

The following topics provide additional details:

- [Enabling Persistent Login](#)
- [Troubleshooting Persistent Login](#)

2.9.3.1 Enabling Persistent Login

The feature is not enabled by default.

To enable Persistent Login globally:

1. Connect to WebLogic Server using `connect()`.
Provide the username and password when prompted.
2. Run the command:

```
configurePersistentLogin(enable="true", validityInDays="30",
maxAuthnLevel="2", userAttribute="obPSFTID")
```
3. Create a new Authentication Scheme for Persistent Login using the values in the following table.
See [Managing Authentication Schemes](#).

The 'Keep me signed in' check box will be displayed only when accessing a resource protected by this scheme.

Attribute	Value
Name	PersistentLoginScheme (or any name)
Description	any description
Authentication Level	2
Challenge Method	FORM
Challenge Redirect URL	/oam/server/
Authentication Module	LDAPPlugin
Challenge URL	/pages/login.jsp
Context Type	default
Context Value	/oam
Challenge Parameters	enablePersistentLogin=true

4. Click the Application Domains link in the Launch Pad.
5. Click the Application Domain for which you will use this PersistentLoginScheme and change its Authentication Scheme as documented in this sub procedure.
See [Defining Authentication Policies for Specific Resources](#).
 - a. Click the Authentication Policies tab in the appropriate Application Domain.
 - b. Change the Authentication Scheme for the Protected Resource Policy to PersistentLoginScheme. This allows persistent login for this policy.

 **Note:**

The Public Resource Policy should not be modified.

6. Click the Application Domain under which you will create a Response for all configured Authorization Policies as documented in this sub procedure.
There may be multiple authorization policies and this needs to be done for all.
See [About Constructing a Policy Response for SSO](#).
 - a. Click the Authorization Policies tab in the appropriate Application Domain.
 - b. One at a time, click an Authorization Policy in this Application Domain to open its configuration tab.
 - c. Click Responses.
 - d. Click Add to create an Authorization Response in the Application Domain.
 - e. Enter the following values in the displayed Add Response pop-up and click Add.

Attribute	Value
Type	Session
Name	allowPersistentLogin

Attribute	Value
Value	true

NOTE: To disable Persistent Login for an Application Domain you must disable Authorization Responses by changing the value of the Value attribute in the Add Response pop-up to *false*.

- i. Go to “Conditions” tab.
- ii. Click Add. Provide Name=TRUE, Type=TRUE.
- iii. Click Add Selected.
- iv. Go to “Rules” tab.
- v. In the “Allow Rule” section move the condition TRUE (true) from “Available Conditions” to “Selected Conditions” section.
- vi. Click Apply.

Perform this procedure for all Authorization Policies before moving on to the next step.

7. Access a resource protected by this scheme.
The 'Keep me signed in' checkbox is displayed on the login page.
8. Provide valid credentials and select 'Keep me signed in'.
9. Close and re-open the browser.
10. Access the same resource.
You will be logged in automatically without asking for credentials.



Note:

Persistent Login can also be enabled and disabled using WLST. See the WebLogic Scripting Tool Command Reference for Identity and Access Management for details on the `configurePersistentLogin` command.

2.9.3.2 Troubleshooting Persistent Login

When enabling Persistent Login using WLST, an LDAP attribute named `obpsftid` is defined to store the Persistent Login properties. When the user is locked, the `obpsftid` attribute needs to be updated but the `oamSoftwareUser` does not have sufficient LDAP rights over it. To give `oamSoftwareUser` permission:

1. Copy the LDIF data below and paste it into a file that you will save as `oam_user_write_acl_users_obpsftid_template.ldif`.

```
#####
# Copyright (c) 2010, 2011, Oracle and/or its affiliates. All rights reserved.
#
# NAME: idm_idstore_groups_acl_template.ldif
#
# DESCRIPTION:
#
# This file provides appropriate ACLs to user and group containers.
```



```

#
#
# SUBSTITUTION VARIABLES:
#
# %s_UsersContainerDN% : The container in which users reside
# %s_GroupsContainerDN% : The container in which groups reside
#
#####
#
dn: %s_UsersContainerDN%
changetype: modify
delete: orclaci
orclaci: access to attr=(obUserAccountControl, obLoginTryCount,
obLockoutTime, oblastsuccessfullogin, oblastfailedlogin,
obpasswordexpirydate, obver, obLastLoginAttemptDate, oblockedon) by
group="cn=orclFAOAMUserWritePrivilegeGroup,%s_GroupsContainerDN%"
(search,read,compare,write) by
group="cn=orclFAUserReadPrivilegeGroup,%s_GroupsContainerDN%"
(search,read,compare) by
group="cn=orclFAUserWritePrivilegeGroup,%s_GroupsContainerDN%"
(search,read,compare,write)
-
add: orclaci
orclaci: access to attr=(obUserAccountControl, obLoginTryCount,
obLockoutTime, oblastsuccessfullogin, oblastfailedlogin,
obpasswordexpirydate, obver, obLastLoginAttemptDate, oblockedon, obpsftid)
by group="cn=orclFAOAMUserWritePrivilegeGroup,%s_GroupsContainerDN%"
(search,read,compare,write) by
group="cn=orclFAUserReadPrivilegeGroup,%s_GroupsContainerDN%"
(search,read,compare) by
group="cn=orclFAUserWritePrivilegeGroup,%s_GroupsContainerDN%"
(search,read,compare,write)

```

2. Do the following in the created

oam_user_write_acl_users_obpsftid_template.ldif.

- Replace %s_UsersContainerDN% with User Search Base.
- Replace %s_GroupsContainerDN% with Group Search Base.

3. Change to the OID directory and run ldapmodify.

```

$ setenv ORACLE_HOME <OID_INSTALL_LOCATION>
$ cd $ORACLE_HOME/bin
$ ./ldapmodify -h <LDAP server> -p <LDAP port> -D <bind DN> -w
<bindpassword>
-v -f oam_user_write_acl_users_obpsftid_template.ldif

```

Part II

Managing Common and System Configurations

Administrators set common properties for the services integrated into Access Manager, administer data sources, register managed server instances and grant responsibilities to other Administrators.

You can manage common system-wide configuration details for Oracle Access Management . This section contains the following chapters:

- [Managing Common Services and Certificate Validation](#)
- [Delegating Administration](#)
- [Managing Data Sources](#)
- [Managing Server Registration](#)

3

Managing Common Services and Certificate Validation

The properties that are used in common can be configured by the services integrated into Oracle Access Management.

This chapter contains the following sections:

- [Configuration Options in Oracle Access Management Console](#)
- [Available Services of the Common Configuration Section](#)
- [Common Settings](#)
- [Certificate Validation and Revocation](#)

3.1 Configuration Options in Oracle Access Management Console

The Oracle Access Management options and settings are collectively called Configuration. Unless explicitly stated, the Configuration options are shared by all Access Manager servers and services in the domain.

[Figure 3-1](#) shows the Configuration options defined in the new Oracle Access Management Console. You can access these settings by clicking **Configuration** at the top of the Console.

Figure 3-1 Oracle Access Management Configuration Options

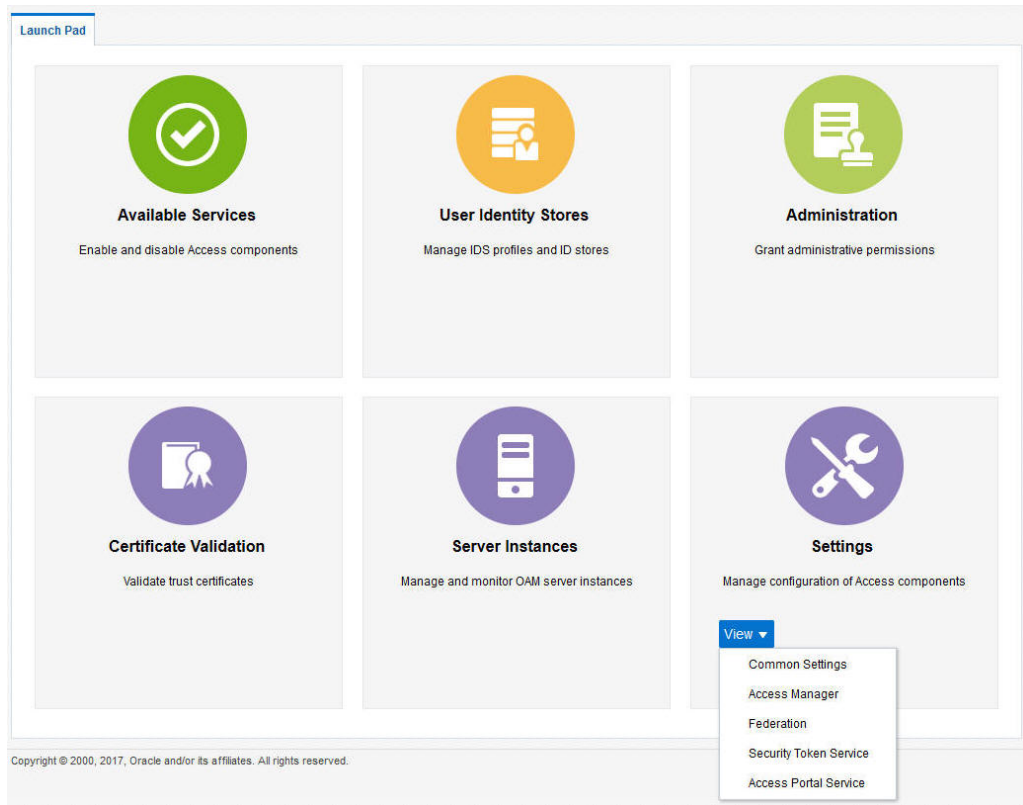


Table 3-1 describes the Configuration options. The items listed apply to all services in the suite.

Table 3-1 Configuration Options

Node	Description
Available Services	See " Available Services of the Common Configuration Section ".
User Identity Stores	See " Registering and Managing User Identity Stores " in Managing Data Sources .
Administration	See Delegating Administration .
Certificate Validation	Provides access to the certificate revocation list and OCSP/CDP settings. See: " Certificate Validation and Revocation ".
Server Instances	Provides access to all registered OAM Server instances. See: Managing Server Registration
Settings > Common Settings	Provides configurations that apply to all Oracle Access Management services including Session properties, Oracle Coherence, Auditing, and Default and System Identity Stores. See: " Common Settings ".

Table 3-1 (Cont.) Configuration Options

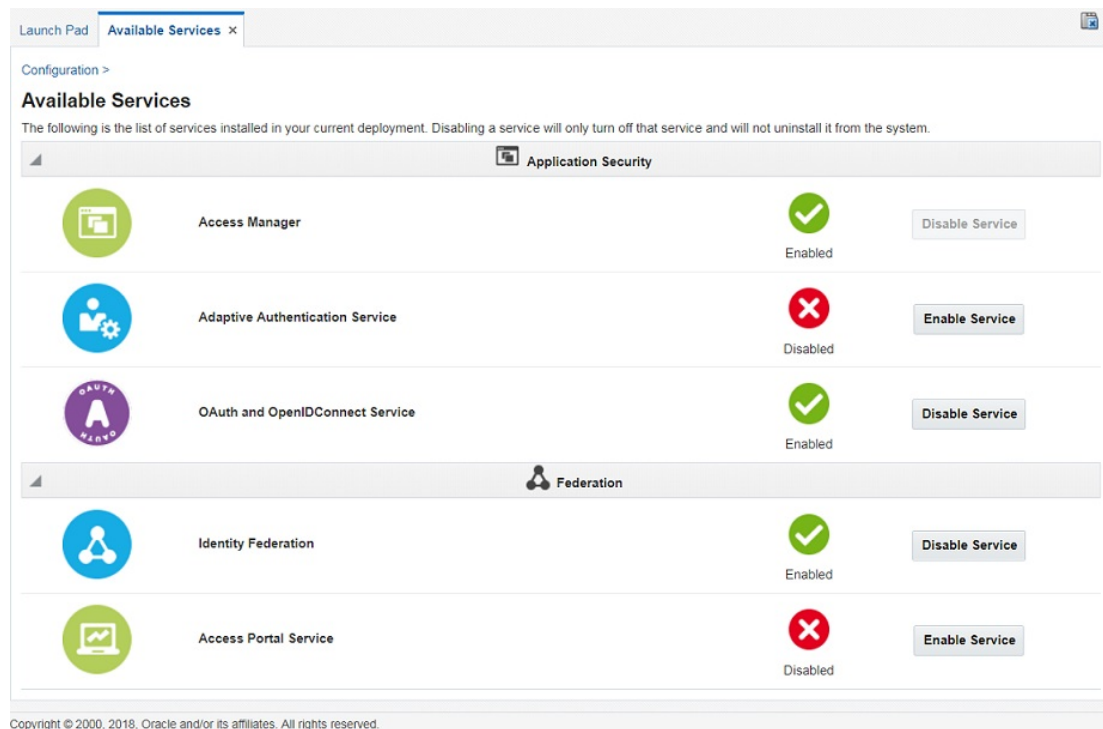
Node	Description
Settings > Access Manager	Provides access to Access Manager operation configurations. See " Managing Common and System Configurations "
Settings > Federati on	Provides access to configurations for Oracle Access Management Identity Federation. See Managing Identity Federation Partners , Managing Settings for Identity Federation and Managing Federation Schemes and Policies .

3.2 Available Services of the Common Configuration Section

Available Services shows the Available Services page of the Common Configuration section, which provides the status of services, and controls to enable or disable a service. Initially, only Access Manager services are enabled.

Oracle Access Management Administrators must enable a service in the Oracle Access Management Console to use the related functionality. The exception to this is Identity Context, which is enabled by default and does not have any controls to disable it.

Figure 3-2 Available Services



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

Table 3-2 Common Services

Service	Description
Access Manager	<p>Access Manager functionality is enabled by default. Access Manager Service is required to set SSO policies, configure Access Manager, as well as Common Configuration, and when REST Services are enabled.</p> <p>Default: Enabled</p> <p>No other services are required for Access Manager and Common Configuration.</p>
Adaptive Authentication Service	<p>Required for adaptive authentication functionality.</p> <p>Default: Enabled</p> <p>See Also: Managing the Adaptive Authentication Service and Oracle Mobile Authenticator.</p>
OAuth and OpenIDConnect Service	<p>Enable this service to activate issuance of standard OAuth tokens and support OpenId Connect flows.</p> <p>Default: Enabled</p> <p>See Also: Managing the Oracle Access Management OAuth Service and OpenIDConnect.</p>
Identity Federation	<p>Must be enabled to manage the federation partners.</p> <p>Default: Disabled</p> <p>Note: The Access Manager service must also be enabled because Identity Federation is another authentication module.</p> <p>See Also: Managing Oracle Access Management Identity Federation.</p>

3.2.1 Enabling or Disabling Available Services

The WebLogic AdminServer and OAM Server must be running to enable or disable an available service.

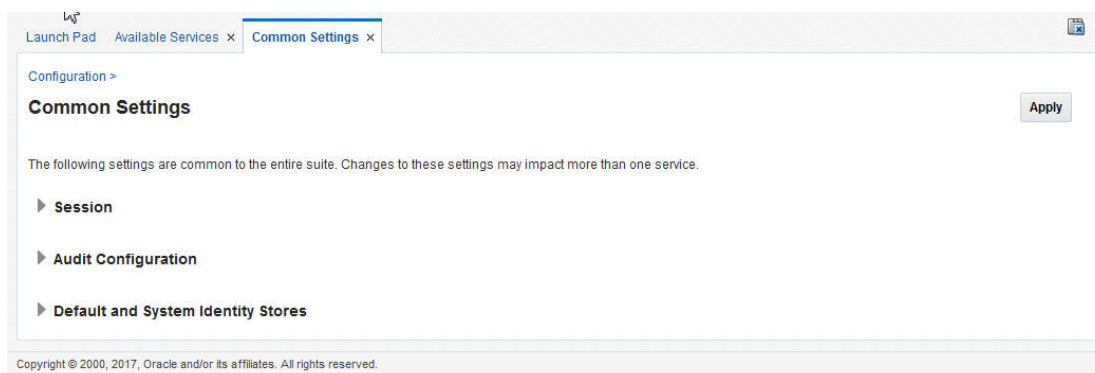
(For details, see [Starting and Stopping Servers in Your Deployment](#)).

1. From the Oracle Access Management Console Launch Pad, click **Available Services** under Configuration.
2. Click **Enable** beside the desired service name (or confirm that the Status check mark is green).
3. Click **Disable** beside the desired service name (or confirm that the Status check mark is red).

3.3 Common Settings

Common Settings apply to all services within the suite.

[Figure 3-3](#) shows the named sections on the Common Settings page, which can be expanded to reveal related elements and values.

Figure 3-3 Common Settings Page (Collapsed View)

Oracle Access Management Administrators can control and specify parameters used by the entire suite, not just a single service, as introduced in [Table 3-3](#).

Table 3-3 Common Settings

Tab Name	Description
Session	Session configuration refers to the process of managing the lifecycle requirements of a session.
Audit Configuration	Oracle Access Management 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 Management auditing configuration is recorded in <code>oam-config.xml</code> . See Also: " Using the Oracle Access Management Console for Audit Configuration ".
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:**

Details for other operations common to all OAM components:

- [Logging Component Event Messages](#)
- [Monitoring Oracle Access Management Performance and Access Manager Health](#)

The following sections have more information.

- [Managing Common Settings](#)

3.3.1 Managing Common Settings

Users with valid Oracle Access Management Administrator credentials can perform the following task to display the Common Settings page and perform changes. To manage common settings, the OAM Server must be running.

(For details, see [Starting and Stopping Servers in Your Deployment](#).)

1. At the top of the Console, click **Configuration**.
2. In the Configuration Launch Pad, select **Common Settings** from the **View** menu in the **Settings** section.
3. **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 (minutes)
 - Idle Timeout (minutes)
 - Maximum Number of Sessions per User
 - c. Click **Apply** to submit your changes.
 - d. See Also: [Maintaining Access Manager Sessions](#).
4. **Audit Configuration:**
 - a. Expand 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 Enabled
 - Filter preset (select from the list to define verbosity of audit data)
 - Audit Configuration Table: Use Add (+) or Delete (x) buttons to specify users.
 - c. Click **Apply** to submit the Audit Configuration (or close the page without applying changes).
 - d. See Also: [Auditing Administrative and Run-time Events](#).
5. **Default Store and System 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 [About using the System Store for User Identities](#) for more information.

3.4 Certificate Validation and Revocation

The Certificate Validation module is used by the Security Token Service to validate X.509 tokens and to verify whether or not the certificates have been revoked.

It supports the following options:

- A Certificate Revocation List (CRL) is a list of certificates (identified by serial numbers) that have been revoked. Revoked certificates are listed with a reason, an issue date, and the issuing entity. (In addition, each list contains a proposed date for the next release.) Entities presenting these (revoked) certificates should

no longer be trusted. When a potential user attempts to access a server, the server allows or denies access based on the CRL entry for the particular user. For more information, see [Enabling the Certificate Revocation List Functionality](#)

- The Online Certificate Status Protocol (OCSP) was developed as an alternative to CRLs. OCSP specifies how the client application that requests information on a certificate's status will obtain it from the server that responds to the request. An OCSP responder can return a signed response signifying that the certificate specified in the request is either *good*, *revoked* or *unknown*. If the OCSP cannot process the request, it returns an error code. For more information, see "[Enabling OCSP Certificate Validation](#) " and [Additional OCSP Configurations](#)
- A CRL Distribution Point extension (CDP extensions) contains information regarding the location of Certificate Revocation Lists (CRLs) and OCSP servers. You o use the Administration Console to define these points. For more information, see [Enabling CRL Distribution Point Extensions](#)

3.4.1 Enabling the Certificate Revocation List Functionality

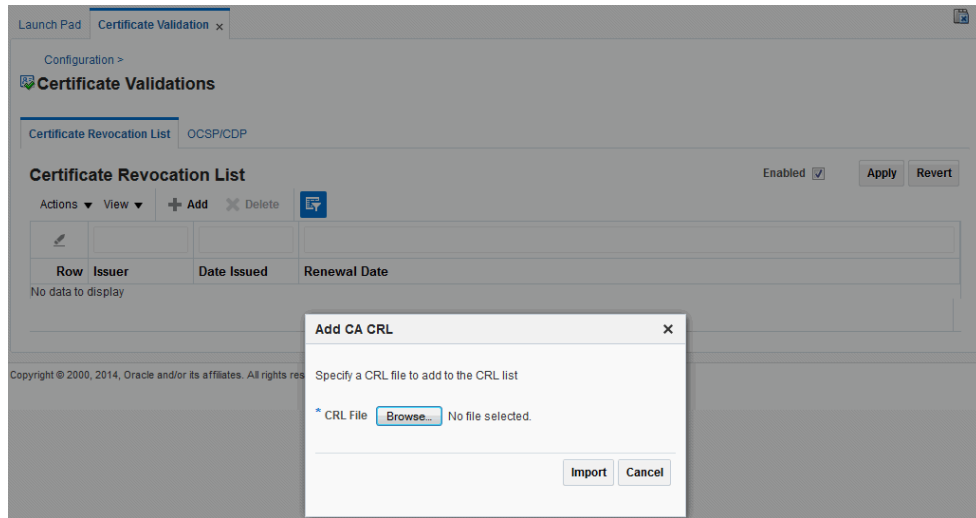
Users with Oracle Access Management Administrator credentials can use the following procedure to enable the CRL functionality and import a current Certificate Authority Certificate Revocation List (CA CRL). Before beginning, you should have your CA CRL ready to import.

To enable:

1. In the Configuration Launch Pad section of the Oracle Access Management Console, click **Certificate Validation**.
The Certificate Revocation List tab is displayed.
2. Confirm that the Enabled box is checked.
3. Add or remove a CRL.
 - **Add:** Click the Add (green plus sign) button, browse for the CRL file, select it, and click Import.
 - **Remove:** Click the name of the list in the table, click the Delete (x) button, and confirm when asked.

[Figure 3-4](#) is a screenshot of the pop-up window used to add a CA CRL to the CRL List using the Administrative Console.

Figure 3-4 Certificate Revocation List Dialog Box



4. Click **Apply** to save the configuration.
5. Proceed to "[Enabling OCSP Certificate Validation](#)".



Note:

To search for CRLs in the table, enable Query by Example from the View drop-down. Enter filter strings in the header fields displayed and hit Enter.

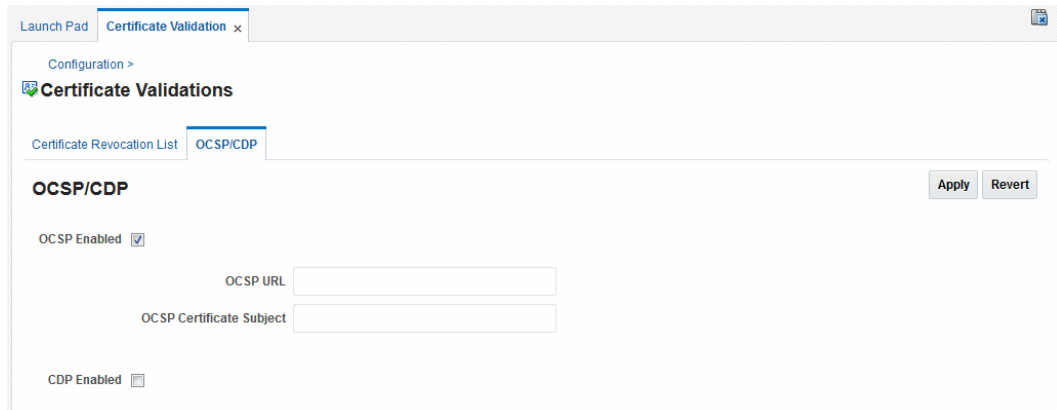
3.4.2 Enabling OCSP Certificate Validation

Users with Oracle Access Management Administrator credentials can use the following procedure to enable the OCSP. Before you begin, you should have the URL of the OCSP service ready to import.

To enable:

1. Under the Configuration section of the Oracle Access Management Console, click Certificate Validation.
The Certificate Revocation List page is displayed. Confirm that the Enabled box is checked.
2. Click the OCSP/CDP tab.
 - a. Enable OCSP.
 - b. Enter the URL of the OCSP Service.
 - c. Enter the Subject DN of the OCSP Service.
 - d. Save this configuration.

Figure 3-5 illustrates how to add an OCSP URL using the Administration Console. See "[WLST configureOAMOCSPCertValidation](#)" for details on how to do this using the WLST command.

Figure 3-5 OCSP/CDP Settings

3. Proceed to "Enabling CRL Distribution Point Extensions".

3.4.3 Enabling CRL Distribution Point Extensions

Users with Oracle Access Management Administrator credentials can use the following procedure to add CRL distribution points in issued certificates.

To enable:

1. Under the Configuration section of the Oracle Access Management Console, click Certificate Validation.

The Certificate Revocation List page is displayed. Confirm that the Enabled box is checked.

2. Open the OCSP/CDP tab.
 - a. Enable CDP.
 - b. Save this configuration.

Figure 3-5 illustrates this.

3.4.4 Additional OCSP Configurations

Support for HTTP Proxy and multiple OCSP Responder configurations are added in Oracle Access Manager.

The following example illustrates the current Certificate Validation Module configuration.

Certificate Validation Module Configuration

```
<Setting Name="CertValidationModule" Type="htf:map">
  <Setting Name="certpathvalidationocspcertsubject"
    Type="xsd:string"></Setting>
  <Setting Name="certpathvalidationocspurl" Type="xsd:string"></Setting>
  <Setting Name="certvalidationcrlstorelocation"
    Type="xsd:string">/scratch/maymaria/installed/wlsHome/user_projects/
domains/base_domain/config/fmwconfig/amcrl.jar</Setting>
  <Setting Name="defaulttrustcastorelocation"
    Type="xsd:string">/scratch/maymaria/installed/wlsHome/user_projects/
domains/base_domain/config/fmwconfig/amtruststore</Setting>
  <Setting Name="defaulttrustcastoretype" Type="xsd:string">jks</Setting>
  <Setting Name="certpathvalidationcdpenabled"
```

```

        Type="xsd:boolean">>false</Setting>
    <Setting Name="certpathvalidationcrlenabled"
        Type="xsd:boolean">>false</Setting>
    <Setting Name="certpathvalidationocspenabled"
        Type="xsd:boolean">>false</Setting>
</Setting>

```

The following sections contain configuration information for these new features.

- [WLST updateHTTPProxyConfig](#)
- [WLST configureOAMOSCSPCertValidation](#)
- [Configuring Multiple OCSP Responders](#)

3.4.4.1 Configuring Multiple OCSP Responders

Certificate authentication currently supports authentication against a single OCSP responder as documented in "[Enabling OCSP Certificate Validation](#)". Support for multiple OCSP responders has been added since the responder URL is now part of the certificate's Authority Information Access Extension.

To support multiple OCSP Responders, the three lines of configuration from the following example of Multiple OCSP Responder Configuration must be added to the top of the Certificate Validation Module configuration section (illustrated in the following example).

Multiple OCSP Reponder Configuration

```

<Setting Name="CertValidationModule" Type="htf:map">
    <Setting Name="certpathvalidationocspurltocamap" Type="htf:map">
        <Setting Name="<url_value>" Type="xsd:string">
            <ocsp_responder_subject></Setting>
        </Setting>
    <Setting Name="useJDKOCSP" Type="xsd:string">>false</Setting>
    ...
</Setting>

```

Configure the first and second lines to enable multiple OCSP responders.

- Set `certpathvalidationocspenabled` to `true`.
- Update the `certpathvalidationocspurltocamap` configuration. It is of type `Map`, the key is the OCSP Responder URL (URL Encoded) and the value is the OCSP Responder's Certificate subject.

```

<Setting Name="certpathvalidationocspurltocamap" Type="htf:map">
    <Setting Name=" http%3A%2F%2Flocalhost%3A9797" Type="xsd:string">
        emailAddress=sagar@pspl.com,CN=ps2436,OU=OBLIX-QA,O=PSPL,
        L=PUNE,ST=MAHA,C=MY</Setting>
</Setting>

```

- (Optionally) set values for `certpathvalidationocspcertsubject` and `certpathvalidationocspurl`.

The Responder URLs will be fetched first from the `AuthorityInformationAccess` extension of the user's X.509 certificate and second from `Modules/Plugin` (`CertValidation`). The Responder Subjects will be fetched first from the defined configuration map and second from the `Module/Plugin` (`CertValidation`) configuration. In cases where these configurations are not found, the OCSP validation will fail.

Configure the third line to provide backward compatibility for those who want to use JDK OCSP validation rather than the new OAM OCSP Checker. By default, the JDK OCSP Checker is enabled. When configuring the OAM OCSP Checker using the WLST command, the flag is set to false. For more information on the WLST command, see [WLST configureOAMOCSPCertValidation](#).

Depending on the Certificate Validation Module configuration there are three different options as documented in [Table 3-4](#).

Table 3-4 OCSP Responder Configuration Options

Configuration	OCSP Configuration (certpathvalidationocspenabled)	CRL Configuration (certpathvalidationcrlenabled)	JDK/OAM OCSP Configuration (useJDKOCSP)
No OCSP Checking Simple certificate validation is performed during OAM X-509 authentication	False	False	False
OAM OCSP X-509 authentication performs certificate validation with OCSP checking using the new OAM OCSP Checker.	True	True/False (does not matter)	False
JDK OCSP X-509 authentication performs certificate validation with OCSP checking using the JDK OCSP Checker.	True	True	True

To enable OCSP validation to be done using one configured responder URL, set the `certpathvalidationcrlenabled` and `certpathvalidationocspenabled` properties to true and set values for the `certpathvalidationocspcertsubject` and `certpathvalidationocspurl` properties. If these properties are not set, OCSP validation will be done using the responder URL defined within the user certificate's AIA Extension. If no URL is defined, OCSP validation will fail.

3.5 WLST updateHTTPProxyConfig

The Oracle Access Manager OCSP checker can perform authentication against OCSP responders that are outside an enterprise's intranet through HTTP Proxy.

Use the `updateHTTPProxyConfig` WLST command to configure the OAM OCSP checker to use HTTP proxy.

Description

Adds or updates proxy information.

Syntax

```
updateHTTPProxyConfig(proxyHost, proxyPort, conTimeOut)
```

Argument	Definition
<i>proxyHost</i>	Mandatory. The host name of the proxy.
<i>proxyPort</i>	Mandatory. The port number of the proxy.
<i>conTimeOut</i>	Mandatory. The connection timeout in milliseconds.

Example

```
updateHTTPProxyConfig(proxyHost="hostname.example.com", proxyPort="8888",
  conTimeOut="600")
```

3.6 WLST configureOAMOSCSPCertValidation

You can use the `configureOAMOSCSPCertValidation` online command to update the OAM OCSP configuration.

It includes the following:

- Updating or adding an OCSP responder URL and subject details to the "certpathvalidationocspurltocamap"
- Clearing the newly added configuration; for example, "certpathvalidationocspurltocamap"
- Setting or unsetting the "useJDKOCSP" flag to enable or disable JDK OCSP

Description

Updates the OAM OCSP configuration by adding/modifying the OCSP responder URL and subject details in the `certpathvalidationocspurltocamap` property and enabling/disabling the use of the JDK OCSP Checker.

Syntax

```
configureOAMOCSPCertValidation(url, subject, clear (optional),
  display (optional), useJDKOCSP (optional))
```

Argument	Definition
<i>url</i>	Mandatory. Takes as a value the valid URL.
<i>subject</i>	Mandatory. Takes the details being modified.
<i>clear</i>	Optional. Takes a value of true or false.
<i>display</i>	Optional. Takes a value of true or false.
<i>useJDKOCSP</i>	Optional. Takes a value of true or false.

Examples

The following example enables the OAM OCSP and sets the Responder URL and subject.

```
configureOAMOCSPCertValidation(url="http://sample:9898",  
    subject="cert-subject-detail")
```

The following example enables the OAM OCSP and updates the Responder URL and subject.

```
configureOAMOCSPCertValidation(url="http://sample:9898",  
    subject="details changed/updated")
```

The following example disables and clears the OAM OCSP.

```
configureOAMOCSPCertValidation(url="http://sample:9898", subject="subject-detail",  
    clear="true")
```

The following example enables/disables the JDK OCSP.

```
configureOAMOCSPCertValidation(url="http://sample:9898",  
    subject="details changed/updated", useJDKOCSP="true")
```

4

Delegating Administration

Delegating administration allows a high-level administrator to grant responsibilities to other, more local administrators. This is useful in large organizations where it may be necessary to administer thousands or millions of users. With this release of Oracle Access Management, a System Administrator can delegate administration of Application Domains to other administrators. An Application Domain Administrator role has been developed towards the end.

The following topics provide an overview of delegating administration, such as determining what rights you want to grant to another user:

- [Understanding Administrator Roles](#)
- [About Delegating the Identity Store](#)
- [Assigning Roles Using the Administration Console](#)
- [Understanding the Container Security Framework and MBeans](#)
- [Using the Remote Registration Utility](#)
- [About Auditing Reports](#)

4.1 Understanding Administrator Roles

After you complete the installation, Access Manager has a set of pre-defined roles that you can assign to administrators, such as the Access Manager System Administrator.

See [About Oracle Access Management Administrators](#).

You can assign the following to Access Manager system administrators:

- All Application and component policy objects (including Resources, Authentication Policies, Authorization Policies, and Token Issuance Policies)
- Shared components (including Authentication Schemes, Host Identifiers, and Resource Types)
- System configuration (including Common Configuration, Access Manager settings and Authentication Modules, Security Token Service Settings, Custom Tokens, Endpoints, Templates and Profiles, and Access Manager Agents and Security Token Service Partners)
- Agents and partners

A System Administrator can grant the rights to administer an Application Domain to an Application (Domain) Administrator. (A virtual Access Manager Administrator group is defined and mapped to the Application Administrator role.) An Application Administrator can further delegate the rights to administer one or more of their Application Domains to other Application Administrators. An Application Administrator can create and edit Resources, Authentication Policies and Authorization Policies. These rights are scoped to one or more Application Domains.



Note:

Only the System Administrator can assign roles to users; users cannot further delegate that role to others.

The System Administrator, Application Administrator and Help Desk Administrator roles are mutually exclusive; that is, a group or user can be assigned to only one such administrator role. However, the Application Administrator and Agent Administrator roles can be assigned to the same user or group.

[Table 4-1](#) documents details about the pre-defined administrator roles.

Table 4-1 Roles for Delegating Administration

Role Name	Description
System Administrator	<p>Access to entire Oracle Access Management Console including policy creation and system configuration; encompasses the privileges to manage all system configurations, policy objects, Access Manager Settings, Agents, Authentication Modules, Authentication Schemes, Host Identifiers, Resource Types, Federation Partners and Enterprise Single Sign-on policies. Additionally, Security Token Service Settings, Partners, Custom Tokens, Endpoints, Templates and Profiles can be managed.</p> <p>NOTE: The System Administrator does not support seamless failover. If one server goes offline, the System Administrator can re-login and continue on the other server(s) in the cluster.</p>
Application Administrator	<p>Access to policy creation and resources in the specified Application Domain. This role has access to the Application Registration Quick Wizard link.</p>
Help Desk Administrator	<p>Access to the Help Desk console.</p>
Agent Administrator	<p>Access to the Agent configuration pages. This role has access to the Agent Registration Quick Wizard link.</p>
Authenticated User	<p>Access to the Self Service Launch Pad and pages.</p>

See [Oracle Access Management Console and the Policy Manager Console](#).

See [Understanding the Oracle Access Management Console](#).

4.2 About Delegating the Identity Store

The Access Manager System Identity Store is used to enforce authentication and authorization during the execution of administrative operations.

The LDAP Directory defined as the System Identity Store will contain all the administrators having access to the Administration Console. An administrator can define a new User Identity Store and select one of the existing profiles as the System Identity Store but only the System Administrator can modify the current System Identity Store or switch to a new one.

When migrating to a new Identity Store, if users from the new store are assigned Access Manager roles, those privileges become active and are enforced by Access Manager. The administrator will be responsible for removing any delegated administration privileges for the new Identity Store and the Access Manager Administrator group will be mapped to the Administrator role of the new identity store.

**Note:**

If the user currently logged in does not have the necessary administrator roles in the new system store, the Administration Console will log out or refresh so that it is compliant with the roles assigned to the current administrator.

4.3 Assigning Roles Using the Administration Console

The System Administrator can use the Oracle Access Management Console to assign roles to users or groups that cover specific Application Domains. Users can be assigned multiple roles as long as the functionality doesn't overlap.

For example, if user X is assigned Global Policy Administrator, the user cannot be granted Policy Administrator for the HR domain because the latter is a child of the former.

**Note:**

Roles can be assigned only to users or groups from the system/default store.

From a high level:

1. When delegating administration for a specific policy object or a set of policy objects, the delegator selects the item(s) and assigns the user(s), group(s), LDAP Search Filter(s) or Domain System role(s) to it.
2. When delegating administration for all objects of a specific type, the delegator will select the user(s), group(s), LDAP Search Filter(s) or Domain System role(s) and grant the rights to administer the objects of that type to the selected. In this case, the administrator can't select objects for which administration is being delegated; the administrator will select a role that is granted to the appropriate delegatee with a specific right.

4.4 Understanding the Container Security Framework and MBeans

MBeans that enforce authentication and authorization using the container security framework are published using the Portable JMX Framework.

Types of MBeans:

- The Configuration Service MBeans are used for configuring the Certificate Validation Module, the STS Endpoints, Templates & Profiles, and the STS Settings & Custom Tokens.

- The Partner and Trust Store Service MBeans are used for managing the STS Partners.

At runtime, the JMX Framework will authenticate the client during the connection operation and ensure that the client belongs to the role specified in the MBean security annotations. Because of this, the Access Manager System Identity Store needs to be configured as an Authentication Provider in the security realm of the domain. Additionally, users accessing the MBeans will need to be assigned the following role depending on the container:

- WebLogic: Admin
- WebSphere: Admin or Configurator

4.5 Using the Remote Registration Utility

The Remote Registration Utility (RREG) is also governed by the roles assigned to the user invoking them. When using RREG to remotely register agents, the administrator provides credentials that allows the RREG client to successfully connect and authenticate to the RREG Access Manager Server, this in turn, propagates the client's identity to the Access Manager components that will enforce the appropriate administration roles.

The following might occur when running the RREG based on the administrator's role:

- In a creation operation:
 - A new agent entry can be provisioned.
 - A HostID for that Agent can be created.
 - An Application for that agent might be created.
 - Resources might be added to the new Application using the newly created HostID.
- In an update operation:
 - Agent settings can be changed.
 - A HostID for that agent can be changed.
 - An Application for that agent can be created if it does not exist.
 - Resources can be added to the Application.

The RREG administrator must be assigned roles to ensure successful completion of the administrative operations.

- The System Administrator role to create/update an Agent.
- The OAM Shared Component Administrator / System Administrator role to create/update an HostID entry.
- The OAM Domain Administrator role / System Administrator to create/update an Application and create/configure Resources.

After executing the RREG command, the administrator will be set as the delegated administrator for the created Application, Agent and HostID.

4.6 About Auditing Reports

Auditing becomes even more critical when administration has been delegated to several users. All policy object and system configuration operations performed by administrators through the Administration Console or programmatically are logged and informational reports can be generated.

See [Introduction to Oracle Fusion Middleware Auditing](#).

5

Managing Data Sources

The term *data source* is a Java Database Connectivity (JDBC) term used within Oracle Access Management to refer to a collection of user identity stores or a database for policies. These data sources must be registered using the Oracle Access Management Console in order to be accessed.

The following topics describe the tasks to register and administer data sources using the Oracle Access Management Console. The information is common to all services available through the Oracle Access Management Console.

- [Data Sources for Oracle Access Management](#)
- [Registering and Managing User Identity Stores](#)
- [Managing the Identity Directory Service User Identity Stores](#)
- [Managing Administrator Roles](#)
- [Managing the Policy and Session Database](#)
- [Introduction to Oracle Access Management Keystores](#)
- [Integrating a Supported LDAP Directory with Oracle Access Manager](#)

5.1 Data Sources for Oracle Access Management

Oracle Access Management supports several types of data sources that are typically installed for the enterprise.

[Table 5-1](#) describes each data source is a storage container for the various types.

Table 5-1 Data Sources for Oracle Access Management

Data Source	Description
Database	<p>A collection of information that is organized and stored so that its content can be easily accessed, managed, and updated.</p> <ul style="list-style-type: none">• Access Manager policy data, including password management data, must be stored in a database that is extended with the Access Manager-specific schema and registered with Access Manager. See Managing the Policy and Session Database .• Session Store: By default, Access Manager session data is stored within in-memory caches that is migrated to the policy store. In production environments, you can have an independent database for policy data and another for session data. For details about sessions and session data, see Maintaining Access Manager Sessions.• 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 administrative and run time events, see Auditing Administrative and Run-time Events.

Table 5-1 (Cont.) Data Sources for Oracle Access Management

Data Source	Description
User Identity Store	<p>Central LDAP storage in which an aggregation of user-oriented data is kept and maintained in an organized way. (Access Manager does not include identity services; there is no native user, group, or role store.) The identity store must be installed and registered with Access Manager 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). During the initial deployment process, described in the <i>Installing and Configuring Oracle Identity and Access Management</i>, the embedded LDAP store is used as the User Identity Store.</p> <p>Oracle recommends that you use only the Oracle Access Management Console or WebLogic Scripting Tool (WLST) commands for changes; do not edit oam-config.xml.</p> <p>By default, Access Manager 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.</p>
Oracle Access Management configuration data file: oam-config.xml	<p>During the initial deployment process, described in the <i>Installing and Configuring Oracle Identity and Access Management</i>, Oracle Access Management configuration data is stored in an XML file: oam-config.xml.</p> <p>See "Updating OAM Configuration".</p>
Keystores	<p>Several keystores are associated with Oracle Access Management services as described in "Introduction to Oracle Access Management Keystores".</p> <ul style="list-style-type: none"> • Embedded Java Keystore: Used for certificates for Simple or Certificate-based communication between OAM Servers and Webgates. The keystore bootstrap occurs upon initial AdminServer startup after running the Configuration Wizard. See: "Access Manager Security Keys and the Embedded Java Keystore" • Security Token Service Keystores: Access Manager and Security Token Service keystore should always be different. For more information, see "Access Manager Keystores". • Identity Federation Keystores: Keystore settings enable you to create aliases (a short hand notation) for keys in the keystore. See: "Identity Federation Keystore"

[Table 5-2](#) contains the Oracle Access Management services and links to information about the data sources used for each.

Table 5-2 Data Sources for Oracle Access Management Services

Service	Description
Access Manager	<p>Access Manager supports multiple Identity Stores and provides SSO authentication using data sources:</p> <ul style="list-style-type: none"> • Registering and Managing User Identity Stores • Managing the Policy and Session Database • Updating OAM Configuration • Access Manager Security Keys and the Embedded Java Keystore

Table 5-2 (Cont.) Data Sources for Oracle Access Management Services

Service	Description
Identity Federation	<p>Identity Federation supports multiple Identity Stores which can be assigned on a per Identity Partner basis. Each Identity Store must be registered with Access Manager. If no Identity Store is defined in the Identity Partner, the designated Default Store is used.</p> <ul style="list-style-type: none"> • About Using Multiple Identity Stores • Identity Federation Keystore • Administering Identity Federation As A Service Provider

**See Also:**

- ["Managing Global Password Policy"](#)
- ["Configuring OAM WebGate and Authentication Policy for DCC"](#)
- ["About using the System Store for User Identities"](#)
- [Auditing Administrative and Run-time Events](#) for details about Audit data stored within audit files or a separate Oracle Database

The following sections contain additional details.

- [Updating OAM Configuration](#)
- [About the Default LDAP Group](#)

5.1.1 Updating OAM Configuration

From 12.2.1.3 release onwards, OAM configuration resides only in the OAM dbstore (OAM SCHEMA).

Therefore, editing `oam-config.xml` in `$DOMAIN_HOME` does not have any effect. To make changes to configuration, follow this procedure:

1. Create a <working directory> and change to the directory:

```
mkdir <working directory>
```

```
cd <working directory>
```

2. Inside the directory create a `dbschema.properties` file that contains the following:

```
oam.entityStore.ConnectionString=jdbc:oracle:thin:@<DB_HOST>:<DB_PORT>/
<DB_SERVICE_NAME>
oam.entityStore.schemaUser=<PREFIX>_OAM
oam.entityStore.schemaPassword= <password>
oam.importExportDirPath=<path to which configuration is exported or imported>
oam.frontending=params=host;port;protocol
```

For example:

```
oam.entityStore.ConnectionString=jdbc:oracle:thin:@dbhost.example.com:1521/
pdb1.example.com
```

```
oam.entityStore.schemaUser=DEV_OAM
oam.entityStore.schemaPassword= <password>
oam.importExportDirPath=<work directory>
oam.frontending=params=host;port;protocol
```

3. Export configuration from the dbstore using config-utility.jar:

```
java -cp $ORACLE_HOME/oam/server/tools/config-utility/config-
utility.jar:$ORACLE_HOME/oracle_common/modules/oracle.jdbc/ojdbc8.jar
oracle.security.am.migrate.main.ConfigCommand $DOMAIN_HOME export
dbschema.properties
```

This will export the oam-config.xml to the <working directory>

4. Modify the <working directory>/oam-config.xml as required.

5. Import the configuration back into the dbstore using config-utility.jar:

```
java -cp $ORACLE_HOME/oam/server/tools/config-utility/config-
utility.jar:$ORACLE_HOME/oracle_common/modules/oracle.jdbc/ojdbc8.jar
oracle.security.am.migrate.main.ConfigCommand $DOMAIN_HOME import
dbschema.properties
```

5.1.2 About the Default LDAP Group

The default LDAP group, **Administrators**, is set during initial deployment using the Oracle Fusion Middleware Configuration Wizard.

For more information, see "[About Oracle Access Management Administrators](#)".

5.2 Registering and Managing User Identity Stores

A User Identity Store is a centralized LDAP repository in which an aggregation of Administrator and user-oriented data is stored and maintained in an organized way.

Oracle Access Management supports multiple LDAP vendors, and multiple LDAP stores can be registered for use by Oracle Access Management and its services. Oracle Access Management addresses each user population and LDAP directory store as an identity domain. Each identity domain maps to a configured LDAP User Identity Store that must be registered with Oracle Access Management. This section provides the information you need to register and manage user identity stores using the Oracle Access Management Console.

- [Understanding User Identity Stores](#)
- [About using the System Store for User Identities](#)
- [About Using Multiple Identity Stores](#)
- [User Identity Store Settings](#)
- [Registering a New User Identity Store](#)
- [Viewing or Editing a User Identity Store Registration](#)
- [Deleting a User Identity Store Registration](#)

 **Note:**

Oracle recommends that you use the Identity Directory Service Profiles to access identity data stores rather than the legacy OAM ID Stores function as it will be deprecated in a future release. The Identity Directory Service is documented in [Managing the Identity Directory Service User Identity Stores](#).

5.2.1 Understanding User Identity Stores

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 for Oracle Access Management. Within the Embedded LDAP, the Administrators group is created with `weblogic` seeded as the default Administrator.

 **Note:**

The Embedded LDAP performs best with fewer than 10,000 users. With more users, consider a separate enterprise LDAP server. In a highly available configuration, Oracle recommends that an external LDAP is used as the User Identity Store. See *Administering Security for Oracle WebLogic Server*.

When attempting to access an Access Manager-protected resource, a user can be authenticated against any store, not simply the designated Default Store. That said, there are a few considerations:

- **System Store:** Only one User Identity Store can (and must) be designated as the System Store. This is used to authenticate Administrators signing in to use the Oracle Access Management Console, remote registration tools, and custom administrative commands in WLST. Thus, Administrators using the Oracle Access Management Console or remote registration utility must have credentials stored in the System 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). For details, see [About using the System Store for User Identities](#)
- **Default Store:** As the name implies, the LDAP store designated as the Default Store is the automatic choice for use by LDAP authentication modules unless you configure use of a different store for the module or plug-in.

 **Note:**

Users attempting to access an Access Manager-protected resource can be authenticated against any user identity store that is registered and defined in the authentication scheme while the Security Token Service uses only the Default User Identity Store. For example, when adding User Conditions to a Token Issuance Policy, the identity store from which the users are chosen must be the Default Store.

In the Oracle Access Management Console, User Identity Store registrations are organized under the Configuration Launch Pad. Administrators can register, view, modify, and delete User Identity Store registrations using either the Oracle Access Management Console or Customization Commands described in *WLST Command Reference for WebLogic Server*.

5.2.2 About using the System Store for User Identities

Users with valid Oracle Access Management System Administrator credentials can designate a registered user identity store as either the Default Store, the System Store or both.

You can select the Default and System Identity Store configurations using the Oracle Access Management Console as documented in [Managing the Identity Directory Service User Identity Stores](#)

UserIdentityStore1 is the embedded Access Manager LDAP store. After installation, the Oracle Access Management Console and OAM Policy Manager are protected by the IAM Suite Agent. The IAM Suite Application Domain is seeded with the OAM Admin Console Authentication Policy which uses the OAMAdminConsoleScheme authentication scheme. In turn, the OAMAdminConsoleScheme uses the LDAP Authentication module, and the System Store and the LDAP authentication module both use the WebLogic Embedded Identity Store (UserIdentityStore1). The Access Manager Administrator roles are mapped to the enterprise groups and users that belong to the System Store.

5.2.2.1 Using the System Store for User Identities

Changing the System Store impacts the entire identity management (IAM Suite) domain. When you want to change the System Store to a remote identity store, you need to create an Authentication Provider in WebLogic for this remote store.

The remote store provider should be displayed in the list of providers in the WebLogic console. Additionally:

- Ensure the control flags for all providers preceding the new remote store provider are set to SUFFICIENT or OPTIONAL.
- Assign ADMIN, the WebLogic global role, to the enterprise groups or users from this remote store. This can be done by following the steps to prepare the remote store using IDM Config Tool and by referring to the WebLogic documentation.
- If using Oracle Unified Directory as a system store, create IPlanetAuthenticator in WebLogic.

The above configuration should be done and tested before you change the System Store to a remote store. You will also have to change the LDAP authentication module configuration to use the remote store. The remote store can be configured using OAM Identity Store or IDS Profile.

 **Note:**

Administrator login works only when the LDAP Authentication Module (used by the `OAMAdminConsoleScheme`) also uses the System Store. If you set another store as a remote store, ensure that the `OAMAdminConsoleScheme` is modified to avoid a lockout.

When you want to use a WebGate to protect the Oracle Access Management Console (on the AdminServer) and the Policy Manager Console (on the OAM Server), in addition to the above procedure, create an OAM Identity Asserter in WebLogic and enable OAM as the SSO provider in JPS using WLST commands. For details, see [Integrating Oracle ADF Applications with Access Manager SSO](#). You will also have to whitelist the OAM Policy Manager Console host name and port to fully enable WebGate protection.

Information regarding administrator roles can be found in [Managing Administrator Roles](#)

5.2.3 About Using Multiple Identity Stores

Administrators can install and register multiple user identity stores for Oracle Access Management. Each identity store can rely on a different LDAP provider.

When more than one identity store is registered, an Administrator must define:

- The System Store: Administrators can login against the System Store only.
- The Default Store: Comes into play during patching and when using Identity Federation, and Security Token Service.
 - Patching: Oracle recommends that before patching, you designate `UserIdentityStore1` as the Default Store and also update LDAP Authentication Modules to use `UserIdentityStore1` (the Embedded LDAP of Weblogic Server).
 - Identity Federation: Supports multiple identity stores, on a per IdP Partner basis. The specified identity store must be registered like any other store. If no identity store is defined in the IdP Partner, the Default Store is used. For details, see [Administering Identity Federation As A Service Provider](#).
- The specific store to use with each LDAP authentication module or plug-in (and Form or Basic authentication schemes)

External LDAP repositories can provide user, role, and group membership information. A user's group memberships, for example, are calculated at login time and stored for the duration of the session. Information is used as follows:

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

 **Note:**

There is no way to flush a user's group memberships information to force Oracle Access Management to recalculate it at a later date.

Registering user identity stores 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 Management addresses each user population and directory as an identity domain. Each identity domain simply maps to a configured identity store name.

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 Management:

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

For instance, in Oracle Access Management 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.

5.2.3.1 Components of Oracle Access Management that use Identity Stores

The run time and administrative components of Oracle Access Management use identity stores.

[Table 5-3](#) documents the various run time and administrative components of Oracle Access Management.

Table 5-3 Components That Use Identity Stores

Component	Description
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 Condition type of an 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. <code>OAMIdentity</code> 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 <code>OAMIdentity</code> 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 <code>OAM_REMOTE_USER</code> header, a second <code>OAM_IDENTITY_DOMAIN</code> header is set on every request for an authenticated user, such that a consuming application can disambiguate the user if needed.

Table 5-3 (Cont.) Components That Use Identity Stores

Component	Description
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:**

- ["User Identity Store Settings"](#)
- ["Managing Administrator Roles"](#)
- ["About using the System Store for User Identities"](#)
- Access Manager WLST Commands in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference for Identity and Access Management*

5.2.4 User Identity Store Settings

You can create an user identity store from the **System Configuration** tab.

[Figure 5-1](#) illustrates the Create User Identity Store Page, which provides fields where you enter details for your store and default settings that you can edit for your environment. The Store Type drop-down list provides supported choices.

Figure 5-1 Creating User Identity Store Registration

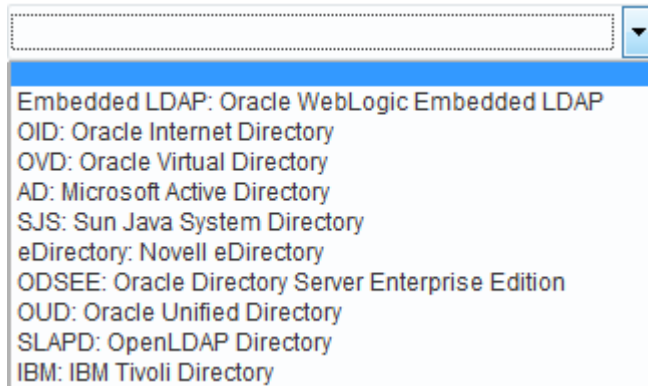
Required settings are identified by the asterisk (*) on the page. [Table 5-4](#) describes each element and is organized by element types.

Table 5-4 User Identity Store Elements

Section	Element	Description
General	Store Name	A unique name for this registration. Use up to 30 characters for the name.

Table 5-4 (Cont.) User Identity Store Elements

Section	Elements	Description
Store Type		A list of all supported LDAP providers from which you can choose. You can have multiple identity stores, as described in " About Using Multiple Identity Stores ".



See Also: [Table 24-6](#).

Description	Optional.
Enable SSL	<p>Click to check this box and indicate that SSL is enabled between the directory server and OAM Server.</p> <p>Using the keytool command line interface, you must also import the appropriate CA root certificate and server certificate to the default JDK keystore (located at \$JAVA_HOME/lib/security/cacerts).</p> <p>NOTE: The CA root certificate can be added to any keystore as long as the appropriate values regarding that keystore are set for the following Java properties. To do this, start the OAM Admin Server and Managed Server instances with the appropriate values and the -D option.</p> <ul style="list-style-type: none"> • javax.net.ssl.trustStore=trust.jks • javax.net.ssl.trustStorePassword=<trustPass> • javax.net.ssl.keyStore=keystore.p12 • javax.net.ssl.keyStoreType=pkcs12 • javax.net.ssl.keyStorePassword=<keyStorePass>
Prefetched Attributes	List of comma-separated user attributes; for example, email, phone, mobile. The OAM server will cache the list of user attributes in memory while it authenticates the user against the identity store. The cached values will be used to compute the Authentication response headers, Authorization policy response headers and Authorization policy conditions. Prefetched attributes provide huge performance improvements by avoiding a round trip to the user identity store. The OAM Administrator has to make sure all the user attributes used in Authentication and Authorization policy response headers and Authorization conditions are defined as prefetched attributes in the user identity store profile.
User Native ID Store	This enables getting the authentication code for natively locked/disabled/pw_must_change code in the LDAP authentication module.

Table 5-4 (Cont.) User Identity Store Elements

Section	Element s	Description
Location and Credentia ls	Location	<p>The URL for the LDAP host, including the port number. Oracle Access Management support 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.</p> <p>Enter one (or more) LDAP URLs in <i>host:port</i> format, Multiple URLs must be separated by a space or new line. There is no need to specify <i>ldap://</i> or <i>ldaps://</i>(which supports SSL_NO_AUTH) in the URL value:</p> <pre>localhost:myhost:7001</pre> <p>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 Management and the browser can handle.</p>
	Bind DN	<p>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.</p> <p>For example:</p> <pre>uid=amldapuser,ou=people,o=org</pre>
Users and Groups	Password	The password of the Principal, which is encrypted for security.
	Login ID Attribute	<p>The attribute that identifies the login ID (user name).</p> <p>For example:</p> <pre>uid</pre>
	User password attribute	<p>The attribute in the user identity store (LDAP directory) which stores the user's password. This is made configurable for added flexibility.</p>
	User Search Base	<p>The node in the directory information tree (DIT) under which user data is stored, and the highest possible base for all user data searches.</p> <p>For example:</p> <pre>ou=people,ou=myrealm,dc=base_domain</pre>
	User Filter Object Classes	The object classes to be included in search results for users, in a comma-separated list of user object class names. For example: <i>user, person</i> .
	Group Name Attribute	<p>The attribute that identifies the group name.</p> <p>Default: <i>cn</i></p>
	Group Search Base	<p>Currently only static groups are supported, with the <i>uniquemember</i> attribute.</p> <p>The node in the directory information tree (DIT) under which group data is stored, and the highest possible base for all group data searches.</p> <p>For example:</p> <pre>ou=groups,ou=myrealm,dc=base_domain</pre>
	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: <i>groups,groupOfNames</i> .

Table 5-4 (Cont.) User Identity Store Elements

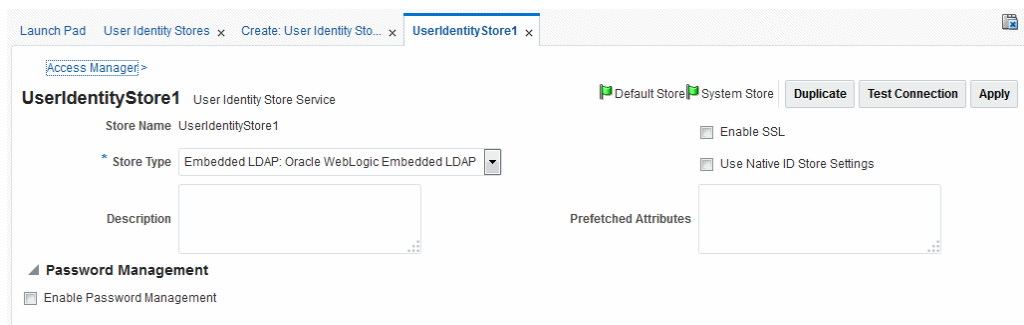
Section	Element s	Description
	Enable Group Members hip Cache	Boolean value for group cache: true or false. Default: true
	Group Cache Size	Integer for the group cache size. Default: 10000
	Group Cache Time-to-Live (seconds)	Integer (in seconds) for Time to Live for group cache elements. Default: 0
Connection 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
	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.
	Referral Policy	One of these values: <ul style="list-style-type: none"> follow: Follows referrals during an LDAP search (Default) ignore: Ignores referral entries during an LDAP search throw: Results in a Referral Exception, which can be caught by the component user.
	Enable Password Management	Enables password policy enforcement against the attribute values listed below. The corresponding options in the password policy must be configured as well.
	Use Oblix User Schema	Enables the use of OBLIX schema instead of standard Oracle schema.
	Global Common ID Attribute	Specifies the User ID attribute name. This attribute will be used as part of the password policy to check that the user ID is not part of the password.
First Name Attribute	Specifies the First Name attribute name. This attribute will be used as part of the password policy to check that the user's first name is not part of the password.	
Last Name Attribute	Specifies the Last Name attribute name. This attribute will be used as part of the password policy to check that the user's last name is not part of the password.	

Table 5-4 (Cont.) User Identity Store Elements

Section	Element	Description
Results Time Limit (seconds)	Results Time Limit (seconds)	<p>Results Time Limit is not supported. Alternatively, use <code>OperationTimeout</code> as described: <code>OperationTimeout</code> specifies the time (in milliseconds) IDS waits for an LDAP request to be acknowledged by the LDAP remote host.</p> <p>Default value is 120000 milliseconds</p> <p>For IDS based profiles, use the <code>modifyLDAPAdapter WLST</code> command to set the timeout value. For example:</p> <pre>modifyLDAPAdapter(adapterName='ADAPTER_NAME', attribute='OperationTimeout', value=120000, contextName='ids')</pre> <p>For original idstore profiles, edit the <code>oam-config.xml</code> and set the value inside the idstore snippet. For example,</p> <pre><Setting Name="CONN_TIMEOUT" Type="xsd:string">120000</Setting></pre> <p>See Updating OAM Configuration</p>
Retry Count	Retry Count	Not currently supported.
Email Address Attribute	Email Address Attribute	Not currently supported.
Challenge Question Attribute	Challenge Question Attribute	Not currently supported.
Challenge Answers Attribute	Challenge Answers Attribute	Not currently supported.

Figure 5-2 shows the Default and System Store designations. Notice the Access System Administrators section. You can add or remove Administrator roles only within the defined System Store and the store itself.

Figure 5-2 System Store Registration



**See Also:**

Details about classifying users in [Managing Policies to Protect Resources and Enable SSO](#)

5.2.5 Registering a New User Identity Store

Users with valid Oracle Access Management Administrator credentials can register a new user identity store using the Oracle Access Management 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 a specific identity store within Identity Conditions in Authorization Policies. Before you begin:

- Install the user identity store that you intend to register with Oracle Access Management.
- Extend the LDAP directory schema for Access Manager, as described in *Installing and Configuring Oracle Identity and Access Management*.
- Create Users and Groups in the LDAP directory, as described in your vendor documentation.

Follow this procedure to register a new identity store definition:

1. At the top of the Oracle Access Management Console, click **Configuration**.
2. In the Configuration console, click **User Identity Stores**.
3. In the OAM ID Stores section, click **Create**.
4. Fill in the form with appropriate values for your deployment ([Table 5-4](#)), then click **Apply** to submit the registration.
5. **Test Connection:** Click **Test Connection** to confirm connectivity, then close the Confirmation window.
6. Close the registration page.
7. **Add Administrators:** See "[Managing Administrator Roles](#)".
 - 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**.
8. **Set Default Store:** See "[About using the System Store for User Identities](#)".
9. Click **Apply** to submit the registration and close the page.
10. Configure one or more authentication modules or plug-ins to use this store, as described in:
 - "[Native LDAP Authentication Modules](#)"
 - "[Orchestrating Multi-Step Authentication with Plug-in Based Modules](#)"

5.2.6 Viewing or Editing a User Identity Store Registration

Users with valid Oracle Access Management Administrator credentials can view or modify the registration of a user identity store. The user identity store that you intend to register must be installed and running.

To view or modify a user identity store registration:

1. At the top of the Oracle Access Management Console, click **Configuration**.
2. In the Configuration console, click **User Identity Stores**.
3. In the OAM ID Stores list, select the target identity store and click **Edit**.
4. Modify values as needed (see [Table 5-4](#)).
5. Click **Apply** to update the registration (or close the tab without applying changes).
6. **Test Connection:** Click **Test Connection** button to confirm connectivity, then close the Confirmation window.
7. **Set as System or Default Store:** See "[About using the System Store for User Identities](#)".
8. **Manage Administrator Roles:** See "[Managing Administrator Roles](#)".
9. Configure one or more authentication modules or plug-ins to use this store, as described in:
 - "[Native LDAP Authentication Modules](#)"
 - "[Orchestrating Multi-Step Authentication with Plug-in Based Modules](#)"
10. Close the page when you finish.

5.2.7 Deleting a User Identity Store Registration

Users with valid Oracle Access Management Administrator credentials can use this procedure to delete a user identity store registration using the Oracle Access Management Console. You cannot delete the Default Store or System Store registration.

To delete a 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. At the top of the Oracle Access Management Console, click **Configuration**.
3. In the Configuration console, click **User Identity Stores**.
4. In the OAM ID Stores list, select the target identity store and click **Delete**.
5. In the confirmation dialog that appears, click **Delete** to confirm the deletion (or click Cancel to dismiss the window and retain the instance).
6. Confirm that the definition is no longer listed in the navigation tree.

5.3 Managing the Identity Directory Service User Identity Stores

Identity Directory Service (IDS) is a flexible and configurable service used by Access Manager as the means for accessing multiple identity data stores. The purpose of IDS is to allow the management of users or groups from identity stores not deployed with Access Manager itself.

The following sections contain the details.

- [Identity Directory Services](#)
- [Creating an Identity Directory Service Profile](#)
- [Editing or Deleting an Identity Directory Service Profile](#)
- [Creating a Form-fill Application Identity Directory Service Profile](#)
- [Understanding the Pre-Configured Identity Directory Service Profile](#)
- [Creating an Identity Directory Service Repository](#)

5.3.1 Identity Directory Services

Identity Directory Service offers a consistent and rationalized technology to access identity stores that eliminates redundant configurations and simplifies Identity Management operations.

IDS provides the following benefits:

1. Support for different types of user directories including integration with native user/password state managed by the directory.
2. Consistent administration user interface and a paradigm for working with different identity stores across Oracle Identity Management components.
3. Built in failover and load balancing capabilities.
4. Logical to physical attribute mapping and entity relationships.

The following list of directory servers are among those supported.

- Microsoft Active Directory
- Novell eDirectory
- Oracle Directory Server Enterprise Edition
- Oracle Internet Directory
- Oracle Unified Directory
- Oracle Virtual Directory
- OpenLDAP
- IBM Tivoli Directory Server
- WebLogic Server Embedded LDAP



Note:

Oracle recommends that you use the Identity Directory Service Profiles to access identity data stores rather than the legacy OAM ID Stores function as it will be deprecated in a future release.

Figure 5-3 is a screen capture of the Identity Directory Service console page.

Figure 5-3 Identity Directory Service Console Page

Configuration >

User Identity Stores

Default and System Store Apply

Default Store: UserIdentityStore1
System Store: msmad

OAM ID Stores Sync IDS Profiles

Manage local User Identity Stores. This includes IDS Profiles that are synchronized by using Sync IDS Profiles button.

View + Create ↺ Duplicate ✎ Edit ✕ Delete

Name	Directory Type	Host Information	Description	Synced IDS Profiles
UserIdentityStore1	EMBEDDED_LDAP	Idap-host7001		No
IDSPROFILE-idxuserrole				Yes
msmad	AD	domain2.bitzerqa1.com		No
IDSPROFILE-ESSOIDS				Yes
IDSPROFILE-idsprofile				Yes
IDSPROFILE-userrole				Yes

Identity Directory Service

Identity Directory Service is a common service used by Oracle Identity Management products to access and manage Identity Directory. The IDS Profiles can be used within Oracle Access Management after they are synchronized.

IDS Profiles Create Form-Fill Application IDS Profile

Manage common Identity Directory Service Profiles. IDS Profiles created here will appear in OAM ID Stores table. You have to synchronize IDS Profiles created outside Oracle Access Management Console using Sync IDS Profile button

View + Create ✎ Edit ✕ Delete

Name	Description	Repository Name	Created By
userrole	User/Role entities in Default Identity Directory		OPSS
idxuserrole	Fusion User/Role entities in Default Identity Direc...		OPSS
idsprofile	Identity profile for omss	idsprofile	OAM
ESSOIDS		ESSOIDSREP	OAM-Form-Fill

IDS Repositories

Manage Identity Directory Service Repositories that are common across Oracle Identity Management

View + Create ✎ Edit ✕ Delete

Name	Directory Type	Host Information
ESSOIDSREP	OU	slc07fyk.us.oracle.com:1389
idsprofile	ACTIVE_DIRECTORY	domain2.bitzerqa1.com:389

 **Note:**

Note this page contains the configuration panel for the legacy OAM ID Stores. Oracle recommends that you use the Identity Directory Service Profiles to access identity data stores rather than the legacy OAM ID Stores function as it will be deprecated in a future release.

Configuring an Identity Directory Service store involves configuring parameters for an IDS Profile and an IDS Repository. The IDS Profile specifies the full scope of traits for a particular type of identity store. It is the logical configuration for the repository and contains the following data.

- Entity definition
- Entity relationship definition
- Default operational configuration (including the tenant search/create base, the tenant filter, timeouts and cache configuration)

The IDS Repository configuration defines the actual location of the store. The IDS Repository is a physical configuration that containing the following data.

- Connection details (including the host machine, port number and credentials)Connection pool detailsHigh-availability/failover configurationEntity attribute mapping

5.3.2 Creating an Identity Directory Service Profile

You can create an Identity Directory Service profile form the **Configuration** console.

To create:

1. At the top of the Oracle Access Management Console, click **Configuration**.
2. In the Configuration console, click **User Identity Stores**.
3. In the IDS Profiles section, click **Create**.

The Create IDS Profile page is displayed as in [Figure 5-4](#).

Figure 5-4 Create IDS Profile Page

x
Create Identity Store Profile

* Name

Description

Repository
Repository Options
 Create New
 Use Existing
Test Connection

* Name

* Directory Type [select one]

* Hosts

View ▼ + Add ✕ Remove

Host Name	Port	Load Distribution (%)
<input style="width: 95%;" type="text"/>	3060 ▲ ▼	100 ▲ ▼

Availability Failover Load balanced

SSL Enabled

* Bind DN

* Bind Password

* Base DN

Password Management Enabled

Use Native ID Store Settings

Use Oblix User Schema

User
Base DN cn=users,dc=oracle.d
Login ID Attribute uid

RDN Attribute cn
Global Common ID Attribute uid

Object Classes

View ▼ + Add ✕ Remove

Object Class Name
<input style="width: 95%;" type="text" value="inetorgperson"/>

Group
Base DN cn=groups,dc=oracle,
ID Attribute cn

RDN Attribute cn

Object Classes

View ▼ + Add ✕ Remove

Object Class Name
<input style="width: 95%;" type="text" value="groupofuniquenames"/>

Create
Cancel

4. Provide the following values for the new Identity Directory Service profile.
 - **Name** - Type a unique name for this User Profile Service Provider.
 - **Description** - (Optional) Type a short description that will help you or another Administrator identify this service in the future.
5. Configure the Repository properties by selecting Create New or Use Existing.

Create New defines a new Repository object (that is, a reference to an LDAP directory server) for the Identity Directory Service connection. Click **Test Connection** after you have defined the values in the Repository section to verify they are correct. This option is only available when defining a new Identity Directory Service connection. **Use Existing** allows you to choose a previously defined Repository object by selecting it from the drop down menu.

- (Repository) **Name** - Enter a new unique name to create, or choose an existing one from the menu. After entering a new name, configure properties for the Identity Directory Service connection.
- **Directory Type** - Select the type of directory server software hosting the Repository; for example, *Microsoft Active Directory* or *Oracle Internet Directory*. If your directory is not listed, leave this field empty. If you are not defining a new Identity Directory Service connection or creating a new repository, this field is read-only.
- **Host Information** - Contains information about the host computer on which the Identity Directory Service Repository is located. Add multiple hosts if the directory server is part of a cluster. Click **Add** to add a new host to the table. In the **Host Name** column type either the IP Address or the name of the computer (or virtual computer) on which the Directory server is running. In the **Port** column, type the port number that the directory server is configured to use. If the hosts are part of a cluster, in the **Load Distribution** column type the load amount as a percentage that should be directed to each host. For multiple hosts, the amount should add up to 100%. To delete a host, select its row in the table and click **Remove**. If you are not defining a new Identity Directory Service connection or creating a new repository, this field is read-only.
- **Availability** - Choose **Failover** if the cluster is configured for failover operation, or choose **Load balanced** if the cluster distributes the load across multiple hosts. This field is read-only if you are using an existing repository.
- **SSL** - Select **Enabled** if the connection is configured for SSL. (See the *Securing Applications with Oracle Platform Security Services* for SSL configuration details.)

 **Note:**

Follow this procedure to add the SSL certificates required for setting the TLS connection.

- a. Create the LibOVD keystore by running this command.

```
MW_HOME/oracle_common/bin/libovdconfig.sh -host
WLS_ADMIN_HOST
-port WLS_ADMIN_PORT -userName weblogic
-domainPath WLS_DOMAIN_PATH -createKeystore
-contextName ids
```

Enter the AdminServer password and the password used for the LibOVD keystore when requested.

- b. Import the OID server certificate into the LibOVD keystore.

```
keytool -importcert
-keystore DOMAIN_HOME/config/fmwconfig/ovd/
ids/keystores/adapters.jks
-storepass KEYSTORE_PASSWORD -alias ALIAS_NAME
-file FULL_PATH_TO_CERTFILE -noprompt
```

- **Bind DN** - Type the distinguished name (DN) of the LDAP Administrator used to authenticate to the Directory server.
 - **Bind Password** - Type the Bind DN password used to authenticate to the Directory server.
 - **Base DN** - Type the base distinguished name (DN) where User and Group data is located.
 - **Password Management** - selecting **Enable Password Management** enables password policy enforcement against the attribute values listed in [Table 5-4](#). The corresponding options in the password policy must be configured as well.
6. Configure the User properties to configure the LDAP User object.

 **Note:**

These fields are read-only if using an existing Identity Directory Service connection.

- **Object Classes** - Click **Add** to add a custom object class that represents people in an organization as defined on your directory server.
- **RDN Attribute** - Type the relative distinguished name attribute (for example, *cn*) designated for the User object on the directory server.
- **Base DN** - Type the base DN (in LDAP form) for the User object on the directory server.
- **Login ID Attribute** - Type the LDAP attribute from which the login ID specifying the User will be extracted.
- **Global Common ID Attribute** - Type the global common user ID attribute.

7. Configure the Group properties to configure the LDAP group object.
 - **Object Classes** - Click **Add** to add a custom object class that represents a group of people in an organization as defined on your Directory server.
 - **RDN Attribute** - Type the relative distinguished name attribute (for example, *cn*) designated for the Group object on the directory server.
 - **Base DN** - Type the base DN (in LDAP form) for the Group object on the directory server.
 - **ID Attribute** - Type the LDAP attribute from which the ID designated for the Group object will be extracted.

8. Click **Create**.

The profile is displayed in the IDS Profiles table.

5.3.3 Editing or Deleting an Identity Directory Service Profile

To edit or delete an IDS Profile, select the name in the table and click **Edit** or **Delete** in the tool bar.

Editing the profile allows for additional configuration properties for the Identity Directory Service connection.

- **Name** - Choose an Identity Directory Service connection to associate with the User Profile Service Provider from the drop down menu.
 - If you choose either of the default Identity Directory Services (either `userrole` or `idxuserrole`) you cannot view or edit the configuration values.
 - If you choose an Identity Directory Service connection that you or another Administrator created, you can view and edit the configuration values as needed.
- **General and Repository** - Use the fields under this tab to edit the Directory Service and Repository configuration values.
 - **Repository Name** - Choose from the menu a repository to associate with the Identity Directory Service connection. After choosing a repository, configure its properties using the following form fields.
 - **Directory Type** - Displays the type of Directory server software hosting the Repository, for example *Microsoft Active Directory*, *Oracle Internet Directory*, and so on. This field is read-only.
 - **Host Information** - Displays information about the host computer where the Identity Directory Service Repository is located. Add multiple hosts if the Directory server is part of a cluster. Click **Add** to add a new host to the table. In the **Host Name** column type either the IP Address or the name of the computer (or virtual computer) that the Directory server is running on. In the **Port** column, type the port number that the Directory server is configured to use. If the hosts are part of a cluster, in the **Load Distribution** column type the load amount as a percentage that should be directed to each host. For multiple hosts, the amount should add up to 100%. To delete a host, select its row in the table and click **Remove**. If you are not defining a new Identity Directory Service connection or creating a new repository, this field is read-only.
 - **Availability** - Choose **Failover** if the cluster is configured for failover operation, or choose **Load balanced** if the cluster distributes the load across multiple hosts. This field is read-only if you are using an existing repository.

- **SSL** - Select **Enabled** if the connection is configured for SSL. Otherwise clear the option box. See **SSL** in [Creating an Identity Directory Service Profile](#) for information on how to add the SSL certificates required for the TLS connection.
- **Bind DN** - Type the distinguished name (DN) of the LDAP Administrator used to authenticate to the Directory server.
- **Bind Password** - Type the Bind DN password used to authenticate to the Directory server.
- **Base DN** - Type the base distinguished name (DN) where User and Group data is located.
- **Password Management** - selecting **Enable Password Management** enables password policy enforcement against the attribute values listed in [Table 5-4](#). The corresponding options in the password policy must be configured as well.
- **Entity Attributes** - Use the fields under this tab to view or edit the attributes. Click **Add** to add an attribute to the table or click **Remove** to delete an attribute.
 - **Name** - The attribute name.
 - **Physical Attribute** - The name of the corresponding physical attribute type in the underlying Repository.
 - **Type** - The attribute's data type.
 - **Description** - A brief description of the attribute.
 - **Sensitive** - Select to mark that the attribute contains sensitive information such as a password.
 - **Read-only** - Select to protect the attribute from modification.
- **Entities / User Properties** - Use the fields under the User sub head to configure interaction with the User entities on the LDAP server.
 - **Create Base** - Specifies the base DN (the top level of the LDAP directory tree) at which Users are defined.
 - **Search Base** - Specifies the search base DN for Users. Only entries at or below the search base DN are considered when processing the search operation.
 - **Create Object Classes** - Specifies the object class under which attributes associated with a person are stored.
 - **RDN Attribute** - Specifies the relative distinguished name attribute, for example *cn*.
 - **ID Attribute** - Specifies the attribute that uniquely identifies the User, such as the *uid* attribute or the *loginid* attribute.
 - **Filter Object Classes** - Specifies the object class by which to filter.
 - **Attributes Configuration** - Specify the User attributes that should be available to, and searchable by, the User Profile Service Provider.
 - * **Used** - Specifies if the attribute is used for Users in the directory service.
 - * **Attribute Name** - Specifies the name of the attribute as defined on the **Entity Attributes** tab.
 - * **In Results** - Select if the specified attribute should be returned in search results.

- * **Searchable** - Select if the specified attribute should be available for search operations.
- * **Search Operator** - Select a search operator from the menu to restrict how the specified attribute is searched.
- **Operations Configuration** - Select from **Create**, **Update**, **Delete**, and **Search** to enable those operations at the User entity level. Clear the option boxes to disable them.
- **Entities / Group Properties** - Use the fields under the Group sub head to configure interaction with the Group entities on the LDAP server.
 - **Create Base** - Specifies the base DN (the top level of the LDAP directory tree) at which Users are defined.
 - **Search Base** - Specifies the search base DN for Groups. Only entries at or below the search base DN are considered when processing the search operation.
 - **Create Object Classes** - Specifies the object class under which attributes associated with a Group are stored.
 - **RDN Attribute** - Specifies the relative distinguished name attribute; for example, *cn*.
 - **ID Attribute** - Specifies the LDAP attribute that uniquely identifies the Group.
 - **Filter Object Classes** - Specifies the object class by which to filter.
 - **Attributes Configuration** - Specify the Group attributes that should be available to, and searchable by, the User Profile Service Provider.
 - * **Used** - Specifies if the attribute is used for Users in the directory service.
 - * **Attribute Name** - Specifies the name of the attribute as defined on the **Entity Attributes** tab.
 - * **In Results** - Select if the specified attribute should be returned in search results.
 - * **Searchable** - Select if the specified attribute should be available for search operations.
 - * **Search Operator** - Select a search operator from the menu to restrict how the specified attribute is searched.
 - **Operations Configuration** - Select from **Create**, **Update**, **Delete**, and **Search** to enable those operations at the Group entity level. Clear the option boxes to disable them.
- **Relationships** - Use the fields under this tab to configure the relationship between attributes for this Identity Directory Service.
 - **Name** - The relationship name.
 - **(From) Entity** - Choose **User** to select from User attributes or choose **Group** to select from Group attributes in the **(From) Attribute** column.
 - **(From) Attribute** - Choose the attribute from which you are mapping.
 - **Relation** - Choose the menu option that describes the relationship between the specified attribute in the **From** column and the specified attribute in the **To** column.
 - **(To) Entity** - Choose **User** to select from User attributes or choose **Group** to select from Group attributes in the **(To) Attribute** column.
 - **(To) Attribute** - Choose the attribute to which you are mapping.

- **Recursive** - Select if the relationship extends down the directory tree to include nested child entities or up the directory tree to include parent entities.
- **Relationship Configuration** - Type the URI segment used to access the corresponding column in the Identity Directory Service. Use **Add** to add a new relationship or **Remove** to remove a configured relationship.
 - **Access URI** - Type a URI segment that will be used to access a corresponding data column in the Identity Directory service. For example, if `memberOf` is the Access URI, then:


```
http://host:port/.../idX/memberOf
```

would be the URI to access related entities of an entity with ID `idX`.
 - **Identity Directory Service Relation** - Choose the Directory Service relationship that is to be accessed by the **Access URI** segment. You can configure relationships on the **Relationships** tab in the **Identity Directory Service** configuration section provided that the Identity Directory Service is *not* the pre-configured `UserProfile` Identity Provider. (You cannot configure Identity Directory Service relationships for the `UserProfile` Service Provider.)
 - **Entity URI Attribute** - Type the JSON attribute name to be used in the URI response. For example, if `person-uri` is the specified entity URI attribute, the URI response would be as follows:


```
{ {"person-uri":uriY1, ...}, {"person-uri":uriY2, ...}, ... }
```

where `uriY1` and `uriY2` are the direct URIs to access each of the related entities.
 - **Scope for Requesting Recursion** - Use Scope attribute values with the scope query parameter to retrieve a nested level of attributes in a relationship search. To access related entities recursively, type the value to be used.

If the **Scope for Requesting Recursion** value is the attribute value `all`, then the following REST URI example is used to make the request:

```
http://host:port/.../idX/reports?scope=all
```

In this example, the URI returns the entities related to the entity with ID `idX`, as well as all further related entities.

5.3.4 Creating a Form-fill Application Identity Directory Service Profile

To create an Identity Directory Service Profile for a Form-fill Application, click the Create Form-fill Application IDS Profile button on the left of the User Identity Stores console page.

(See [Figure 5-3](#).)

[Creating an Identity Directory Service Profile](#) and [Editing or Deleting an Identity Directory Service Profile](#) contain definitions for most of the Form-fill attributes. Additional definitions for the Entity Search Bases section specific to this type of profile are listed below.

- User Search Base - Full DN for the node at which enterprise users are stored in the directory; for example, `cn=Users,realm_DN`.
- App Template Search Base - Full DN for the node from which searches for the Application Templates will begin.

- Top Search Base - Full DN for the node from which searches will begin; for example, cn=realm_DN.

5.3.5 Understanding the Pre-Configured Identity Directory Service Profile

Mobile and Social provides a pre-configured IDS Profile named UserIdentityStore1. The Pre-Configured Identity Directory Service Profile allows lookup and update tasks to be performed on directory objects using Mobile and Social.

5.3.6 Creating an Identity Directory Service Repository

You can create an Identity Directory Service repository from the Configuration console.

To create an Identity Directory Service repository:

1. At the top of the Oracle Access Management console, click **Configuration**.
2. In the Configuration console, click **User Identity Stores**.
3. Click **Create** under IDS Repository.

The Create IDS Repository page is displayed as in [Figure 5-5](#).

Figure 5-5 Create IDS Repository Page

Create LDAP Repository [Close]

[Test Connection]

* Name

* Directory Type [select one] ▼

* Hosts View ▼ + Add ✕ Remove

Host Name	Port	Load Distribution
<input type="text"/>	3060 ▲ ▼	100 ▲ ▼

Availability Failover Load balanced

SSL Enabled

* Bind DN

* Bind Password

* Base DN

Password Management Enabled

Use Native ID Store Settings

Use Oblix User Schema

[Create] [Cancel]

4. Provide the following values for the new Identity Directory Service repository.
 - a. Name: the entry must be a unique.
 - b. Select the Directory Type from the drop down choices.
5. Click Add to configure the physical location of the repository (Host name, Port number and Load Weightage percentage).
6. Configure the Repository properties as follows:
 - a. (Repository) **Name** - Enter a new unique name to create, or choose an existing one from the menu. After entering a new name, configure properties for the Identity Directory Service connection.
 - b. **Directory Type** - Select the type of directory server software hosting the Repository; for example, *Microsoft Active Directory* or *Oracle Internet Directory*. If your directory is not listed, leave this field empty. If you are not defining a new Identity Directory Service connection or creating a new repository, this field is read-only.

- c. **Host Information** - Contains information about the host computer on which the Identity Directory Service Repository is located. Add multiple hosts if the directory server is part of a cluster. Click **Add** to add a new host to the table. In the **Host Name** column type either the IP Address or the name of the computer (or virtual computer) on which the Directory server is running. In the **Port** column, type the port number that the directory server is configured to use. If the hosts are part of a cluster, in the **Load Distribution** column type the load amount as a percentage that should be directed to each host. For multiple hosts, the amount should add up to 100%. To delete a host, select its row in the table and click **Remove**. If you are not defining a new Identity Directory Service connection or creating a new repository, this field is read-only.
 - d. **Availability** - Choose **Failover** if the cluster is configured for failover operation, or choose **Load balanced** if the cluster distributes the load across multiple hosts. This field is read-only if you are using an existing repository.
 - e. **SSL** - Select **Enabled** if the connection is configured for SSL. See **SSL** in [Creating an Identity Directory Service Profile](#) for information on how to add the SSL certificates required for the TLS connection. (See the *Securing Applications with Oracle Platform Security Services* for SSL configuration details.)
 - f. **Bind DN** - Type the distinguished name (DN) of the LDAP Administrator used to authenticate to the Directory server.
 - g. **Bind Password** - Type the Bind DN password used to authenticate to the Directory server.
 - h. **Base DN** - Type the base distinguished name (DN) where User and Group data is located.
 - i. **Password Management** - selecting **Enable Password Management** enables password policy enforcement against the attribute values listed in [Table 5-4](#). The corresponding options in the password policy must be configured as well.
7. Click **Test Connection** to confirm the values are correct.
 8. Click **Create**.

The repository is displayed in the IDS Repositories table.

5.4 Managing Administrator Roles

By default, the Oracle Access Management Administrators role is the same as the WebLogic Administrators role (Administrators).

You can register another User Identity Store (Oracle Internet Directory, for example); however, user `weblogic` must be defined with at least one user in the registered store to authenticate against. Administrator login works only when the Authentication Scheme (and assigned Authentication Module), also uses the System Store. This section provides the following topics:

- [Understanding Administrator Roles](#)
- [Defining and Removing Administrator Roles](#)

5.4.1 Understanding Administrator Roles

Your enterprise might require independent sets of Administrators: one set of users responsible for Access Manager and another for 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 appears. You can add new Administrator roles when adding or editing a User Identity Store registration. [Figure 5-6](#) shows the page and controls to use.

Figure 5-6 Add System Administrator Roles

Add System Administrator Roles [X]

Search and select the users and/ or groups to add as System Administrators.

Search

Name

Type

[Search] [Reset]

View ▾ [Detach]

Name	Type
No data to display.	

[Add selected] [Cancel]

5.4.2 Defining and Removing Administrator Roles

Oracle Access Management Administrator roles which must be stored in the User Identity Store designated as the System Store can be defined or removed.

First, define the desired LDAP group to use for Administrators and then ensure that your Administrators group is available in the group search base. (See [About using the System Store for User Identities.](#)) To add or remove an Administrator role from the System Store, follow this procedure.

- 1. View 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.** At the top of the Oracle Access Management Console, click **Configuration**.
 - b.** In the Configuration console, click **Administration**.

The registered System Store can not be changed from this page.

- c. Search the System Store to find configured administrators.
2. **Add User Roles:**
 - a. Click the Grant (+) button above the Access System Administrators table to display the Add Users and Groups dialog box.
 - b. Select **User** in the Type list and click **Search**.
 - c. In the results list, click the desired user, then click **Add Selected**.
 - d. Repeat as need to add desired Administrator User roles.
 - e. Click **Apply** to submit user roles.
3. **Add Group Roles:**
 - a. Click the **Grant** (+) button above the Access System Administrators table to display the Add Users and Groups 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.
4. **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.
5. Correct any authentication plug-ins that use the System Store (if this is a new store).
This procedure is described in "[Orchestrating Multi-Step Authentication with Plug-in Based Modules](#) "
6. **Test the New Role:** Close the browser window, then re-open it.
 - a. Sign out of the Oracle Access Management Console and close the browser window.
 - b. Start up the Oracle Access Management 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.
Login Failure: See "[Administrator Lockout](#)".

5.5 Managing the Policy and Session Database

Oracle Access Management requires a database to store Access Manager policy data, password management data, and Access Manager sessions in a production environment. At most, your deployment can have one policy store database (which serves password management) and one session store. By default, a single JDBC data source is used for both.

This section includes the following topics:

- [About the Database Store for Policy, Password Management, and Sessions](#)
- [About Database Deployment](#)

- [Configuring a Separate Database for Access Manager Sessions](#)

5.5.1 About the Database Store for Policy, Password Management, and Sessions

In a production environment, the Oracle database stores policy data, password management data, and sessions of the Access Manager.

The policy store database, by default, stores the following data:

- Policy data, including authentication modules and schemes, Application Domains, and policies.
- Password Management data, which includes password policy type for each configured User Identity store as well as the policy that governs password requirements, expiry, notification,
- Sessions, as a persistent backup to distributed in-memory storage

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 and Session Database "](#)

5.5.2 About Database Deployment

Oracle requires a single database as the policy store in production environments. This single database 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).

Note:

You can have up to two databases: one policy database and one session database. Access Manager is agnostic with respect to the actual back end repository and does not manage this policy store configuration directly.

The policy database must be installed according to vendor instructions. The policy database is configured for use in a Oracle WebLogic Server domain using Oracle Fusion Middleware Configuration Wizard and policy store Database configuration template.

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

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

An Administrator must extend the database with the Access Manager-specific schema using RCU, as described in Navigating the Repository Creation Utility Screens to Create the Schemas in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*. Basic schema creation occurs when the RCU is invoked. The RCU prepares the database to accept Access Manager policy, password management, and session data.

Using the WebLogic Configuration Wizard you can register and test the connection to the database.

Actual Access Manager policy elements are created the first time the WebLogic AdminServer is started with the Oracle Access Management Console deployed.



See Also:

Configuring the Oracle OAM Suite Domain in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

5.5.3 Configuring a Separate Database for Access Manager Sessions

Access Manager includes a data source named `oamDS` which is configured against the database instance extended with the Access Manager 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 Management Console.



Note:

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

1. Install and configure the database for session data and then use RCU with the Access Manager-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:

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>
```

See [Updating OAM Configuration](#)

3. Restart AdminServer and OAM Servers.

5.6 Introduction to Oracle Access Management Keystores

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.

This section provides the following topics:

- [Access Manager Security Keys and the Embedded Java Keystore](#)
- [Access Manager Keystores](#)
- [Identity Federation Keystore](#)

5.6.1 Access Manager Security Keys and the Embedded Java Keystore

Keystores are created and configured during Access Manager installation. The password and the key entries password were randomly generated.

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

- Every OAM Agent has a secret key that other agents cannot read.

- During agent and application registration, a key is generated for encrypting and decrypting SSO Cookies (for Webgates).

Administrators use the Oracle-provided `importcert` tool for several different procedures related to keystores, keys, and certificates, as described in [Securing Communication](#).

[Table 5-5](#) identifies the generated Access Manager cryptographic keys.

Table 5-5 Access Manager Keys and Storage

Keys and Storage	Description
Access Manager Cryptographic keys	<ul style="list-style-type: none"> • One per agent secret key shared between OAM Webgate and OAM Server • One OAM Server key
Key storage	<ul style="list-style-type: none"> • Agent side: A per-agent key is stored locally in the Oracle Secret Store in a wallet file. Client keystore/scratch/clientTrustStore.jks and /scratch/clientKey.jks can be used. • OAM Server side: .oamkeystore contains a per-agent key, and server key, are stored in the credential store on the server side.

Keystores are not accessible using the Oracle Access Management Console. You can manage keystores and certificates as described in [Securing Communication](#).



See Also:

"Identity Federation Keystore"

- "About Communication Between OAM Servers and WebGates"
- Secure Communication in *Administering Oracle Fusion Middleware* for details about the SSL automation tool and managing ports for WebLogic Server, Oracle HTTP Server, and Oracle Fusion Middleware

5.6.2 Access Manager Keystores

Keystores for Access Manager and Security Token Service are created and configured during the installation of the Access Manger.

[Table 5-6](#) provides a summary of keystores used for Access Manager.

Table 5-6 Keystores for Access Manager and Security Token Service

Keystore	Description
System Keystore / Partner Keystore	The container for keys and certificates associated with OAM Server instances (OAM secret keys for signing and encryption).
.oamkeystore	The container 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.
	Only one System Keystore of type JCEKS can be present: .oamkeystore. \$DOMAIN_HOME/config/fmwconfig/.oamkeystore
	The certificate alias and password can be configured using the Oracle Access Management Console.

Table 5-6 (Cont.) Keystores for Access Manager and Security Token Service

Keystore	Description
Trust Keystore	The Trust Keystore is used to validate keys and certificates presented by clients to establish trust in entities interacting with OAM Server instances.
amtruststore	<p><code>\$DOMAIN_HOME/config/fmwconfig/amtruststore</code></p> <p>amtruststore is created during installation, and must include at least one trusted anchor.</p> <p>The Trust Keystore is managed by using the JRE's keytool application. Security Token Service can use a custom trust keystore.</p>
Certificate Revocation Lists (CRL)	Certificate revocation information lists are stored in a ZIP archive on the filesystem. These are used by OAM Servers when performing CRL-based certificate revocation checking.
amcrl.jar	<p>amcrl.jar contains CRL files in the DER format:</p> <p><code>\$DOMAIN_HOME/config/fmwconfig/amcrl.jar</code></p> <p>The OAM Server defines a notification listener for the Keystores and the CRL Zip file. Any changes to these files causes Security Token Service to reload the keystore/crl-zip at runtime, without requiring any restarts.</p> <p>amcrl.jar is created by installation and can be modified using the Oracle Access Management Console.</p> <p>See Also:</p> <ul style="list-style-type: none"> • "Certificate Validation and Revocation"
Oracle WSM Agent Keystore	The Oracle WSM Agent uses this keystore for various cryptographic operations. For these operations, the Oracle WSM Agent uses the keystore configured for Oracle WSM tasks.
default-keystore.jks	Oracle strongly recommends that the Oracle WSM Agent keystore and the Access Manager and Security Token Service keystore always be different. Otherwise, keys could be available to any modules authorized by OPSS to access the keystore and Access Manager/Security Token Service keys might be accessed.
OPSS Keystore	<p>For special cases where clients use referencing schemes such as SKI (as opposed to a certificate token being received as part of the web service request), the requester's certificates need to be populated in the OPSS Keystore.</p> <p>This is an uncommon scenario that requires manually provisioning keys to the OPSS keystore.</p> <p>See Also:</p> <ul style="list-style-type: none"> • Managing Keys and Certificates in <i>Securing Applications with Oracle Platform Security Services</i>.

5.6.3 Identity Federation Keystore

Identity Federation and Access Manager store key pairs and certificates that are used for digital signatures and encryption operations.

Identity Federation uses keys to:

- Sign outgoing assertions
- Decrypt incoming XML encrypted data contained inside the SAML message

The following keystore is used to store the encryption and signing certificates:

`$DOMAIN_HOME/config/fmwconfig/.oamkeystore`

Identity Federation uses CSF to securely store keystore passwords, as well as server credentials such as HTTP Basic Authentication usernames and passwords.

 **See Also:**

- ["About Communication Between OAM Servers and WebGates"](#)
- ["Defining Keystore Settings for Federation"](#)

5.7 Integrating a Supported LDAP Directory with Oracle Access Manager

A centralized LDAP store can be enabled for use with Oracle Access Manager post-installation.

Oracle Internet Directory is featured in this discussion however the tasks are the same regardless of your chosen LDAP directory.

Oracle Access Manager addresses each user population and LDAP directory store as an identity domain. Each identity domain maps to a configured LDAP User Identity Store that is registered with Oracle Access Manager. Multiple LDAP stores can be used with each one relying on a different supported LDAP provider.

During initial WebLogic Server domain configuration, the Embedded LDAP is configured as the one and only User Identity Store for Oracle Access Manager. Within the Embedded LDAP, the Administrators group is created, with `weblogic` seeded as the default Administrator:

- Only the User Identity Store designated as the System Store is used to authenticate Administrators signing in to use the Oracle Access Management 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 designated as the Default User Identity Store.
- 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.

After registering a User Identity Store with Access Manager, administrators can reference the store in one or more authentication modules, which form the basis for Oracle Access Manager Authentication Schemes and Policies. When you register a partner (either using the Oracle Access Management Console or the remote registration tool), an application domain can be created and seeded with a policy that uses the designated default Authentication Scheme. When a user attempts to access an Oracle Access Manager-protected resource, she is authenticated against the store designated by the authentication module. For more information, see Introduction to IdM Suite Components Integration in *Integration Guide for Oracle Identity Management Suite*.

6

Managing Server Registration

Managed server instances must be registered in order to interact with Oracle Access Management. (In this book, these managed servers are referred to as OAM Servers.) Accomplish this registration task using the Oracle Access Management Console. You need to familiarize yourself with the following topics to manage server registrations:

- [Before You Register](#)
- [Understanding OAM Server Registration and Management](#)
- [Managing Individual OAM Server Registrations](#)

6.1 Before You Register

Ensure that the environment meets the requirements before you register.

The following environmental considerations should be 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.
See [Oracle Fusion Middleware Administrator's Guide](#).

Oracle recommends that you review the following topic:

See [Understanding OAM Server Registration and Management](#).

6.2 Understanding OAM Server Registration and Management

The Oracle Access Management 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.

The Oracle Access Management Console might be referred to as the OAM Administration Server. However, this is not a peer of the OAM Server deployed on a WebLogic Managed Server.

The Oracle Access Management runtime instance deployed on Oracle WebLogic Managed Servers is referred to as an OAM Server. Each OAM Server must be registered with Access Manager to enable communication with registered agents during authentication, authorization, and resource access.

Administrators can extend the WebLogic Server domain and add more OAM Server instances whenever needed, using either:

- The WebLogic Server Administration Console, after which you manually register the OAM Server instance using the Oracle Access Management Console
- The WebLogic Configuration Wizard

- Customized Oracle WebLogic Scripting Tool (WLST) commands. See Customization Commands in *WLST Command Reference for WebLogic Server*

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

 **See Also:**

Installing and Configuring Oracle Identity and Access Management.

This section introduces OAM Server instance registration and management using the Oracle Access Management Console:

- [About Individual OAM Server Registrations](#)
- [About Communication Between OAM Servers and WebGates](#)

 **See Also:**

Features Not Supported in Access Manager.

6.2.1 About Individual OAM Server Registrations

Administrators can add one or more Managed Servers to the WebLogic Server domain for Oracle Access Management.

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.

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 Management Console and to every OAM Server in the cluster.

 **See Also:**

Customization Commands in *WLST Command Reference for WebLogic Server*

Only OAM Servers are registered with Oracle Access Management. The Oracle Access Management Console (on the WebLogic Administration Server) is not registered with itself.

Regardless of the method used to register an OAM Server, details for each instance are located on the System Configuration tab, Common Configuration section in the Oracle Access Management Console, including:

- Server name, Host, Port

Administrators can search for a specific instance registration, register a newly installed OAM Server, view, modify, or delete server registrations using the Oracle Access Management Console. For more information, see "[OAM Server Registration Page](#)".

6.2.2 About Communication Between OAM Servers and WebGates

The OAM Server communication mode can be changed after a successful agent registration. The Webgate mode needs to be at the same level as the OAM Server mode or higher for the server to continue communicating with the agent.

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.
- **HTTP:** This mode is auto configured in WebGate user defined parameters (based on the settings in WebGate Load Balancer of Access Manager Settings page). Use this unencrypted communication mode if the parameter `OAMServerCommunicationMode` is set to HTTP. It is a user defined configuration parameter.
- **HTTPS:** This mode is auto configured in WebGate user defined parameters (based on the settings in WebGate Load Balancer of Access Manager Settings page). Use this encrypted communication mode if the parameter `OAMServerCommunicationMode` is set to HTTPS. It is a user defined configuration parameter.

On each individual OAM Server registration, the security mode is defined on the Proxy tab, as described in "[OAM Server Registration Page](#)".

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](#)".
- Appropriately signed X.509 digital certificates, as described in [Securing Communication](#).

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, if 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:

[Securing Communication](#)

6.2.3 Conditions Requiring Server Restart

Most Oracle Access Management functional services take up changes made through the Oracle Access Management Console without restarting OAM Server.

[Table 6-1](#) identifies conditions that do require a server restart.

Table 6-1 Conditions Requiring Server Restart

Event	Description
Load balancer server definition	A change requires an OAM Server restart.
Managed Server port number	A change requires an OAM Server restart.
New Managed Server	Adding a new managed server to the cluster requires restarting the AdminServer to policy enable uptake.

6.3 Managing Individual OAM Server Registrations

OAM Server instances can be registered and managed using the Oracle Access Management Console.

Topics here include:

- [OAM Server Registration Page](#)
- [Registering a Fresh OAM Server Instance](#)
- [Viewing or Editing Individual OAM Server Registrations and Proxy Settings](#)
- [Deleting an Individual Server Registration](#)

6.3.1 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 Management Console.

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

See Also:

Access Manager WLST Commands in *WebLogic Scripting Tool Command Reference for Identity and Access Management*

[Figure 6-1](#) illustrates a typical OAM Server registration page when viewed within the Oracle Access Management Console. To access the OAM Server registration page using the Oracle Access Management Console, click Configuration in the top right of the console and then click the Server Instances link on the Configuration page. From

the resulting Server Instances search page, click Create in the Search Results table to display the Create: OAM Server page. See [Registering a Fresh OAM Server Instance](#) for details on how to configure this page.

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

The screenshot shows a web browser window with two tabs: 'Launch Pad' and 'Server Instances'. The 'Server Instances' tab is active, displaying a configuration page titled 'Create: OAM Server OAM Server Instance'. The page has a breadcrumb 'Configuration >' and an 'Apply' button in the top right. The form contains several required fields:

- * Server Name: A text input field.
- * Host: A text input field.
- * Port: A spinner control with up and down arrows.
- OAM Proxy**: A section header with a small triangle icon.
- * Proxy Server Id: A text input field.
- * Port: A spinner control with the value '0' and up/down arrows.
- * Mode: A dropdown menu.

 At the bottom of the page, there is a copyright notice: 'Copyright © 2000, 2017, Oracle and/or its affiliates. All rights reserved.'

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: <i>host2.domain.com</i> .
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 of the authentication scheme must be set to the absolute URL which points to <code>http</code> as the protocol and non-SSL port. See Also: Securing Communication



See Also:

["Managing Individual OAM Server Registrations"](#)

6.3.1.1 OAM Proxy Settings

An integrated proxy server (OAM Proxy) is installed with each Managed Server for OAM Server.

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 access clients to interact with Access Manager.



Note:

For Access Clients, Access Manager provides authentication and authorization functionality only. Policy modification through Access Clients is not supported.

OAM Proxy settings are documented in [Table 6-3](#).

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

OAM Proxy Setting	Value
Port	The unique port on which this OAM Proxy instance is listening. On a default installation, the port is 5575.
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. On a default installation, the Proxy Server ID is AccessServerConfigProxy.
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): <ul style="list-style-type: none"> • 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. On a default installation, the Mode is Open. 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 ". See Also: Securing Communication if you are configuring Simple or Cert transport security modes.

OAM Proxy Logging: Oracle Access Management services use the same logging infrastructure as any other Oracle Fusion Middleware component, as described in [Auditing Administrative and Run-time Events](#). However, OAM Proxy uses Apache log4j for logging.

6.3.2 Registering a Fresh OAM Server Instance

Users with valid Administrator credentials can register a new Managed Server (OAM Server) instance using the Oracle Access Management Console. Each OAM Server must be registered to communicate with agents.

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

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 Management Console and click **Configuration** in the top bar.
3. In the Configuration console, click **Server Instances**.
4. In the tab that appears, click **Create OAM Server**.

The OAM Server registration page illustrated in [Figure 6-1](#) is displayed.

5. On the Create: OAM Server page, enter details for your instance, as described in [Table 6-2](#):
 - Server name
 - Host
 - Port
6. Proxy: Enter or select details for this OAM Proxy instance.
 - Port
 - Proxy Server ID
 - Mode (Open, Simple, or Cert)



See Also:

[Securing Communication](#) if you are using Simple or Cert mode

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.



See Also:

- *Installing and Configuring Oracle Identity and Access Management*
- ["OAM Server Registration Page "](#)

6.3.3 Viewing or Editing Individual OAM Server Registrations and Proxy Settings

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

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

See Also:

- "OAM Server Registration Page "
- Access Manager WLST Commands in *WebLogic Scripting Tool Command Reference for Identity and Access Management*
- Movement Scripts in *Administering Oracle Fusion Middleware*

1. At the top of the Oracle Access Management Console, click **Configuration**.
2. In the Configuration console, click **Server Instances**.
3. In the page that appears, click **Search**, then double-click the target instance 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.
4. On the OAM Server page, change details for your instance, as described in [Table 6-2](#).
5. **Proxy**: Change details for this OAM Proxy instance, as described in [Table 6-3](#).

See Also:

[Securing Communication](#) if you are using Simple or Cert mode

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 delete an OAM server registration, effectively disabling it.

To delete:

1. At the top of the Oracle Access Management console, click Configuration.
2. In the Configuration console, click Server Instances.

3. In the tab that appears, double-click the target instance to confirm its details, then close the tab.
4. In the list of instances, select the target instance, click **Delete** in the tool bar, and confirm removal in the dialog that appears.
5. Confirm that the instance has been removed from the instance list.
6. Remove the deleted instance from the WebLogic Server Administration Console.
The Node Manager on Managed Server host handles the rest automatically.

Part III

Logging, Auditing, Reporting and Monitoring Performance

Administrators can log component and WebGate event messages, audit administrative and run-time events, and performance monitoring for Oracle Access Management services.

This section contains the following chapters:

- [Logging Component Event Messages](#)
- [Auditing Administrative and Run-time Events](#)
- [Logging WebGate Event Messages](#)
- [Understanding Oracle Access Management Reports](#)
- [Monitoring Oracle Access Management Performance and Access Manager Health](#)
- [Monitoring Performance and Logs with Fusion Middleware Control](#)

7

Logging Component Event Messages

Logging is the mechanism by which components and services write messages to a log file in order to capture critical component events, processes, and state information. Administrators can configure logging to provide information at various levels of granularity using the same logging infrastructure and guidelines as any other component in Oracle Fusion Middleware 12c: `java.util.logging` (standard and available in all Java environments).

Configuring logging and locating log files are the focus of this chapter which contains the following sections.

- [About Oracle Access Management Logging](#)
- [Logging Component Event Messages](#)
- [Configuring Logging for Access Manager](#)
- [Configuring Logging for Identity Federation](#)
- [Validating Run-time Event Logging Configuration](#)

7.1 About Oracle Access Management Logging

The logging system writes output to flat files only. Logging to an Oracle Database instance is not supported. Unless explicitly stated, information in this chapter is the same whether using any of the services in Oracle Access Management.

Additionally:

- You can use a custom Oracle WebLogic Scripting Tool (WLST) command to change logging levels.
- Diagnosing problems using the information in log files is outside the scope of this chapter.
- Before you can perform tasks in this chapter ensure that the Oracle Access Management Console and a managed OAM Server are running.

Oracle also recommends that you review [Managing Server Registration](#) .

7.2 Logging Component Event Messages

The logging infrastructure records messages that can be used for problem diagnosis. Security Token Service is a J2EE Web application, part of the Access Manager J2EE Application. Both use OJDL for logging purposes. 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.

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 Management makes use of the files in [Table 7-1](#).

Table 7-1 Logging Files

File Type	Description
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 Note: By default, Security Token Service and Identity Federation messages are logged in the OAM Server's log file. However, for convenience, you can edit logging.xml to direct Security Token Service or Identity Federation information to a separate log file, as described in " Configuring Logging for Identity Federation ".
Log File	Logged information is stored in the following location: \$DOMAIN_HOME/servers/SERVER-NAME/logs/ SERVER-NAME-diagnostics.log

Oracle Access Management uses the WebLogic container's logging defaults in [Table 7-2](#).

Table 7-2 Logging Defaults

Log Type	Description
Events	The following events are logged automatically: <ul style="list-style-type: none"> OAM Server events (managed run-time servers) Administrative events (generated for configuration changes made using the console)
Levels	By default, the log level for all Oracle Access Management 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:

- [Component Loggers for Security Token Service and Access Manager](#)
- [Sample Logger and Log Handler Definition](#)
- [About Logging Levels](#)



See Also:

- [Monitoring Performance and Logs with Fusion Middleware Control](#) for details about how you can configure and view logs using Fusion Middleware Control
- For logging information, see Logging Audit Events Programmatically in the *Securing Applications with Oracle Platform Security Services*.

7.2.1 Component Loggers for Security Token Service and Access Manager

The component loggers for Security Token Service and Access Manager are different.

Security Token Service has only a single logger: `oracle.security.fed`. For more information, see "[Configuring Logging for Identity Federation](#)".

Each Access Manager component is associated with its own logger name, as listed in the following tables:

- [Table 7-3](#)
- [Table 7-4](#)
- [Table 7-5](#)

Table 7-3 Oracle Access Management Server-Side Component Loggers

Component Name	OAM Logger Name	Description
Protocol Binding	<code>oracle.oam.binding</code>	Responsible for marshalling/unmarshalling wire protocol request and response to a Java Object representation
SSO Controller	<code>oracle.oam.controller.sso</code>	Responsible for managing the user session lifecycle and orchestrating the SSO and logout flows
OAM Proxy	<code>oracle.oam.proxy.oam</code>	Responsible for interacting with OAM Webgates by marshalling/unmarshalling OAP protocol requests and responses and performing the data/message transformation necessary to help the OAM Server process OAP requests/responses
Credential Collector	<code>oracle.oam.credcollector</code>	Responsible for interacting with the user to acquire the necessary information required by the Authentication Scheme
Remote Registration of Partners	<code>oracle.oam.engine.remotereg</code>	Responsible for registering partners with the OAM Server and managing associated protected policies
Oracle Access Management Console	<code>oracle.oam.admin.console</code>	Console that supports administration and monitoring of the Access Management deployment
Admin-Service Config	<code>oracle.oam.admin.service.config</code>	Module used by the UI Console to manage the configuration
Diagnostics and Monitoring	<code>oracle.oam.dia</code> <code>g</code>	Provides instrumentation used by the OAM Server components to enable Diagnostic and Monitoring

Table 7-4 Oracle Access Management Shared-Service Engine Component Loggers

Component Name	OAM Logger Name	Description
Authentication Engine	<code>oracle.oam.engine.authn</code>	Supports establishing the identity of the user by validating the credentials and other data as required by the specified Authentication scheme
Policy Service Engine	<code>oracle.oam.engine.policy</code>	Supports management of Authentication, Authorization and Token Issuance Policies. In addition, it also provide a policy decision service to support runtime processing

Table 7-4 (Cont.) Oracle Access Management Shared-Service Engine Component Loggers

Component Name	OAM Logger Name	Description
Session Management Engine	oracle.oam.engine.session	Supports managing user session and token context information with support for user/administrator-initiated and time-out based events
Token Engine	oracle.oam.engine.token	Supports managing the entire token life cycle from generation to cancellation
SSO Engine	oracle.oam.engine.sso	Supports the single sign-on experience by managing the lifecycle of the user login session(s)
PartnerTrustMetadata Engine	oracle.oam.engine.ptmetadata	Supports management of partner metadata and trust information
Authorization Engine	oracle.oam.engine.authz	Wrapper that provides methods that map directly to OAP runtime request operations

Table 7-5 Oracle Access Management Foundation API Component Loggers

Component Name	OAM Logger Name	Description
Session Access	oracle.oam.session.access	** Not useful unless your are decompiling code
Session Access Implementation	oracle.oam.session.accessimpl	** Not useful unless your are decompiling code
Policy Access	oracle.oam.policy.access	** Not useful unless your are decompiling code

7.2.2 Sample Logger and Log Handler Definition

Here is a sample logger and a log handler for Access Manager only.

Security Token Service has only one logger and log handler, as described in ["Configuring Logging for Identity Federation"](#).

Following example illustrates the configuration of an Access Manager logger and a log handler in the file `logging.xml`.

```
<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 Java EE application logging information, see Naming and Logging Audit Files in the *Securing Applications with Oracle Platform Security Services*.

7.2.3 About Logging Levels

In Oracle Access Management, 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 "[Component Loggers for Security Token Service and Access Manager](#)"), and *notifLevel* is either an ODL message level or a Java message level.

[Table 7-6](#) shows the correspondence between ODL message levels and Java message levels, in increasing order:

Table 7-6 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)
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 Security Token Service), after restarting the server you would see logs for the OAM Server. For more information, see "[Configuring Logging for Identity Federation](#)".

7.3 Configuring Logging for Access Manager

Graphical user interface is not available to change logger levels, only WLST commands can be used.

This section describes tasks only for Access Manager.

 **See Also:**

"[Configuring Logging for Identity Federation](#)"

This section provides the following topics:

- [Modifying the Logger Level for Access Manager](#)
- [Adding an Access Manager-Specific Logger and Log Handler](#)

7.3.1 Modifying the Logger Level for Access Manager

Administrators can use custom WLST commands for Access Manager to change logger settings.

Your deployment and choices will be different.

 **Note:**

Use the WLST command `help("fmw diagnostics")`.

Follow this procedure to modify the OAM logger level.

1. Confirm that the OAM Server is running.
2. Acquire the custom WLST script for 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:

```
connect([username, password])
```
4. List available loggers for the OAM Server. For example:

```
wls:/base_domain/serverConfig> listLoggers (pattern="oracle.oam.*",
target="oam_server1")
```

```
wls:/WLS_IDM/serverConfig> listLoggers (pattern="oracle.oam.*",
target="oam_policy_mgr1")
```

Here pattern= represents the oam.controller component and target= represents the desired OAM Server as it was specified during registration.

- View the list of 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>

- 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")
```

```
wls:/WLS_IDM/domainRuntime> setLogLevel(logger="oracle.oam", level="TRACE:32",
persist="0", target="oam_policy_mgr1")
```

- 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_INSTANCE_NAME/logs/
```

9. Proceed to "[Validating Run-time Event Logging Configuration](#)".

7.3.2 Adding an 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:

Customization Commands in *WLST Command Reference for WebLogic Server*

Follow this procedure 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 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 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/
```

7.4 Configuring Logging for Identity Federation

By default Identity Federation messages are logged into the OAM Server's log files.

You can view and configure these logs in Fusion Middleware Control. However, you can also edit `logging.xml` and direct Identity Federation information to a separate log file, as described in this section. The files involved in this procedure are:

- **Logging Configuration File:** Provides logger names and other configuration information for logging. This file is stored in: `$DOMAIN_HOME/config/fmwconfig/servers/SERVER-NAME/logging.xml`.
- **Log File:** `$DOMAIN_HOME/ostslogs/SERVER-NAME-diagnostics.log`, for example.

Security Token Service and Identity Federation do not categorize log handlers as Access Manager does. Instead, there is only one logger that affects the log levels for Identity Federation. [Table 7-7](#) provides details for this logger, which are required in the WLST command.

Table 7-7 Oracle Identity Federation Loggers

Component Name	Logger Name	Log Handler Name	Log Class
Identity Federation	oracle.security.fed	stsfed-handler	class=oracle.core.ojdl.logging.ODLHandlerFactory

For details, see:

- [Configuring Logging for Identity Federation](#)
- [Defining Log Level and Log Details for Security Token Service or Identity Federation](#)

 **See Also:**

- [Monitoring Performance and Logs with Fusion Middleware Control](#) for details about how you can configure and view logs using Fusion Middleware Control
- Logging information in the Logging Audit Events Programmatically in the *Securing Applications with Oracle Platform Security Services*

7.4.1 Configuring Logging for Identity Federation

Administrators can separate Identity Federation log messages from OAM Server message logs.

To configure:

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 Identity Federation:

```
<log_handler name='stsfed-handler' class='oracle.core.ojdl.logging.ODLHandlerFactory'>
  <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='stsfed-handler' />
</logger>
```

3. Save the file.
4. Proceed with "[Defining Log Level and Log Details for Security Token Service or Identity Federation](#)".

7.4.2 Defining Log Level and Log Details for Security Token Service or Identity Federation

Administrators can use custom WLST commands for Oracle Access Management to change logger settings for Security Token Service as described here. This specifies an independent output file for only Security Token Service log messages.

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

This sample procedure for Security Token Service logging is very similar to the one for Access Manager. However, there are a few differences. Your deployment choices will be different.

1. Confirm that the OAM Server is running.
2. Acquire the custom WLST script for Oracle Access Management:

```
$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  
$DOMAIN_HOME/oiflogs/SERVER-NAME-diagnostics.log
```
7. Proceed to ["Validating Run-time Event Logging Configuration"](#).

7.5 Validating Run-time Event Logging Configuration

You can use the following procedure to test your run-time event logging configuration.

Before you begin:

- Configure logging using WLST commands
- Ensure the Agents and Servers are running.
- Configure an Application Domain to protect the resource as described in [Managing Policies to Protect Resources and Enable SSO](#).

To validate run-time event logging 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. On the physical server, verify all the logs appear in:

```
$DOMAIN_HOME/oamlogs/  
$DOMAIN_HOME/ostslogs/SERVER-NAME-diagnostics.log  
$DOMAIN_HOME/oiflogs/SERVER-NAME-diagnostics.log
```
4. Open the log file and look for the last entries to confirm authentication failure and success, respectively.

8

Auditing Administrative and Run-time Events

In Oracle Fusion Middleware, auditing refers to the process of collecting 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, and 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. (Analyzing and using audit data is outside the scope of this chapter.)

The following topics describe the administrative and run-time events that can be audited for Oracle Access Management services as well as information on configuring common auditing settings and validating your auditing configuration:

- [Introduction to Oracle Fusion Middleware Auditing](#)
- [Oracle Access Management Auditing](#)
- [Access Manager Events You Can Audit](#)
- [Identity Federation Events You Can Audit](#)
- [Setting Up Auditing for Oracle Access Management](#)
- [Validating Auditing and Reports](#)

8.1 Introduction to Oracle Fusion Middleware Auditing

Auditing provides a measure of accountability and records the data of who has done what and when.

Review the following topics in the *Securing Applications with Oracle Platform Security Services* to gain an understanding of auditing and the Audit Framework in Oracle Fusion Middleware:

- [Introduction to Oracle Fusion Middleware Audit Framework](#)
- [Setting up Oracle Business Intelligence Publisher](#)
- [Customizing Audit Reports](#)
- [Oracle Fusion Middleware Audit Framework Reference](#) for details about how the Audit database is laid out



Note:

There is nothing specific or separate related to auditing Identity Context. Unless explicitly stated, information is the same for all Oracle Access Management services.

8.2 Oracle Access Management 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.

The following topics provide information about:

- [Understanding Oracle Access Management Auditing](#)
- [About Oracle Access Management Auditing Configuration](#)
- [About Audit Record Storage](#)
- [About Audit Reports and Oracle Business Intelligence Publisher](#)
- [Oracle BI Enterprise Edition \(Oracle BI EE\)](#)
- [About the Audit Log and Data](#)

8.2.1 Understanding Oracle Access Management Auditing

Oracle Access Management uses the Oracle Fusion Middleware Common Audit Framework to support auditing for a large number of user authentication and authorization run-time events, and administrative events (changes to the system). The Oracle Fusion Middleware Common Audit Framework provides uniform logging and exception handling and diagnostics for all audit events.

Auditing is based on configuration parameters set using the Oracle Access Management Console which enables data capture for a user or set of users. While auditing can be enabled or disabled, it is normally enabled in production environments. Audit data can be written to either a single, centralized Oracle Database instance or to flat files known as bus-stop files.

 **Note:**

The Oracle Fusion Middleware Common Audit Framework database audit store does not include Access Manager policy or session-data and is not configured through the Oracle Access Management Console.

Auditing has minimal performance impact, and the information captured by auditing can be useful (even mission-critical). The audit log file helps the audit Administrator track errors and diagnose problems if the audit framework is not working properly.

8.2.2 About Oracle Access Management Auditing Configuration

An Administrator controls certain auditing parameters using the Oracle Access Management Console.

Additional auditing configuration is required through the Common Audit Framework.

 **Note:**

Oracle recommends that you use only the Oracle Access Management Console or WebLogic Scripting Tool (WLST) commands for changes. See [Updating OAM Configuration](#)

Event configuration (mapping events to levels) occurs in the `component_events.xml` file. An audit record contains a sequence of items that can be configured to meet particular requirements.

Within the Oracle Access Management Console, you can set the maximum log file and log directory size. Audit policies (known as Filter Presets) declare the types of events to be captured by the audit framework for particular components.

Audit policies cannot be configured using Fusion Middleware Control. Oracle Access Management does not use JPS infrastructure to configure the audit configuration. There are no WebLogic Scripting Tool (WLST) commands for auditing.

 **See Also:**

- [Access Manager Events You Can Audit.](#)

8.2.3 About Audit Record Storage

Audit data can be written to either a single, centralized Oracle Database instance or to flat files known as *bus-stop* files. By default, audit data is recorded to the file but 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.

- **Audit Bus-stop:** Local files containing audit data records before they are pushed to the audit data store. In the event that no audit data store is configured, audit data remains in these bus-stop files. The bus-stop files are simple text files that can be queried easily to look up specific audit events. When an audit data store is in place, the bus-stop acts as an intermediary between the component and the audit data store. The local files are periodically uploaded to the audit data store based on a configurable time interval.

Bus-stop files for Java components are located in:

```
$DOMAIN_HOME/servers/$SERVER_NAME/logs/auditlogs/OAM/audit.log
```

Bus-stop files for system components are located in:

```
$ORACLE_INSTANCE/auditlogs/OAM/oam_server1/audit.log
```

- **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.
- **Database Audit Store:** In production environments, Oracle recommends using a database audit store to provide scalability and high-availability for the Common Audit Framework. A key advantage of the audit data store is that audit data from multiple components can be correlated and combined in reports; for example, authentication

failures in all Middleware components and instances. Audit data is cumulative and grows over time so ideally this is a stand-alone RDBMS database for audit data only and not used by other applications.

 **Note:**

The preferred mode in production environments is writing audit records to a stand-alone RDBMS database for audit data only.

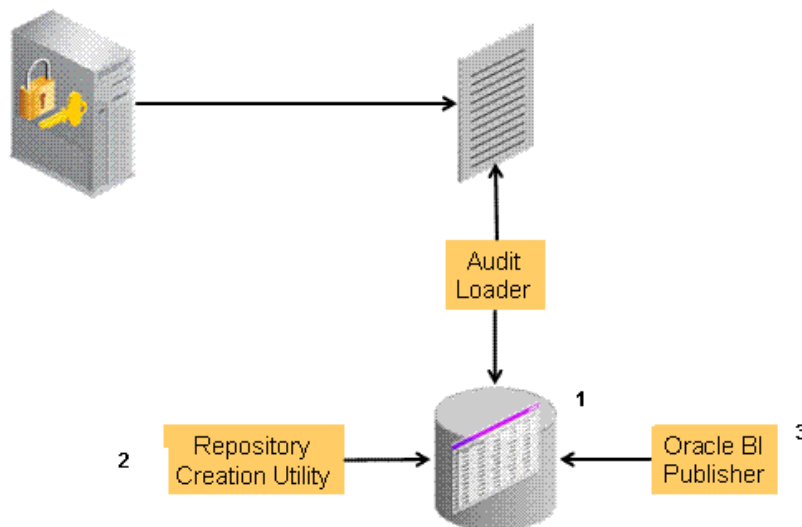
To switch to a database as the permanent store for your audit records, you must first use the Repository Creation Utility (RCU) to create a database schema for audit data. The RCU seeds that database store with the schema required to store audit records in a database. After the schema is created, configuring 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

As previously documented, the Oracle Fusion Middleware Audit Framework schema is provided by the RCU.

Figure 8-1 provides a simplified view of the audit architecture with a supported database.

Figure 8-1 Audit to Database Architecture



 **See Also:**

- See Managing Audit in the *Securing Applications with Oracle Platform Security Services*.
- See [Setting Up the Audit Database Store](#).

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.

8.2.4 About Audit Reports and Oracle Business Intelligence Publisher

Oracle Access Management integrates with Oracle Business Intelligence Publisher, which provides a pre-defined set of compliance reports through which the data in the database audit store is exposed. 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 Management and accessible with Oracle Business Intelligence Publisher. You can also use Oracle Business Intelligence Publisher to create your own custom audit reports.

8.2.5 Oracle BI Enterprise Edition (Oracle BI EE)

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.

The components of Oracle BI EE 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 BI EE 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 *Securing Applications with Oracle Platform Security Services*.

You may need to prepare Oracle BI EE for use with auditing reports for Oracle Access Management.

See [Preparing Oracle Business Intelligence Publisher EE](#).

Oracle BI EE reports contain enumerated fields, the data fields and labels of which are self-explanatory. Content of reports is described in [Table 8-1](#) (taken from Knowledge Base Doc ID 1495333.1 on My Oracle Support).

Table 8-1 Oracle Business Intelligence Enterprise Edition Reports for OAM

Report Type	Description
Account Management	User ID Timestamp Component/ Application Name Event Details
Authentication_Statistics	Authentication_statistics Failure Userid Number of Events AuthenticationFromIPByUser IP Address Distinct User Count Total Attempts Users AuthenticationPerIP IP Address Distinct Users Total Number of Attempts AuthenticationStatisticsPerServer Server Instance Name Success Count Failure Count
Errors_and_Exceptions	All_Errors_and_Exceptions User ID Timestamp Component/Application Name Client IP Address Message Event Event Details Authentication_Failures User ID Timestamp Component/ Application Name Client IP Address Authentication Method Message Event Details Authorization_Failures Users_Activities Authentication_History User ID Timestamp Component/ Application Name Client IP Address Authentication Method Message Event Details Authorization_Failures Multiple_Logins_From_Same_IP IP Address Usernames Used

See the following topics:

- [Access Manager Events You Can Audit](#)
- [Identity Federation Events You Can Audit](#)

8.2.6 About the Audit Log and Data

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 (but not limited to) Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID ContextFields, SessionId, TargetComponentType, ApplicationName, and EventCategory.



See Also:

Managing Audit in the *Securing Applications with Oracle Platform Security Services*.

8.3 Access Manager Events You Can Audit

Oracle Access Management uses the Oracle Fusion Middleware Common Audit Framework to support auditing for a large number of user authentication and authorization run-time events, and administrative events.

The following topics describe how to audit Access Manager events:

- [Access Manager Administrative Events You Can Audit](#)
- [Access Manager Run-time Events You Can Audit](#)
- [Auditing Authentication Events](#)

See Also:

- [Identity Federation Events You Can Audit](#)

8.3.1 Access Manager Administrative Events You Can Audit

Administrative events are those generated when the Oracle Access Management Console is used.

The Access Manager-specific administrative events that can be audited and the details captured for them are listed in [Table 8-2](#). These event definitions and configurations are implemented as part of the audit service in Oracle Platform Security Services.

Note:

The amount and type of information that is logged is controlled by choosing a filter preset from the Audit Configuration section. Auditable events for each filter preset are fixed in the read-only `component_events.xml` file. Editing or customizing this file is not supported.

[Table 8-3](#) lists the details that have been captured.

Table 8-2 Access Manager Administrative Audit Events

Administrative Event	Event Data Include
Oracle Access Management Console Login success/failure	<ul style="list-style-type: none"> • User name • Remote IP • Roles
Authentication Policy Creation	<ul style="list-style-type: none"> • Policy name • Authentication scheme details • Resource details • Policy type (authentication or authorization)

Table 8-2 (Cont.) Access Manager Administrative Audit Events

Administrative Event	Event Data Include
Authentication Policy Modification	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Policy name • Authentication scheme details • Resource details • Policy type (authentication or authorization)
Resource Creation	<ul style="list-style-type: none"> • Resource name • URI • Operation • Resource type
Resource Modification	<ul style="list-style-type: none"> • Resource name • URI • Operation • Resource type • Old Resource name • Old URI • Old Operation
Resource Removal	<ul style="list-style-type: none"> • Resource name • URI • Operation • Resource type
Authentication Scheme Creation	<ul style="list-style-type: none"> • Scheme name • Authentication modules • Level
Authentication Scheme Modification	<ul style="list-style-type: none"> • Scheme name • Authentication modules • Level • Old Scheme name • Old Authentication modules • Old Level
Authentication Scheme Removal (Delete)	<ul style="list-style-type: none"> • Scheme name • Authentication modules • Level
Response Creation	<ul style="list-style-type: none"> • Response name • Response key • Data source • Response Type

Table 8-2 (Cont.) Access Manager Administrative Audit Events

Administrative Event	Event Data Include
Response Modification	<ul style="list-style-type: none"> • Response name • Response key • Data source • Response Type • Old Response name • Old Response key • Old Data source
Response Removal (Delete)	<ul style="list-style-type: none"> • Response name • Response key • Data source • Response Type
Partner Addition	<ul style="list-style-type: none"> • Partner name • Partner ID • Partner URL • Logout URL
Partner Modification	<ul style="list-style-type: none"> • Partner name • Partner ID • Partner URL • Logout URL • Old Partner name • Old Partner URL • Old Logout URL
Partner Removal	<ul style="list-style-type: none"> • Partner name • Partner ID • Partner URL • Logout URL
Conditions creation	<ul style="list-style-type: none"> • Condition Name • Condition type • Condition data
Conditions Modification	<ul style="list-style-type: none"> • Condition Name • Condition type • Condition data • Old Condition name • Old Condition type • Old Condition data
Conditions Removal	<ul style="list-style-type: none"> • Condition Name • Condition type • Condition data
Server Domain creation	<ul style="list-style-type: none"> • Domain Name
Server Domain Modification	<ul style="list-style-type: none"> • Domain Name • Old Domain Name
Server Domain Removal	<ul style="list-style-type: none"> • Domain Name

Table 8-2 (Cont.) Access Manager Administrative Audit Events

Administrative Event	Event Data Include
Server configuration change	<ul style="list-style-type: none"> • New details • Old details • Instance Name • Application Name • User Name • Remote ID • Roles • Date and time

8.3.2 Access Manager Run-time Events You Can Audit

Run-time events are those generated by some of the events the Access Manager component engines issue when interacting with one another. The run-time events that can be audited, when they are issued. These event definitions and configurations are implemented as part of the audit service in Oracle Platform Security Services.

 **Note:**

The amount and type of information that is logged is controlled by choosing a filter preset in the Audit Configuration. Auditable events for each filter preset are fixed in the read-only `component_events.xml` file. Editing or customizing this file is not supported.

Table 8-3 Access Manager 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.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • Remote IP • User Name • User DN • Resource ID • Authentication scheme ID • Authentication Policy ID • Partner ID

Table 8-3 (Cont.) Access Manager Run-time Audit Events

Run-time Event	Issued When	Event Details Include
Authentication Failure	A client submits credentials and credential validation fails.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • SSO Session ID • User Name • User DN • Remote IP • Resource ID • Authentication scheme ID • Authentication Policy ID
Session Destroy	Authentication succeeds.	<ul style="list-style-type: none"> • SSO Session ID • User Name • User DN • Partner ID
Login success	A client finishes the login procedure and it is forwarded to the agent.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • Remote IP • User DN • Authentication level • SSO Session ID • Partner ID
Logout failure	A client fails to logout.	<ul style="list-style-type: none"> • Remote IP • User DN • SSO Session ID • Failure details • Partner ID

Table 8-3 (Cont.) Access Manager Run-time Audit Events

Run-time Event	Issued When	Event Details Include
Credential Collection	A client is redirected to the credential collection page.	<ul style="list-style-type: none"> Remote IP Resource Name Resource ID Authentication scheme ID Authentication Policy ID
Credential Submit	A client submits credentials.	<ul style="list-style-type: none"> Remote IP User Name Resource ID Authentication scheme ID Authentication Policy ID
Authorization Success	A client has been authorized to access a resource.	<ul style="list-style-type: none"> Remote IP User DN Resource ID Authorization Policy ID
Authorization Failure	A client has not been authorized to access a resource.	<ul style="list-style-type: none"> Remote IP User DN Resource ID Authorization Policy ID
Server Start Up	The server starts up.	<ul style="list-style-type: none"> Date and time Instance Name Application Name User Name
Server Shut Down	The server shuts down.	<ul style="list-style-type: none"> Date and time Instance Name Application Name User Name

8.3.3 Auditing Authentication Events

Auditing events during authentication can help Administrators scrutinize security weaknesses in their systems.

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 may 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.

Information about users requesting authentication or brute force attacks can be stored in the file system or in a back-end database.

8.4 Identity Federation Events You Can Audit

The Identity Federation service also uses the Fusion Middleware Audit Framework for auditing.

The following data is part of each audit record, regardless of the event or event type that is audited:

- timestamp - Date and time the audit event occurred
- initiator - the initiator of the audit event (for some events this attribute may be empty)
- ECID - the execution context ID

The Fusion Middleware Audit Framework supports the following audit levels:

- None
- Low
- Medium
- Custom

Events can be audited in different categories and audit levels.

[Table 8-4](#) lists the event categories.

Table 8-4 Categories of Audit Events for Identity Federation

Category	Described in ...
Session Management	Session Management Events for Identity Federation
Protocol Flow	Protocol Flow Events for Identity Federation
Server Configuration	Server Configuration Events for Identity Federation
Security	Security Events for Identity Federation

The following section contain more information.

- [Session Management Events for Identity Federation](#)
- [Protocol Flow Events for Identity Federation](#)
- [Server Configuration Events for Identity Federation](#)
- [Security Events for Identity Federation](#)

8.4.1 Session Management Events for Identity Federation

Session Management events for this Identity Federation release, include a subset of auditable events for the previous release.

Table 8-5 Identity Federation Session Management Events

Auditable Events	Auditing Not Supported in This Release for ...
CreateUserSession – Creation of a session after a successful login	CreateUserFederation – Creation of a user federation between two remote servers
DeleteUserSession – Deletion of a session after logout	UpdateUserFederation - Updating the user federation between two remote servers
CreateActiveUserFederation – Creation of an active federation after successful login	DeleteUserFederation – Deletion of a user federation between two remote servers
CreateActiveUserFederation – Creation of an active federation after successful login	
DeleteActiveUserFederation - Deletion of an active federation after logout	
LocalAuthentication – Authentication of a user at OIF	
LocalLogout - Logout of a user at Identity Federation	

8.4.2 Protocol Flow Events for Identity Federation

Protocol flow events for this Identity Federation release, include a subset of auditable events for the previous Identity Federation release.

Table 8-6 Protocol Flow Events for Identity Federation

Auditable Events	Auditing Not Supported in This Release for ...
IncomingMessage Message being received by Identity Federation	AssertionCreation Creation of an assertion by Identity Federation (Success only)
OutgoingMessage Message being sent by Identity Federation (Success only)	
AssertionConsumption Consumption of an assertion by Identity Federation (Success only)	

8.4.3 Server Configuration Events for Identity Federation

Auditable Server configuration events for this Identity Federation release, include a subset of auditable events for the previous Identity Federation release.

Table 8-7 Server Configuration Identity Federation

Auditable Events	Auditing Not Supported in This Release for ...
CreateConfigProperty Adding a new configuration property (Success only)	SetDataStoreType Changing the type of a data store (Success only)
ChangeConfigProperty Changing the value of an existing configuration property (Success only)	ChangeDataStore Setting of the federation data store (Success only)
DeleteConfigProperty Deleting a configuration property (Success only)	
CreatePeerProvider Adding a new provider to the list of trusted providers (Success only)	
UpdatePeerProvider Updating the information on an existing provider in the list of trusted providers (Success only) PeerProviderID	
DeletePeerProvider Deleting a provider from the list of trusted providers (Success only)	
LoadMetadata Loading of metadata (Success only)	
ChangeFederation Changing of the trusted providers (Success only)	
ChangeServerProperty Changing of a server configuration property (Success only)	

8.4.4 Security Events for Identity Federation

Auditable security events for this Identity Federation release, include all auditable events for the previous Identity Federation release.

Table 8-8 Security Events for Identity Federation

Auditable Events	Auditing Not Supported in This Release for ...
CreateSignature Creation of a digital signature by Identity Federation	n/a

Table 8-8 (Cont.) Security Events for Identity Federation

Auditable Events	Auditing Not Supported in This Release for ...
VerifySignature Verification of a digital signature by Identity Federation	
EncryptData Encryption of data by Identity Federation	
DecryptData Decryption of data by Identity Federation	

8.5 Setting Up Auditing for Oracle Access Management

Before you perform auditing for Oracle Access Management, ensure to set up the audit data store and set up publishing for audit reports.

The following overview provides a list of the tasks that must be performed before auditing:

1. Set up the audit data store.
See [Setting Up the Audit Database Store](#).
2. Set up publishing for audit reports.
See [Preparing Oracle Business Intelligence Publisher EE](#).
3. Edit the Audit Configuration in the Oracle Access Management Console, as described in:
 - [Using the Oracle Access Management Console for Audit Configuration](#)
 - [Adding, Viewing, or Editing Audit Settings](#)

See [Validating Auditing and Reports](#) for details on how to test and validate the audit configuration.

8.5.1 Setting Up the Audit Database Store

Here is 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 Management if you choose a database store for audit data.



See Also:

- *Managing the audit Store in the [Securing Applications with Oracle Platform Security Services](#).*
- *[Oracle Fusion Middleware Repository Creation Utility User's Guide](#)*

To create an audit database store:

1. Create an audit database, version 11.1.0.7 or later.
See Audit Database Administration in the *Securing Applications with Oracle Platform Security Services*.
2. Run the RCU against the database.
See "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.
See About Audit Data Sources in the *Securing Applications with Oracle Platform Security Services*:
 - Use the Java EE audit loader configuration for WebLogic Server.
 - Use the JNDI name of the data source 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_ORACLE`. For example:

```
<serviceInstance name="audit.db" provider="audit.provider">
  <property name="audit.loader.repositoryType" value="DB_ORACLE"/>
  <property name="audit.timezone" value="utc"/>
  <property name="audit.loader.interval" value="15"/>
  <property name="audit.maxFileSize" value="104857600"/>
  <property name="audit.reports.jndi" value="jdbc/AuditViewDataSource"/>
  <property name="audit.filterPreset" value="None"/>
  <property name="audit.loader.jndi" value="jdbc/AuditAppendDataSource"/>
  <propertySetRef ref="props.db.1"/>
</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 Audit Terminology in the *Securing Applications with Oracle Platform Security Services*.
7. Maintain the bus-stop files, as described in Managing the bus-stop files in the *Securing Applications with Oracle Platform Security Services*.

8.5.2 Preparing Oracle Business Intelligence Publisher EE

You must prepare Oracle Business Intelligence Publisher Enterprise Edition (EE) for use with Oracle Access Management audit reports.

Here is an outline of the procedure to prepare Oracle Business Intelligence Publisher EE.

 **See Also:**

- 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*
- *Securing Applications with Oracle Platform Security Services*

To prepare Oracle Business Intelligence Publisher:

1. Install Oracle BI Publisher.

See the *Oracle Business Intelligence Enterprise Edition Installation and Upgrade Guide*.

 **Note:**

While configuring BI domain in the **Configuration Assistant**, select only **Business Intelligence Publisher** to ensure OAM reports can be viewed by performing the following steps. If you integrate **Business Intelligence Publisher** with **Business Intelligence Enterprise Edition**, or select only **Business Intelligence Enterprise Edition**, you must upload the reports from the BI Publisher admin console only and the following steps are not applicable.

2. Perform the following tasks:

See Oracle Business Intelligence Publisher Time Zone in the *Securing Applications with Oracle Platform Security Services*:

- Unjar the common report `AuditReportTemplates.jar` to `$BI_DOMAIN/bidata/components/bipublisher/repository/Reports/` from the location `$OAM_MW_HOME/oracle_common/modules/oracle.iau/reports`.
- Unzip the oam-specific reports `oam_audit_reports.zip` into `$BI_DOMAIN_HOME/bidata/components/bipublisher/repository/Reports/Oracle_Fusion_Middleware_Audit/Component_Specific` from the location `$OAM_ORACLE_HOME/oam/server/reports`.
- Set up the JNDI connection for the audit data source or the JDBC connection the audit database.

The datasource name must be "Audit".

3. Set up audit report templates. See About Audit Reporting in the *Securing Applications with Oracle Platform Security Services*.
4. Set up audit report filters, as described in Managing Audit Policies with Fusion Middleware Control in the *Securing Applications with Oracle Platform Security Services*.

5. View reports from the following path: `Reports/Oracle_Fusion_Middleware_Audit_Reports/Component_Specific/OAM`.



See Also:

[Validating Auditing and Reports](#)

8.5.3 Using the Oracle Access Management Console for Audit Configuration

Within Oracle Access Management, 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 8-2 shows the Audit Configuration section of the Common Settings page.

Figure 8-2 Common Settings: Auditing Configuration

▲ Audit Configuration

* Maximum Directory Size (MB)

* Maximum File Size (MB)

Filter Enabled

* Filter Preset

Audit Configuration

View ▼	+ Add	× Delete
Users		
<input type="text" value="orcladmin"/>		
<input type="text" value="SSOAdmin"/>		

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 Management 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 8-9 describes the elements in the Audit Configuration page.

Table 8-9 Audit Configuration Elements

Elements	Description
Maximum Directory Size	<p>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.</p> <p>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.</p> <p>This is configured using the <code>max.DirSize</code> property described in the configuration file <code>filejps-config.xml</code>. This property controls the maximum size of a bus-stop directory for Java components as described in the About Audit in Java SE Applications in the <i>Securing Applications with Oracle Platform Security Services</i>.</p>
Maximum File Size	<p>The maximum size, in MBs, of an audit log file. Once the size of a file reaches the maximum size, a new log file is created. For example, specifying 10 directs file rotation when the file size reaches 10 MB.</p> <p>This is configured using the <code>max.fileSize</code> property described in the configuration file <code>filejps-config.xml</code>. This property controls the maximum size of a bus-stop file for Java components as described in About Audit in Java SE Applications in the <i>Securing Applications with Oracle Platform Security Services</i>.</p>
Filter Enabled	Check this box to enable event filtering.
Filter Preset	<p>Defines the amount and type of information that is logged when the filter is enabled. The default value is Low.</p> <ul style="list-style-type: none"> • 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 <p>Events for each filter preset are fixed in the read-only <code>component_events.xml</code> file. Editing or customizing this file is not supported for Oracle Access Management. Only items that are configured for auditing at the specified filter preset can be audited.</p>
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.

8.5.4 Adding, Viewing, or Editing Audit Settings

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.



Note:

Auditable events for each filter preset are fixed in the read-only `component_events.xml` file. Editing or customizing this file is not supported.

The following procedure describes how to add, view, or edit OAM Server Common Audit Configuration settings. Individual audit policies cannot be configured using Fusion Middleware Control. Oracle Access Management does not use JPS infrastructure to configure the audit configuration. There are no WebLogic Scripting Tool (WLST) commands for auditing.

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. In the **Settings** section, select **Common Settings** from the **View** menu.
3. In the Audit Configuration section, enter appropriate details for your environment See [Table 8-9](#):
 - Maximum Log directory size
 - Maximum Log file size
 - Filter Enabled
 - Filter Preset (to define verbosity of audit data)
 - Users to include specific users from the audit by clicking the Add (+) button above the Users table and entering a value in the field.
4. Click Apply to submit the Audit Configuration (or close the page without applying changes).

8.6 Validating Auditing and Reports

The run-time event auditing configuration can be tested.

Before you begin:

- Configure auditing parameters.
See [Setting Up Auditing for Oracle Access Management](#).
 - Ensure the Agents and Servers are running.
 - Prepare BI EE Publisher.
See [Preparing Oracle Business Intelligence Publisher EE](#).
1. **To validate an Authentication Event:** Audit Console login success/failure as described here or any administrative event.
See [Table 8-2](#).

- a. Sign out of Oracle Access Management Console.
- b. Sign in to Oracle Access Management Console with invalid user (not Administrator) credentials.
- c. Sign in to Oracle Access Management Console using the proper Administrator credentials.
- d. **Review Log File:** Open the audit.log file and search for the last Administrative event entries:

`$DOMAIN_HOME/servers/$ADMINSERVER_NAME/logs/auditlogs/OAM/audit.log`

- e. **Review Database Log:**
 - i. Perform the following tasks.
See [Setting Up the Audit Database Store](#).
 - ii. Generate an Authentication event as described in Step 1.
 - iii. Connect to the database and connecting to the database and reviews audit events under IAU_BASE table.
2. **To validate a Runtime Event:** Audit Authorization success/failure as described here or any runtime event that is described as follows:

See [Table 8-3](#).

- a. In a browser window, enter the URL of a protected resource for which you are not authorized.
- b. **Review Log File:** Open the audit.log file and search for the last Administrative event entries:

`$DOMAIN_HOME/servers/$RUNTIMESERVER_NAME/logs/auditlogs/OAM/audit.log`

- c. **Review Database Log:**
 - i. Perform the following tasks.
See [Setting Up the Audit Database Store](#).
 - ii. Generate and Authentication event as described in Step 1.
 - iii. Connect to the database and connecting to the database and reviews audit events under IAU_BASE table.

3. **To validate Audit Configuration Changes:**

See Also [Adding, Viewing, or Editing Audit Settings](#) .

- a. From the Oracle Access Management Console, System Configuration tab, Common Configuration, modify Maximum Directory Size (MB) and Maximum File Size (MB) parameters.
- b. Repeat Steps here to confirm auditing is working.

4. **To View Reports:**

- a. 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.

- b. In Oracle BI Publisher Enterprise, locate the desired reports. For example:

Click Shared Folders, the component that contains the report you would like to view and then select the desired report.

- c. Perform any analysis as desired, or edit your auditing configuration as needed.

```
$MW_HOME/user_projects/domains/base_domain/servers/oam_server1/logs/  
auditlogs/OAM/
```

- 5. Archive and manage audit logs according to your company policies.

9

Logging WebGate Event Messages

Each WebGate instance can write information about its processes and states to a log file.

The logs can be configured to provide information at various levels of granularity. For example, you can record errors, errors plus state information, or errors, states, and other information to the level of a debug trace. You can also eliminate sensitive information from the logs.

This chapter provides the following sections.

- [Understanding Logging for WebGate Instances](#)
- [About Log Configuration File Paths and Contents](#)
- [About Directing Log Output to a File or the System File](#)
- [Structure and Parameters of the WebGate Log Configuration File](#)
- [Activating and Suppressing Logging Levels](#)
- [Mandatory Log Configuration File Parameters](#)
- [Configuring Different Threshold Levels for Different Types of Data](#)
- [Filtering Sensitive Attributes](#)

9.1 Understanding Logging for WebGate Instances

The logging feature enables you to analyze system performance and to troubleshoot issues.

You can configure logging for individual WebGate instances of the following components:

- OAM WebGates
- Custom Access Clients (Access Manager SDK)

This section discusses the following topics:

- [About Logging, Log Levels, and Log Output](#)
- [Log Levels](#)
- [Log Output](#)

9.1.1 About Logging, Log Levels, and Log Output

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: 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](#)".

Configuring Logging: You configure logging by editing a configuration file that is stored with the Webgate. See "[About Log Configuration File Paths and Contents](#)".

Logging Levels: You can request logging at various levels. The highest level is Fatal and the lowest level is Trace. See "[Log Levels](#)" 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](#)" 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 "[Parameters in the WebGate Second Compound List and Log Handlers](#)" for details.

9.1.2 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 9-5](#) describes the levels. The default log level is Warning: LOGLEVEL_WARNING.

Table 9-1 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.
LOGLEVEL_DEBUG1	> 400	Records debugging information. Typically, the information at this level is only meaningful to a developer.

Table 9-1 (Cont.) Logging Levels

Level	Number of Events Reported	Description
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 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 "[Parameters in the WebGate Second Compound List and Log Handlers](#)" 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 "[Parameters in the WebGate Simple List and Logging Threshold](#)" 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](#)" for details.

 **Note:**

The Trace and Debug3 level logs can contain sensitive information. For more information about sensitive information, see "[Filtering Sensitive Attributes](#)".

9.1.3 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_mgr.cpp:212
```

To the right of the internal code reference columns, you see the log message that is associated with this log level, for example, "Function called" or "Function returned," followed by the name of the function, as illustrated in the following example:

```
"Function called"    _CallName^ldap_init
```

The log message and function name can be followed by additional information, for example, the duration of the process, the address space where the function is running, or state information, as illustrated in the following examples:

```
"Connection health check result"    Server^dlsun4072    Port^389    Server
Priority^1    Connection available^true
```

```
"Function entered"    _TraceName^ConnectionWatcherThread::CheckPrimaries
```

```
"Function exited"    _TraceName^ConnectionWatcherThread::CheckPrimaries
TraceDuration^0.000028
```

```
"Connection Pool Status in ValidateConnections()"    "NumLivePrimaryConnections^1
Maximum Connections^1    UpConnections^1    Failover Threshold^1    Max
Session Time^0    SleepFor^60
```

To secure sensitive information and ensure that it is not included in the output of the logging operation, see ["Filtering Sensitive Attributes"](#).



See Also:

["Log Configuration File Contents"](#)

9.2 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 can be edited with a plain text editor. Changes made to these files are effective immediately.

Details are in the following sections:

- [Log Configuration File Paths and Names](#)
- [Log Configuration File Contents](#)

9.2.1 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 "[Log Output](#)".

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:

11g WebGates: `$WEBGATE_HOME` or `$ORACLE_HOME/webgate/ohs/config`. The same `oblog_config_wg.xml` file is copied to the WebGate instance directory (`$INSTANCE_HOME/webgate/config`) when the WebGate instance is created. The later is to be used when configuring logging.



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, `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 9-2](#) lists the names of the log configuration files. Do not change the names.

Table 9-2 Log Configuration File Names for Components

Component	Log Configuration File Name
Webgate	<code>oblog_config_wg.xml</code>
Access Manager SDK (custom Access Client)	<code>oblog_config.xml</code>



Note:

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 effect in about 60 seconds.

9.2.2 Log Configuration File Contents

The configuration file controls log items such as file rotation intervals and it contains XML statements that you can edit in a text editor.

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

9.2.2.1 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.

9.2.2.2 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.    -->
```

Following example shows a typical 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 -->

```



```

<NameValPair ParamName="Passwd" Value="40" />
<NameValPair ParamName="response" Value="40" />
<NameValPair ParamName="Response" Value="40" />
</ValNameList>
</CompoundList>

```

9.3 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 "[Parameters in the WebGate Second Compound List and Log Handlers](#)" for details.

The log writers are described in [Table 9-3](#).

Table 9-3 Log Writers

Writer	Description
<code>SysLogWriter</code>	<p>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.</p> <p>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.</p> <p>The default log configuration file sends Fatal, Error, and Warning messages to the system log file.</p>
<code>FileLogWriter</code>	<p>This writer is recommended when you want to save log data for an OAM Server or other single-process application on a disk file.</p> <p>The <code>FileLogWriter</code> 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 <code>MultipartFileWriter</code>.</p>

Table 9-3 (Cont.) Log Writers

Writer	Description
MPFileLogWriter	<p>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.</p> <p>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.</p>

9.4 Structure and Parameters of the WebGate 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.

The rest of this section discusses the following topics:

- [Structure of WebGate Log Configuration XML File Header](#)
- [Structure of WebGate Initial Compound List](#)
- [Parameters in the WebGate Simple List and Logging Threshold](#)
- [Parameters in the WebGate Second Compound List and Log Handlers](#)
- [Parameters in the WebGate List for Per-Module Logging](#)
- [Parameters in the WebGate Filter List](#)
- [WebGate XML Element Order](#)

9.4.1 Structure of WebGate Log Configuration XML File Header

The XML file header is available at the beginning of the log configuration file.

```
<?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.

9.4.2 Structure of WebGate Initial Compound List

The header is followed by an initial compound list.

The initial compound list 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.

9.4.3 Parameters in the WebGate Simple List and Logging Threshold

After the start-tag for the first compound list, a simple list sets the global defaults for logging.

The start and end tags for this list are as follows:

```
<SimpleList>
. . .
</SimpleList>
```

Between the start and end tags of the simple list, you configure the following:

Table 9-4 Global Parameters in the First Compound List

Parameter	Description
<code>LOG_LEVEL_THRESHOLD</code>	<p>Sets the default logging threshold. Default value: <code>LOGLEVEL_WARNING</code> Possible Values: Refer to log levels in "Log Levels"</p> <p>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" for details.</p>
<code>SECURE_LOGGING</code>	<p>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</p>
<code>LOG_SECURITY_THRESHOLD_LEVEL</code>	<p>Indicates the log threshold for which secure logging is effective. Default value: <code>LOGLEVEL_TRACE</code> Possible Values: Refer to log levels in "Log Levels"</p> <p>Note: Ensure that <code>LOG_THRESHOLD_LEVEL</code> and <code>LOG_SECURITY_THRESHOLD_LEVEL</code> are the same or are consistent with one another. For example, if <code>LOG_THRESHOLD_LEVEL</code> is set to <code>LOGLEVEL_TRACE</code> while <code>LOG_SECURITY_THRESHOLD_LEVEL</code> is set at <code>LOGLEVEL_WARNING</code>, then secure logging applies to <code>LOGLEVEL_WARNING</code> and above but does not apply to <code>LOGLEVEL_TRACE</code>.</p>

Table 9-4 (Cont.) Global Parameters in the First Compound List

Parameter	Description
LOG_SECURITY_ESCAPE_CHARS	<p>Configure escape sequence characters used to avoid additional information being overwritten due to the secure logging mechanism. Use a comma separated list as shown here.</p> <p>Default value:),]</p> <p>Possible Values: Characters only</p> <p>Note: Default values are recommended. Configuring inappropriate characters may lead to sensitive information being unmasked.</p>
LOG_SECURITY_MASK_LENGTH	<p>Specifies the default masking length if none is specified in FILTER_LIST.</p> <p>Default value: 300</p> <p>Possible Values: Positive integer</p> <p>Note: FILTER_LIST appears after the second compound list (log handlers). For more information, see "Filtering Sensitive Attributes".</p>

Following example shows the simple lists containing global settings, which appear in the first compound list in the oblog_config_wg.xml file.

```
<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>
```

9.4.4 Parameters in the WebGate 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 9-1](#), 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 9-3](#), 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 9-7](#) 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 file size that determines when a new log file is generated, as follows:

```
<NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" />
```

- A time in seconds that determines the interval at which a new log file is generated, as follows:

```
<NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" />
```

9.4.5 Parameters in the WebGate 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"](#) for details.

9.4.6 Parameters in the WebGate 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 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 the following example:

```
<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.

9.4.7 WebGate 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 the following examples are equivalent:

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
  <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
  <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
```

```
<ValNameList xmlns="http://www.example.com" ListName="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 the following examples are equivalent:

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
<ValNameList ListName="LogError2Sys" xmlns="http://www.example.com">
```

9.5 Activating and Suppressing Logging Levels

For a particular log-handler, the active status of a logging level is determined by a set of factors.

Table 9-5 lists these factors.

Table 9-5 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 9-1 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 " 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 " 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 ".

9.5.1 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 9-3](#).

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](#)" for details.

9.6 Mandatory Log Configuration File Parameters

At minimum, each log-handler definition contains five parameters.


The parameters are listed in [Table 9-6](#).

Table 9-6 Mandatory Log Configuration File Parameters

Parameter	Comment
xmlns	This parameter is specified in the opening <code>ValNameList</code> tag. It specifies the relevant XML namespace for the current list and is identical for all log-handler definitions in a given logging configuration file. Example: <code>http://www.example.com</code>
ListName	This parameter is specified in the opening <code>ValNameList</code> 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: <code>WarningsAndAboveToSyslog</code> sends Fatal, Error, and Warning messages to the system log file. <code>WarningsOnlyToFileLog128KBuffer</code> sends messages from just the Warning level to a 128KB buffer, and hence to a disk file. <code>TraceOnlyToMPRotateDaily</code> sends messages from just the Trace level to the multi-process file writer, which opens and closes the file each time it writes to disk. This file is replaced with a fresh (empty) file every day, regardless of the size of the file at the time of replacement.
LOG_LEVEL	This specifies a log level. See Table 9-1 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 9-3 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 9-7](#) are relevant.

Table 9-7 Log Data File Configuration Parameters

Parameter	Description	Default
FILE_NAME	<p>Mandatory. Used only for the FileLogWriter or MPFileLogWriter. It is the name and location of the file where log data is written.</p> <p>You can prepend an absolute path to the file name to store it somewhere other than the default location, which is: <i>component_install_dir\oblix\logs</i></p> <p>Where <i>component_install_dir</i> is the root installation directory for the component whose system events you are logging.</p> <p>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.</p>	oblog.log
BUFFER_SIZE	<p>Optional. This is the size of the buffer, in bytes, for logged data as it is being written to the log file.</p> <p>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.</p>	65535 (64KB)
MAX_ROTATION_SIZE	<p>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.</p>	52428800 (512KB)
MAX_ROTATION_TIME	<p>Optional. A time interval, in seconds, when the log file is renamed, whether or not it has reached the maximum rotation size.</p> <p>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.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 20px;"> <p> Note:</p> <p>The minimum value that can be set for this parameter is 3600 seconds.</p> </div>	86400 (1 day, in seconds)

The following sections contain more information.

- [Settings in the Default Log Configuration File](#)
- [Description of the Settings in the Default Log Configuration File](#)

9.6.1 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 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.

Following example shows the default log configuration file with comments removed to expose the file structure:

```
<?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>
```

9.6.2 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:

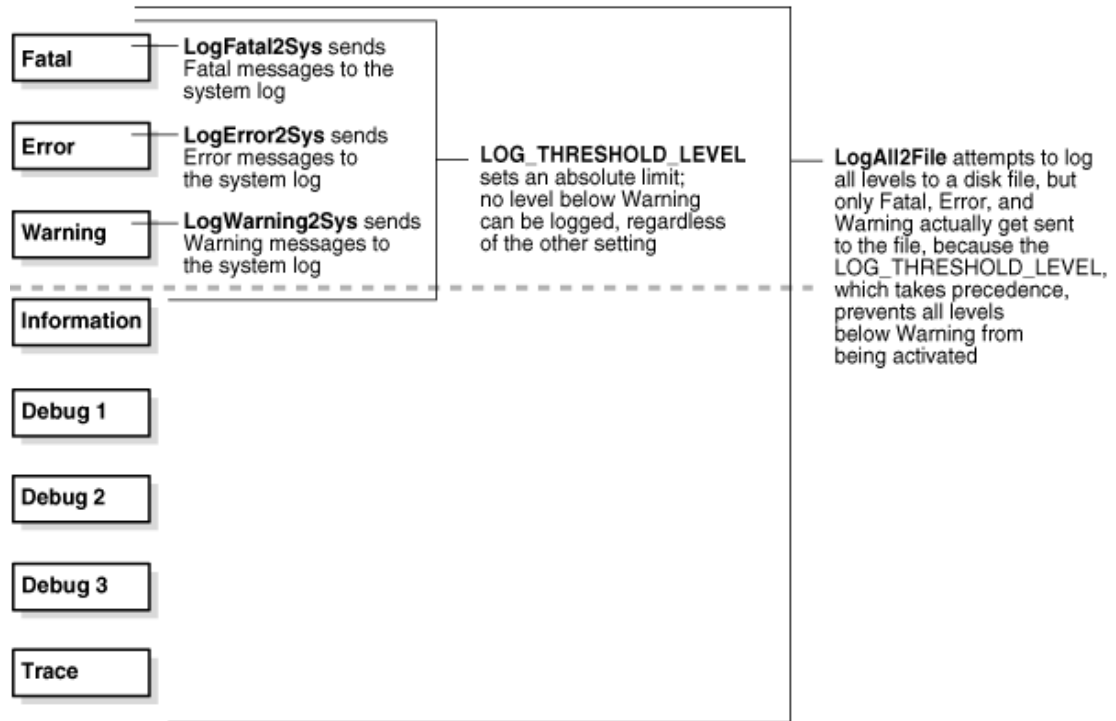
- It sets the `LOG_THRESHOLD_LEVEL` to `Warning`.
The threshold suppresses logging for levels that are more fine-grained than `Warning`. You can override this threshold. See "[Configuring Different Threshold Levels for Different Types of Data](#)" for details.

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 9-1](#) illustrates log-level activation in the default log confirmation file.

Figure 9-1 Log-Level Activation in the Default Log Configuration File



9.7 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 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](#)

9.7.1 About the MODULE_CONFIG Section

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`.

As described in "[Structure and Parameters of the WebGate Log Configuration File](#)", 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>
```

Only one instance of `MODULE_CONFIG` 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 9-8](#) 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 9-1](#) 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.

The following sections contain more information.

- [Location of the Per-Module Logging Section in the Log Configuration File](#)
- [List of Modules That Can Be Logged](#)

9.7.1.1 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 "[Configuring a Log Level Threshold for a Function or Module](#)" for details.

9.7.1.2 List of Modules That Can Be Logged

You can specify values for the `ParamName` parameter in the `MODULE_CONFIG` list.

[Table 9-8](#) describes the a partial list of values that can be logged.

Table 9-8 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. <pre><ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG"> <NameValPair Paramname="AAA_ACTIONS" Value="OFF"> </NameValPair></pre>
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 <i>Developing Applications with Oracle Access Management</i> 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 <i>Developing Applications with Oracle Access Management</i> for details.
AUDIT	Sets a logging threshold for auditing. See Auditing Administrative and Run-time Events 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.

Table 9-8 (Cont.) ParamName Values You Can Configure for Per-Module Logging Threshold

ParamName Value	Logging Threshold That This Parameter Sets
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.
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.
LDAP	Sets a logging threshold for LDAP SDK, for example: <pre><ValNameList xmlns="http://www.oblix.com" ListName="MODULE_CONFIG"> <NameValPair Paramname="LDAP" Value="LOGLEVEL_TRACE"> </NameValPair></pre>
NET	Sets a logging threshold for network APIs.
OBMYGROUPS	Sets a logging threshold for <code>ObMyGroups</code> 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.
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.

9.7.2 Configuring a Log Level Threshold for a Function or Module

You can configure a function- or module-specific log level threshold.

To configure:

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 9-8](#) for the modules that you can supply on the `ParamName` parameter. See [Table 9-1](#) 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>

```

9.8 Filtering Sensitive Attributes

You can activate secure logging and expand the default filter list to mask sensitive information from the log file.

When you add an 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 simply use `Phone`.

See Also:

- ["Understanding Logging for WebGate Instances"](#)
- ["Parameters in the WebGate Simple List and Logging Threshold"](#)
- ["Parameters in the WebGate Filter List"](#)
- ["Settings in the Default Log Configuration File "](#)

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 9-4](#). 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 "[Parameters in the WebGate Filter List](#)":

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.

10

Understanding Oracle Access Management Reports

Oracle Access Manager enables you to use Oracle BI Publisher as the reporting solution for Oracle Access Management services. Access Manager provides a restricted-use license for Oracle BI Publisher and easy-to-use reporting packages.

This chapter contains the following sections.

- [About Reports in Oracle Access Management](#)
- [Accessing Oracle Access Management Reports](#)
- [Supported Output Formats](#)
- [Classification of Reports for Access Manager](#)
- [About Creating Reports Using Third-Party Software](#)

Note:

For large-scale deployments, it is recommended that you deploy a dedicated enterprise-class reporting solution. A solution based on tools such as Oracle Business Intelligence Enterprise Edition can provide the flexibility, automation, and performance required for a large-scale organizations.

10.1 About Reports in Oracle Access Management

Oracle Access Management integrates with Oracle Business Intelligence Publisher, which provides a pre-defined set of compliance reports. 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 Management and accessible with Oracle Business Intelligence Publisher. You can also use Oracle Business Intelligence Publisher to create your own custom 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. The following are Oracle Access Management reporting features:

- Select and view reports from a predefined list in the BI Publisher.
- Filter report information.
- View reports on-screen in the desired format.
- Provide interactive reports.

10.2 Accessing Oracle Access Management Reports

To access Access Manager Reports, you must start BI Publisher and run them. BI Publisher cannot be accessed through the Access Manager Console. You must open BI publisher explicitly to access Access Manager reports.

Follow this procedure to start BI Publisher.

1. Navigate to **Start, Oracle BI Publisher Desktop, Oracle - BIPHome10134** and click **Start BI Publisher**.

The Oracle BI Publisher Home page appears.

2. Enter the user name and password.
3. Click Sign In.

Follow this procedure to run a report.

1. Start Access Manager Reports.
See "[Accessing Oracle Access Management Reports](#)" for more information.

2. Click the more... link under Shared Folders.
3. Click Access Manager Reports to access the reports.

Alternately, click the more... link under Access Manager Reports. The resulting page displays the Access Manager Reports classified according to functional area.

4. Select the report to view by clicking its name.
5. Click View.

The Report Input Parameters page displays the input parameters that must be provided to run a report. The parameters act as filter criteria. In some cases, at least one or more fields are mandatory while some reports do not require any input parameters. If you leave the input parameter field blank and click View, all the information associated with the report is displayed.

6. Enter the required parameters, if any.
7. Click View to run the report.

The report is displayed.

10.3 Supported Output Formats

All BI Publisher reports are generated in a native XML format. This XML can be transformed into other output formats.

The following formats are supported:

- HTML
- PDF
- RTF
- MHTML

10.4 Classification of Reports for Access Manager

Access Manager Reports are classified based on functional area.

For example, Access Policy Reports, Attestation, Request and Approval Reports and Password Policy Reports are available. (It is no longer named Operational and Historical.) Oracle Access Manager Reports are classified into the following categories based on their functional areas:

- [Account Management Reports](#)
- [Authentication Reports](#)
- [Errors and Exceptions](#)

10.4.1 Account Management Reports

The Accounts_Locked_Out Report is the account management report that allows administrators to view details about accounts that have been locked out.

Table 10-1 Accounts_Locked_Out Report Fields

Field	Description
User ID	Identifier of the locked out user
Timestamp	Time stamp of the lockout
Component/Application Name	Component from which the user has been locked out
Event Details	Additional information

10.4.2 Authentication Reports

Authentication reports allow administrators to view details regarding user authentications.

The reports include:

- [Authentication Statistics Report](#)
- [AuthenticationFromIPByUser](#)
- [AuthenticationPerIP](#)
- [AuthenticationStatisticsPerServer Report](#)

10.4.2.1 AuthenticationFromIPByUser

The AuthenticationFromIPByUser report contains details regarding failed and successful authentications from a particular IP address.

Table 10-2 AuthenticationFromIPByUser Report Fields

Field	Description
IP Address	IP address of the client
Distinct User Count	Number of distinct users
Total Attempts	Number of authentication attempts from this IP address
Users	List of users attempting authentication from this IP address

10.4.2.2 AuthenticationPerIP

The AuthenticationPerIP report contains details regarding failed and successful authentications from this IP address.

Table 10-3 AuthenticationPerIP Report Fields

Field	Description
IP Address	IP address of the server
Distinct Users	Number of users authenticated
Total Number of Attempts	Number of authentication attempts (successful and failed)

10.4.2.3 Authentication Statistics Report

The authentication report contains details regarding failed and successful authentications.

Table 10-4 Authentication_statistics Report Fields

Field	Description
Failure	Failed (yes) or successful (no) authentication
Userid	Identifier of the user
Number of Events	Number of authentication events

10.4.2.4 AuthenticationStatisticsPerServer Report

The AuthenticationStatisticsPerServer report contains details regarding failed and successful authentications from a particular server instance.

Table 10-5 AuthenticationStatisticsPerServer Report Fields

Field	Description
Server Instance Name	Identifier of the server instance
Success Count	Number of successful authentications
Failure Count	Number of failed authentications

10.4.3 Errors and Exceptions

The errors and exceptions report allows administrators to view errors and exceptions logged during the authentication process.

This report include:

- [All Errors and Exceptions](#)
- [Authentication Failures](#)
- [User Activities](#)
- [Authentication History](#)
- [Authorization History](#)
- [Multiple Logins From Same IP](#)

10.4.3.1 All Errors and Exceptions

All Errors and Exceptions report contains details regarding errors and exceptions encountered during runtime.

Table 10-6 All Errors and Exceptions Report Fields

Field	Description
User ID	Identifier of the locked out user
Timestamp	Time stamp of the lockout
Component/Application Name	Component from which the user has been locked out
Client IP Address	IP address of the client
Message Event	The error or exception
Event Details	Information regarding the error or exception

10.4.3.2 Authentication Failures

The Authentication Failures report contains details regarding failed and successful authentications.

Table 10-7 Authentication Failures Report Fields

Field	Description
User ID	Identifier of the locked out user
Timestamp	Time stamp of the lockout
Component/Application Name	Component from which the user has been locked out
Client IP Address	IP address of the client
Authentication Method	Authentication method
Message Event Details	Message regarding the failed authentication
Authorization_Failures	Authorization failure

10.4.3.3 User Activities

There are no fields to define in the User Activities report.

10.4.3.4 Authentication History

The Authentication History report contains details regarding failed and successful authentications.

Table 10-8 Authentication History Report Fields

Field	Description
User ID	Identifier of the locked out user
Timestamp	Time stamp of the lockout
Component/Application Name	Component from which the user has been locked out
Client IP Address	IP address of the client
Authentication Method	Authentication method
Message Event Details	Message regarding the failed authentication
Authorization_Failures	Authorization failure

10.4.3.5 Authorization History

The Authorization History report contains details regarding failed and successful authorizations.

Table 10-9 Authorization History Report Fields

Field	Description
User ID	Identifier of the locked out user
Timestamp	Time stamp of the lockout
Component/Application Name	Component from which the user has been locked out
Client IP Address	IP address of the client
Authentication Method	Authentication method
Message Event Details	Message regarding the failed authentication
Authorization_Failures	Authorization failure

10.4.3.6 Multiple Logins From Same IP

The Multiple Logins From Same IP report contains details regarding multiple logins from the same IP address.

Table 10-10 Multiple Logins From Same IP Report Fields

Field	Description
IP Address	IP address

Table 10-10 (Cont.) Multiple Logins From Same IP Report Fields

Field	Description
Username Used	Identifiers of users

10.5 About Creating Reports Using Third-Party Software

Access Manager supports the creation of reports by using third-party tools such as Crystal Reports.

To learn how to create reports by using third-party software, see the third-party software documentation. Additional information on the audit schema and creating custom reports can be found in the *Securing Applications with Oracle Platform Security Services*.

Monitoring Oracle Access Management Performance and Access Manager Health

Monitoring performance refers to observing (viewing) performance metrics to make yourself aware of the state of specific components of Oracle Access Management.

Monitoring the server health can be performed through heartbeat URL or Health Check Framework.

The heartbeat URL performs a set of predefined tests and returns only the https status code with no additional information.

The Health Check Framework supports additional information in the body of the http response. The information can indicate the nature of the test and the result of the test information. The tests performed can also be controlled either by configuration or request itself.

The tests currently supported depend exclusively on DMS metrics and log information.

This chapter contains the following sections on monitoring Oracle Access Management performance and Access Manager health.

- [Introduction to Performance Monitoring](#)
- [Monitoring Server Metrics Using Oracle Access Management Console](#)
- [OAM Proxy Metrics and Tuning](#)
- [Monitoring Metrics Using the DMS Console](#)
- [Monitoring the Health of an Access Manager Server](#)
- [Monitoring Server Health with Health Check Framework](#)

In addition to these, the admin server exposes DMS metrics in JMX. Refer to the `displayOAMMetrics WLST` command.

See Also:

- [Monitoring Performance and Logs with Fusion Middleware Control](#) if you are using Oracle Enterprise Manager Fusion Middleware Control

11.1 Introduction to Performance Monitoring

Component performance metrics is collected in memory during the completion of particular events. These metrics are kept only in memory so there are several mechanisms to extract and display them including (but not limited to) Oracle Enterprise Manager Fusion Middleware Control (FMW), the Oracle Dynamic Monitoring Service (DMS) and the Oracle Process Manager and Notification Server (OPMN).

- FMW Control is a Web browser-based, graphical user interface that offers monitoring options. See [Monitoring Performance and Logs with Fusion Middleware Control](#) for details.
- DMS uses the DMS Spy Servlet to provide access to DMS metric data from a web browser. Information is categorized by Noun Types; for Oracle Access Management the prefix is OAMS.OAM_. See [Monitoring Metrics Using the DMS Console](#).
- dmsdump is provided by DMS to take metrics from the servers based on definitions in a dms configuration file. There are many OAM metrics exposed when dms dumps are generated. See About Dynamic Monitoring Service (DMS) in the *Oracle Fusion Middleware Performance and Tuning Guide*.
- OPMN provides access to metrics using dmsdump.

11.2 Monitoring Server Metrics Using Oracle Access Management Console

Users with valid Oracle Access Management Administrator credentials can log into the Oracle Access Management Console and monitor various performance metrics.

This section provides the following topics:

- [Monitoring Server Instance Performance](#)
- [Oracle Access Manager Server Metrics](#)

11.2.1 Monitoring Server Instance Performance

Users with valid Oracle Access Management Administrator credentials can monitor performance for Access Manager using the Monitoring command on the Actions menu under the System Configuration tab using the Oracle Access Management Console.

See [Understanding the Oracle Access Management Console](#) for details.

Before you begin, the OAM Server must be running.

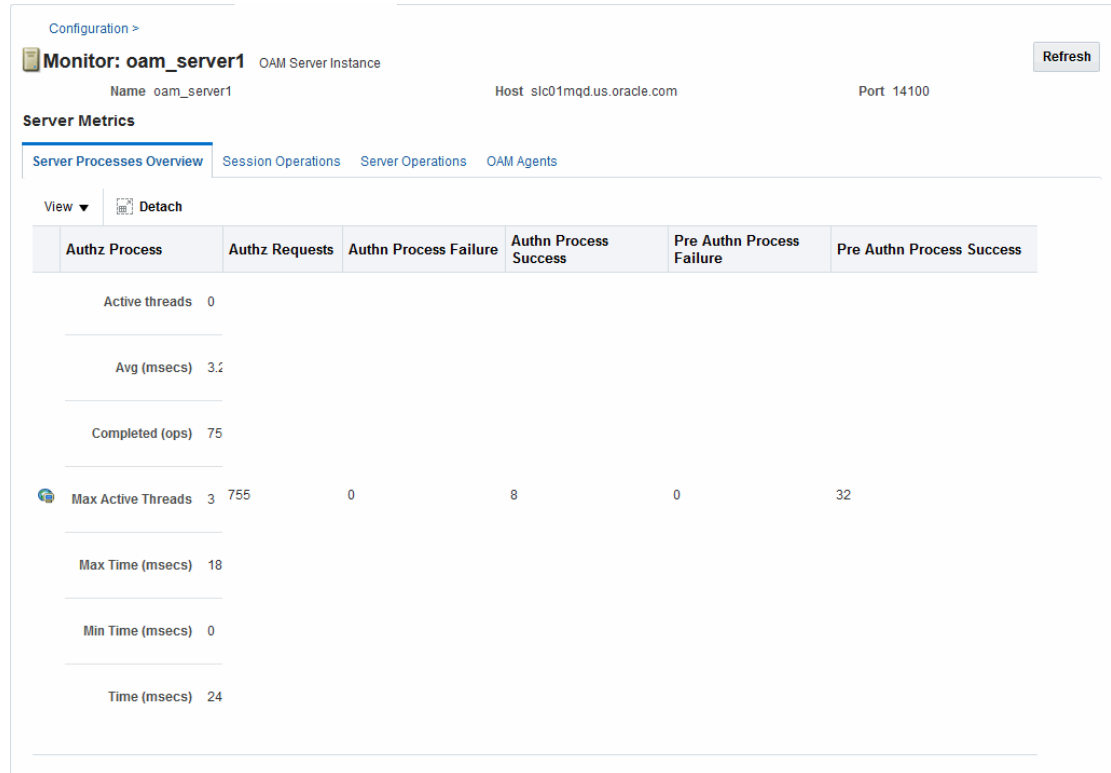
1. From the Oracle Access Management Console, click Server Instances and the desired server instance.
2. **Server Instance:**
 - a. From the **Actions** menu in the navigation tree, click **Monitor Menu**.
 - b. On the Monitor page, click the desired subtab to view results for the server instance:
 - Server Processes Overview
 - Session Operations
 - Server Operations
 - WebGates
 - c. Proceed to [Oracle Access Manager Server Metrics](#)
3. See also, "[OAM Proxy Metrics and Tuning](#)".

11.2.2 Oracle Access Manager Server Metrics

This topic provides a look at the Server metrics available through the **Monitor** option from the **Server Instances** tab in the **Configuration** section of the console.

Figure 11-1 shows the Server Processes page.

Figure 11-1 Server Processes Overview Page



Server Processes Overview provides the following OAM Server events, organized in individual columns on the tab.

The following are the server metric columns in the Server Process Overview Tab:

- Authorization Process
- Authorization Requests
- Authentication Process Failure
- Authentication Process Success
- Pre Authentication Process Failure
- Pre Authentication Process Success

Figure 11-2 shows the Session Operations tab.

Figure 11-2 OAM Server Metrics: Session Operations Monitoring Page

Server Processes Overview **Session Operations** Server Operations OAM Agents

View ▾ Detach

	Check Session Valid	Create Session	Destroy Session	Delete Client Session
Active threads	0	0	0	0
Avg (msecs)	1.044	560.0	0.0	0.0
Completed (ops)	1617	10	0	0
Max Active Threads	5	1	0	0
Max Time (msecs)	195	1405	0	0
Min Time (msecs)	0	17	0	0
Time (msecs)	1689	5600	0	0

Columns Hidden 8

OAM Server Session Operations metrics include:

- Check Session Valid
- Check Session Valid Failure
- Check Session Valid Success
- Create Session
- Create Session Failure
- Create Session Success
- Destroy Session
- Destroy Session Failure
- Destroy Session Success
- Delete Client Session
- Delete Client Session Failure

Figure 11-3 shows the Server Operations tab.

Figure 11-3 OAM Server Metrics: Server Operations Tab

Server Processes Overview Session Operations **Server Operations** OAM Agents

View ▾ Detach

Auth Policy Response Success	Auth Scheme Response Success	Authn Policy Response	Authz	Is Resource Protected
		Active threads 0	Active threads 0	Active threads 0
		Avg (msecs) 0.07644882860665844	Avg (msecs) 3.2386363636363636	Avg (msecs) 0.951073985
		Completed (ops) 811	Completed (ops) 792	Completed (ops) 838
811	811	Max Active Threads 2	Max Active Threads 3	Max Active Threads 2
		Max Time (msecs) 9	Max Time (msecs) 183	Max Time (msecs) 202
		Min Time (msecs) 0	Min Time (msecs) 0	Min Time (msecs) 0
		Time (msecs) 62	Time (msecs) 2565	Time (msecs) 797

Columns Hidden 11

The following are the OAM Server Operations metrics in the Server Operations Tab:

- Authentication Policy Response Failure
- Authentication Policy Response Success
- Authentication Scheme Response Failure
- Authentication Scheme Response Success
- Authentication Failure
- Authentication Failure Responses
- Authentication Policy Response
- Authentication Requests
- Authentication Scheme Response
- Authorization Failure
- Authorization Failure
- Authorization Process Failure
- Authorization Process Success

Figure 11-4 shows the OAM Server Metrics: WebGates tab with all available metrics showing.

Figure 11-4 OAM Server Metrics: WebGates Tab

Agent Name	Agent Status	Version
Agent_IAMSuiteAgent	Connected	10.x

WebGate performance metrics include:

- Agent Name
- Agent Status
- Version

11.3 OAM Proxy Metrics and Tuning

Administrators can tune the performance of OAM proxy through the Java EE container Administration Console.

This section provides the following topics:

- [OAM Proxy Metrics](#)
- [OAM Proxy Server Tuning Parameters](#)

See Also:

- [Tuning Performance](#)

11.3.1 OAM Proxy Metrics

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 11-1](#) lists the various OAM Proxy metrics available.

Table 11-1 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

Table 11-1 (Cont.) OAM Proxy Metrics

Metric	Description
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

11.3.2 OAM Proxy Server 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 Oracle Access Management Administrator can tune performance using the Java EE container Administration Console, which is outside the scope of this book.

[Table 11-2](#) provides the tuning parameters for the OAM Proxy.

Table 11-2 OAM Proxy Tuning Parameters

Purpose	Parameter	Type	Value	Description
Denial of Service Attacks	ConnectionValidationInterval	Integer	120	The time interval in seconds for validating the connections periodically for denial of service attacks
Denial of Service Attacks	BacklogQueue	Integer	50	Maximum length of backlog queue
Denial of Service Attacks	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

11.4 Monitoring Metrics Using the DMS Console

Oracle Access Management uses the Oracle Dynamic Monitoring Systems (DMS) to measure application-specific performance information for OAM Servers and registered Agents. The metrics can be used to monitor the time spent in a particular area, or track particular occurrences or state changes.

To access the DMS console, type the following URL in a browser window and log in with your Oracle Access Management Administrator credentials.

`http:// <example_AdminServer:Port>/dms/Spy`

Once logged into the DMS console you can monitor metrics as discussed in the following sections.

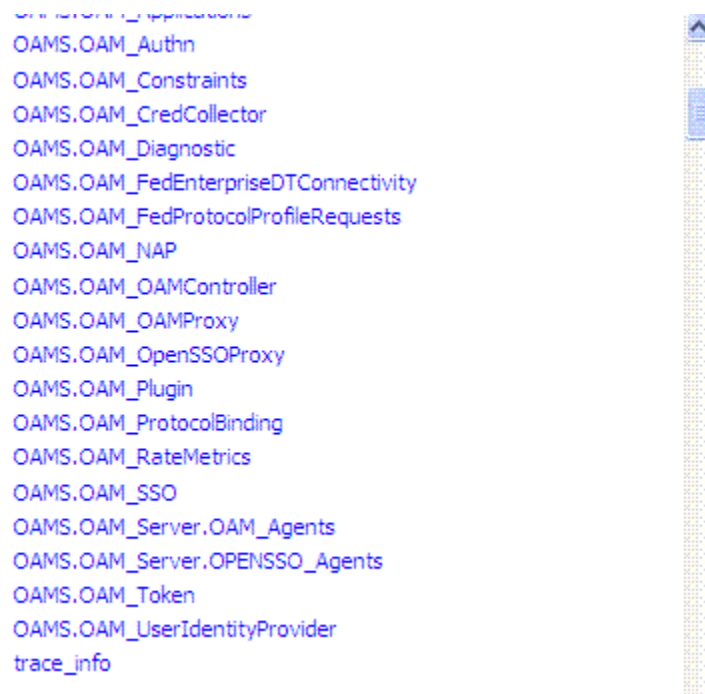
- [Monitoring OAM Metrics](#)

11.4.1 Monitoring OAM Metrics

The OAM metrics can be reviewed in the DMS Metric Tables panel.

You can access metrics regarding OAM as illustrated in [Figure 11-5](#). Click the desired metric from those listed to view the results on the right-side of the console.

Figure 11-5 OAM Metrics Table



11.5 Monitoring the Health of an Access Manager Server

Access Manager Services are business critical and must always be available to control user access to an organization's protected web services and applications. Because

hardware, network connectivity issues and other failures can happen, HeartBeat monitoring can be leveraged by Load Balancers to ensure user traffic is routed to healthy OAM Servers.

For example, when there is a firewall installed between a User Agent or WebGate and the Access Manager server, perimeter devices can check availability of the Access Manager server (its *health*) by hitting its HeartBeat URL. The following sections contain details.

- [Understanding WebGate and Access Manager Communications](#)
- [Monitoring Access Manager Server Health](#)

11.5.1 Understanding WebGate and Access Manager Communications

When deploying a network firewall between a WebGate and Access Manager server, the WebGate communicates using the OAP protocol by creating a TCP socket connection with Access Manager to establish a message channel. The WebGate uses the message channel to send different OAP messages necessary to serve the resource requests (isprotected, isauthorized, and the like). Now, consider a situation in which the WebGate/Oracle HTTP Server is idle. In this case, the WebGate has received no resource request and will not send any messages to Access Manager for authentication or authorization; there will also not be any read/write activity on the socket connection.

The firewall determines this connection is *idle* after 30-40 minutes of inactivity (depending on its configuration) and terminates the socket connection but does not inform/notify the WebGate or Access Manager server. In this case, when a request for a resource arrives at the WebGate and it sends a OAP message to the Access Manager server, it uses the existing connection and waits for a reply. Because the connection was dropped by the firewall, the WebGate does not receive any reply; so it waits for the TCP timeout. Following the TCP timeout, WebGate understands the message channel is of no use and starts the process to get a new message channel. TCP timeout is OS specific and may vary from several minutes to hours which makes the WebGate unable to process user requests.

Note:

The `setKeepAlive` WebGate parameter ensures that load balancers do not drop the OAP connection. See [Table 15-2](#) for details.

11.5.2 Monitoring Access Manager Server Health

The OAM monitoring model allows Web Tier components (load balancers) to ping an OAM Managed Server's HeartBeat endpoint at a scheduled interval over HTTP(S). This allows Web Tier components to route incoming HTTP traffic away from unhealthy OAM Managed Server(s).

Every OAM Managed Server exposes this HeartBeat URL:

```
Scheme://ManagedServerHost:ManagedServerPort/oam/server/HeartBeat
```

In this URL, the following is true:

- `scheme` = `https` | `http`
- `ManagedServerHost` = Host name of the Access Manager WLS Managed Server
- `ManagedServerPort` = Port used by the Access Manager WLS Managed Server

The HeartBeat URL works as follows:

1. The Web Tier components will send an HTTP request to the HeartBeat endpoint of the Access Manager Managed Server.
2. The Access Manager Managed Server will then do the following:
 - Verify Id Store Connectivity
 - Verify Policy Store Connectivity
 - Verify the Credential Collector URLs are reachable
 - Sanity check the working of the Coherence Layer
 - Check for NAP connectivity

If the above tests succeed, the Access Manager server is considered to be healthy and a HTTP 200 response is sent to the Load Balancer. Any other HTTP Status Code value signifies that the Access Manager Managed Server is not healthy.

3. When multiple Access Manager Managed Servers are present in the deployment, the Web Tier component will repeat this for each OAM Managed Server.

**Note:**

Neither the health status test results or check results can be communicated in the body of the HTTP Response. A successful heartbeat check will return the HTTP code 200.

**WARNING:**

OAM Server health check raises a WARNING on the weblogic admin console for OAM if the server is configured to have a maximum heap size less than **1.5 GB**.

11.6 Monitoring Server Health with Health Check Framework

HealthCheck Framework enables health check on servers.

The following sections provides information on the health checks

- [Introduction to HealthCheck Framework](#)
- [Understanding HealthCheck Test Configuration](#)
- [Running Health Checks Using REST API](#)
- [Configuring Scheduled Health Checks](#)
- [Using the Health Script Evaluator](#)

11.6.1 Introduction to HealthCheck Framework

HealthCheck Framework enables health check on servers. These checks can be performed using REST API or by scheduling periodic checks on the server. Each schedule can be associated with a specified set of tests to be run.

The REST API invocation performs preconfigured health-check tests on the server and returns the status of the test runs.

The framework supports notification of issues by components through the `/health/check` API. If the test lists are not specified in the request, the results of a default set of tests are returned.

The framework provides aggregating services to health check tests. These aggregating services allows the tests to cumulate results over a configured window of time.

The test results are mapped to actions to be performed on the server. These actions are based on the test and the health result. The action and the mapping is configured in the `oam-config.xml` file.

These actions can include subscribing to the Weblogic Health Check callback and setting the Weblogic server state appropriately. For more information, see *Configure server health monitoring in Oracle® Fusion Middleware Administering Oracle WebLogic Server with Fusion Middleware Control*

11.6.2 Understanding HealthCheck Test Configuration

The HealthCheck Framework provides preconfigured health-check tests that are run when the health check API is invoked or during a scheduled Health Check.

The HealthCheck tests are configured under the `TestList` setting under the `HealthCheck` element in the `oam-config.xml` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration xsd:schemaLocation="http://higgins.eclipse.org/sts/
Configuration Configuration.xsd" Path="/DeployedComponent/Server/NGAMServer/
Profile/HealthCheck">
  <Setting Name="HealthCheck" Type="htf:map">
    <Setting Name="TestLists" Type="htf:list">
      <Setting Name="0" Type="hcf:testList">
        <Setting Name="Id" Type="xsd:string">TL001</Setting>
        <Setting Name="Name" Type="xsd:string">TL001</Setting>
        <Setting Name="Lang" Type="xsd:string">EN</Setting>
        <Setting Name="Validity" Type="xsd:duration">PT5M</Setting>
        <Setting Name="TestList" Type="htf:list">
          <Setting Name="0" Type="xsd:string">HeapSizeCheck</Setting>
          <Setting Name="1" Type="xsd:string">FreeHeapCheck</Setting>
          <Setting Name="2" Type="xsd:string">LoginFailureCheck</Setting>
          <Setting Name="3" Type="xsd:string">DirectoryOutage</Setting>
          <Setting Name="4" Type="xsd:string">DirectoryLatency</Setting>
          <Setting Name="5" Type="xsd:string">AuthenticationLatency</Setting>
          <Setting Name="6" Type="xsd:string">AuthorizationLatency</Setting>
        </Setting>
      </Setting>
    </Setting>
  </Setting>
```

```

<Setting Name="Schedules" Type="htf:list">
  <Setting Name="0" Type="hcf:schedule">
    <Setting Name="Id" Type="xsd:string">TS001</Setting>
    <Setting Name="Name" Type="xsd:string">TS001</Setting>
    <Setting Name="Desc" Type="xsd:string">Default schedule. Runs
every minute.</Setting>
    <Setting Name="Lang" Type="xsd:string">EN</Setting>
    <Setting Name="Cron" Type="xsd:string">* * * * *</Setting>
    <Setting Name="Enabled" Type="xsd:boolean">>true</Setting>
    <Setting Name="TestListId" Type="xsd:string">TL001</Setting>
  </Setting>
</Setting>
<Setting Name="ComponentTests" Type="htf:list">
  <Setting Name="0" Type="hcf:compTest">
</Setting>
  <Setting Name="1" Type="hcf:compTest">
</Setting>
  <Setting Name="2" Type="hcf:compTest">
</Setting>
  <Setting Name="3" Type="hcf:compTest">
    <Setting Name="Id" Type="xsd:string">DirectoryOutage</Setting>
    <Setting Name="Name" Type="xsd:string">DirectoryOutage</
Setting>
    <Setting Name="Lang" Type="xsd:string">EN</Setting>
    <Setting Name="Criticality"
Type="hcf:criticality">AUXILIARY </Setting>
    <Setting Name="Timeout"
Type="xsd:duration">P0Y0M0DT0H0M1.000S</Setting>
    <Setting Name="Class"
Type="xsd:string">oracle.security.am.healthcheck.featuretest.dms.DmsMet
ricsDrivenChecks</Setting>
    <Setting Name="Parameters" Type="htf:list">
      <Setting Name="0" Type="xsd:string">/*Directory outages are
detected based on LIBOVD-40067 messages that are issued every minute.*//
refid =
LogsUtil.recordLogMessage("oracle.ods.virtualization.engine.backend.jnd
i.adapter1", "LIBOVD-40067");windowSize =
LogsUtil.getLogOccurrencesWindow(refid);if (windowSize <> 180000.0)
{ LogsUtil.setLogOccurrencesWindow(refid, 180000.0); windowSize =
LogsUtil.getLogOccurrencesWindow(refid);};count =
LogsUtil.getLogOccurrences(refid);ScriptUtil.removeVariable("refid");co
unt < 0.5;</Setting>
    </Setting>
  </Setting>
  <Setting Name="4" Type="hcf:compTest">
</Setting>
  <Setting Name="5" Type="hcf:compTest">
</Setting>
  <Setting Name="6" Type="hcf:compTest">
</Setting>
</Setting>
</Setting>
</Configuration>

```

You can use the `GET` and `PUT` methods of the `/iam/admin/config/api/v1/config` API to fetch and update the configuration. For more information, see [Configuring Scheduled Health Checks](#).

HealthCheck Test	Description
HeapSizeCheck	Helps identify the misconfigured servers. DMS metrics are invoked to determine the heap size. If the heapsize is less than the configured value, the test is considered failed and the server is put in warning state.
FreeHeapCheck	Helps determine if the server is a candidate for throttling. DMS metrics are invoked to determine free heap. If the ratio of the free heap to the max heap, in percentage, is less than the value of the threshold variable <code>freeThreshold</code> , the test is considered failed and the server is put in warning state.
LoginFailureCheck	Helps determine user login failure rate.
DirectoryOutage	The server tests for directory state every minute. If there is an outage, log messages are generated. Such log messages, for example, LIBOVD-40067 are detected by this test, resulting in test failure, and the server is put on failed state.
DirectoryLatency	Directory latency increase is detected on current samples in the configured sampling window. If the latency is greater than the configured allowed value, the test is considered failed, and the server is put in warning state.
AuthenticationLatency	Authentication latency increase is detected on current samples in the configured sampling window. If latency is greater than the configured allowed value, the test is considered failed, and the server is put in warning state.
AuthorizationLatency	Authorization latency increase is detected on current samples in the configured sampling window. If latency is greater than the configured allowed value, the test is considered failed, and the server is put in warning state.

The HealthCheck Framework also provides a Health Script Evaluator tool for creating your own tests. For more information, see [Using the Health Script Evaluator](#).

11.6.3 Running Health Checks Using REST API

You can use the `/health/check` REST API to run the preconfigured tests on the servers.

Run the following REST API command to perform health check on the servers:

```
curl -X GET --header 'Authorization:'
--header 'Accept: application/json'
'http://<ManagedServerHost>:<ManagedServerPort>/health/check'
-d {report: summary, testlistid}
```

The following table provides the parameter details:

Table 11-3 Health Check REST API Parameters

Parameters	Description
Report	Determines the content in the response. Following values are supported. <ul style="list-style-type: none"> summary – A summary report of each tested component is sent back in the response. details – A detailed report of each tested component is sent back in the response.
testlistid	Optional. Specifies the collection of tests to run. If this parameter is not provided then all the tests that are listed in the <code>testList</code> element with id <code>restTestListId</code> in the <code>oam-config.xml</code> file are run.

11.6.4 Configuring Scheduled Health Checks

The HealthCheck framework allows scheduled health checks on server. A collection of tests associated with the specified schedule is run periodically. Multiple schedules can run the configured tests.

The periodic health checks are performed based on the parameters and values configured under the `HealthCheck` element in the `oam-config.xml` file.

To create a schedule for the specified set of tests, follow the steps as described:

1. Fetch the configuration settings using the admin config API. Specify the path to the `HealthCheck` setting, for example:

```
http://<AdminServerHost>:<AdminServerPort>/iam/admin/config/api/v1/
config
?path=/DeployedComponent/Server/NGAMServer/Profile/HealthCheck
```

2. Add the tests that are required to be run to the `TestLists` setting as shown.

```
<Setting Name="TestLists" Type="htf:list"
Path="/DeployedComponent/Server/NGAMServer/Profile/HealthCheck/
TestLists">
<Setting Name="0" Type="hcf:testList">
<Setting Name="Id" Type="xsd:string">TL001</Setting>
<Setting Name="Name" Type="xsd:string">TL001</Setting>
<Setting Name="Lang" Type="xsd:string">EN</Setting>
<Setting Name="Validity" Type="xsd:duration">-PT5M</Setting>
<Setting Name="TestList" Type="htf:list">
<Setting Name="0" Type="xsd:string">HeapSizeCheck</Setting>
<Setting Name="1" Type="xsd:string">FreeHeapCheck</Setting>
<Setting Name="2" Type="xsd:string">LoginFailureCheck</Setting>
<Setting Name="3" Type="xsd:string">DirectoryOutage</Setting>
<Setting Name="4" Type="xsd:string">DirectoryLatency</Setting>
<Setting Name="5" Type="xsd:string">AuthenticationLatency</Setting>
```



```

<Setting Name="6" Type="xsd:string">AuthorizationLatency</Setting>
</Setting>
</Setting>
<Setting Name="1" Type="hcf:testList">
<Setting Name="Id" Type="xsd:string">LoginTestList</Setting>
<Setting Name="Name" Type="xsd:string">LoginTestList</Setting>
<Setting Name="Lang" Type="xsd:string">EN</Setting>
<Setting Name="Validity" Type="xsd:duration">-PT5M</Setting>
<Setting Name="TestList" Type="htf:list">
<Setting Name="0" Type="xsd:string">LoginFailureCheck</Setting>
</Setting>
</Setting>
<Setting Name="2" Type="hcf:testList">
<Setting Name="Id" Type="xsd:string">LDAPOutageTestList</Setting>
<Setting Name="Name" Type="xsd:string">LDAPOutageTestList</Setting>
<Setting Name="Lang" Type="xsd:string">EN</Setting>
<Setting Name="Validity" Type="xsd:duration">-PT5M</Setting>
<Setting Name="TestList" Type="htf:list">
<Setting Name="0" Type="xsd:string">DirectoryOutage</Setting>
</Setting>
</Setting>
<Setting Name="3" Type="hcf:testList">
<Setting Name="Id" Type="xsd:string">LatencyTestList</Setting>
<Setting Name="Name" Type="xsd:string">LatencyTestList</Setting>
<Setting Name="Lang" Type="xsd:string">EN</Setting>
<Setting Name="Validity" Type="xsd:duration">-PT5M</Setting>
<Setting Name="TestList" Type="htf:list">
<Setting Name="0" Type="xsd:string">DirectoryLatency</Setting>
<Setting Name="1" Type="xsd:string">AuthenticationLatency</Setting>
<Setting Name="2" Type="xsd:string">AuthorizationLatency</Setting>
</Setting>
</Setting>
</Setting>

```

3. Add a schedule, for the tests to be run, under the Schedules setting. For example, to schedule the test LoginTestJob every two minutes and LDAPOUTAGEJob every 10 minutes, set the parameters as shown:

```

<Setting Name="Schedules" Type="htf:list" Path="/DeployedComponent/Server/
NGAMServer/Profile/HealthCheck/Schedules">
  <Setting Name="1" Type="hcf:schedule">
    <Setting Name="Id" Type="xsd:string">LoginTestJob</
Setting>
    <Setting Name="Cron" Type="xsd:string">*/2 * * * *</
Setting>
    <Setting Name="Enabled" Type="xsd:boolean">>true</Setting>
    <Setting Name="TestListId" Type="xsd:string">LoginTest</
Setting>
  </Setting>
  <Setting Name="2" Type="hcf:schedule">
    <Setting Name="Id" Type="xsd:string">LDAPOUTAGEJob</
Setting>
    <Setting Name="Cron" Type="xsd:string">*/10 * * * *</
Setting>

```

```

        <Setting Name="Enabled" Type="xsd:boolean">true</
Setting>
        <Setting Name="TestListId"
Type="xsd:string">LDAPOUTAGE</Setting>
    </Setting>
</Setting>

```

4. Update the configuration using the `PUT` method of the admin config API as shown:

```

curl -u username:password -H
'Content-Type: text/xml'
-X PUT http://<AdminServerHost>:<AdminServerPort>/iam/admin/
config/api/v1/config
--data-binary @ConfigFile

```

11.6.5 Using the Health Script Evaluator

Health Check Framework includes a script evaluator tool, using which you can create health-check tests based on DMS metrics, and evaluate them within the tool.

The evaluator tool provides utility *functions* to extract values from the DMS sensors, and *variables* for capturing those values. The tool also supports branching along with arithmetic and boolean expressions for evaluating your tests.

The evaluator tool consists of a text area where you can input your scripts for evaluating. The results of the evaluation are displayed in the bottom panels.

Interim variables that are created during evaluation and the values after complete execution are displayed in the **Variables Created** panel.

The final value is displayed in the **Exceptions and Returned Values** panel.



Note:

The scripts must be created to return a boolean value indicating the result of the test.

If a function is passed an invalid value, the returned exception may contain possible values for the parameter in the **Exceptions and Returned Values** panel.

You can access the health script evaluator from the following link:

```

http://<ManagedServerHost>:<ManagedServerPort>/iam/access/api/v1/
health/script/evaluate

```

Click **Help** in the evaluator tool for information about the syntax and details regarding the Functions, Branching, Variables and Expression that the tool supports. The **Help** also provides snippets of code that gets copied directly into the text area when you click them.

For information about all the available operations on the tool, refer to `http://<ManagedServerHost>:<ManagedServerPort>/iam/access/api/v1/health/dms/info`

You can use the sensors information and metrics listed in the following link to get the values for parameters added to the functions: `http://<ManagedServerHost>:<ManagedServerPort>/iam/access/api/v1/health/dms/info?operation=sensors`

 **Note:**

After creating and evaluating your health-check scripts in the evaluator tool, you can also add your tests to the configuration file for scheduled checks. See [Configuring Scheduled Health Checks](#) for details.

Building a Simple Health Check Test in the Evaluator Tool

This example helps identify misconfigured servers with heap size less than 1500000KB. Form the script as explained in the following steps and add it to the evaluator tool:

1. To retrieve the heap size configured on the server, use the Sensor function `DmsUtil.getMetricValue("type", "noun", "path", "sensorName", "metric")` and assign its value to `maxHeap` variable.
Refer to `http://<ManagedServerHost>:<ManagedServerPort>/iam/access/api/v1/health/dms/info?operation=sensors` to get the required details for the parameters of the function. In this example, the `type` is `JVM_MemorySet`, and `noun` `Heap memory`, `path` is `Path:/JVM/MxBeans/memory/type/Heap memory` `sensorName` is `max` and `metric` can be `value`.
2. Retrieve the units value using `DmsUtil.getMetricUnits(sensor, "metric")` and assign it to `units` variable .
3. Set a variable `heapThreshold = 1500000`.
4. Compare the `maxheap` with `heapThreshold` using the boolean `>` operator.

```
// Threshold determines free memory in percentage that triggers server
criticality.
heapThreshold = 1500000;

maxHeap = DmsUtil.getMetricValue("JVM_MemorySet", "Heap memory", "/JVM/
MxBeans/memory/type/Heap memory", "max", "value");
units = DmsUtil.getMetricUnits(DmsUtil.locateSensor("JVM_MemorySet", "Heap
memory",
    "/JVM/MxBeans/memory/type/Heap memory", "max"), "value");

maxHeap > heapThreshold;
```

Click **Evaluate**. The tool returns the following response:

Sample Response

```
Variables Created
{heapThreshold=1500000.0, maxHeap=1864192, units=KB}
Exceptions and Returned Values
result:true
```

Example Test Scripts

The following examples provide additional health test scripts showing the various functions that the evaluator tool supports:

Example 11-1 User Login Failure

The following example is a script, testing the health of the system. It measures the number of user login failures within the specified window of time and returns results. Login failures above 90% of the user login, within two minutes and thirty seconds suggests problems with the system that might need immediate attention.

This script uses a Math function `MathUtil.doCount()` for counting the number of user authentication requests and failures. See Help in the evaluator tool for more information about Math functions.

```

/*
User Login Failure
*/
//Number of User login requests, when the test becomes effective
requestsThreshold = 100;
//Number of user login failures, in percentage
failedThreshold = 90;
//Free percentage averaged over two minutes and 30 seconds window
reqWinSize = 150000; //2.5 * 60 * 1000;

windowSize = MathUtil.getAveWindowSize("LoginFailure","failed");
if (windowSize > reqWinSize) {
    MathUtil.setAveWindowSize("LoginFailure","failed", reqWinSize);
    windowSize = MathUtil.getAveWindowSize("LoginFailure","failed");
};

windowSize = MathUtil.getAveWindowSize("LoginFailure","users");
if (windowSize < reqWinSize) {
    MathUtil.setAveWindowSize("LoginFailure","users", reqWinSize);
    windowSize = MathUtil.getAveWindowSize("LoginFailure","users");
};
//Count of user authentications
MathUtil.doCount("LoginFailure","failed",DmsUtil.getMetricValue("OAMS.OA
M_UserIdentityProvider", "UserIdentityProvider", "/OAMS/OAM/
UserIdentityProvider", "authenticateUserFailure", "count"));
failed = MathUtil.getCount("LoginFailure","failed");

//Count of user authentication requests
MathUtil.doCount("LoginFailure","users",DmsUtil.getMetricValue("OAMS.OA
M_UserIdentityProvider", "UserIdentityProvider", "/OAMS/OAM/
UserIdentityProvider", "authenticateUser", "count"));

//The computed value of the count from MathUtil.doCount retrieved and
assigned to this variable. This ensures the script does not hang, if
the count takes longer duration to compute.
requests = MathUtil.getCount("LoginFailure","users");

//Always returns true if the number of requests are less than 100.
result = requests < requestsThreshold || failed/requests*100 <

```

```

failedThreshold;
if (requests > requestsThreshold) {
    ScriptUtil.removeVariable("requestsThreshold");
};
ScriptUtil.removeVariable("reqWinSize");
result;

```

Sample Response

```

Variables Created
{failed=0.0, failedThreshold=90.0, requests=0.0, requestsThreshold=100.0,
result=true, windowSize=150000.0}
Exceptions and Returned Values
result:true

```

Example 11-2 Directory Outage

The following example counts the log messages for a specified threshold window and can be used to determine if there has been directory outages. The result can further be used to generate a warning on the **Health Monitoring** in the Admin Server.

This script uses Log functions `LogsUtil.recordLogMessage()` and `LogsUtil.setLogOccurrencesWindow()`. It also uses `ScriptUtil.removeVariable()` function to remove the intermediate variable `refid`. See **Help** in the evaluator tool for more information about the Log functions.

```

/*
Directory outages are detected based on LIBOVD-40067 messages that are
issued every minute.
*/
//set a reference to the log message when it happens
refid =
LogsUtil.recordLogMessage("oracle.ods.virtualization.engine.backend.jndi.adap
ter1", "LIBOVD-40067");
//set a window size, using the reference id of the log message
windowSize = LogsUtil.getLogOccurrencesWindow(refid);
if (windowSize > 180000.0) {
    LogsUtil.setLogOccurrencesWindow(refid, 180000.0);
    windowSize = LogsUtil.getLogOccurrencesWindow(refid);
};

//count the number of log occurrences
count = LogsUtil.getLogOccurrences(refid);
ScriptUtil.removeVariable("refid");
count < 0.5;

```

Sample Response

```

Variables Created
{count=0.0, windowSize=180000.0}
Exceptions and Returned Values
result:true

```

12

Monitoring Performance and Logs with Fusion Middleware Control

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

You can monitor performance and log messages for Access Manager 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 Overview of Oracle Fusion Middleware Administration Tools in *Administering Oracle Fusion Middleware*.

Note:

Unless explicitly stated, information in this chapter is the same for both services. There are no metrics in Oracle Fusion Middleware Control for Identity Federation.

This chapter includes the following topics.

- [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](#)
- [Managing Log File Configuration from Fusion Middleware Control](#)
- [Viewing Log Messages in Fusion Middleware Control](#)
- [Displaying MBeans in Fusion Middleware Control](#)

12.1 Introduction to Fusion Middleware Control

Within Fusion Middleware Control, information is updated dynamically during live sessions of Access Manager 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:

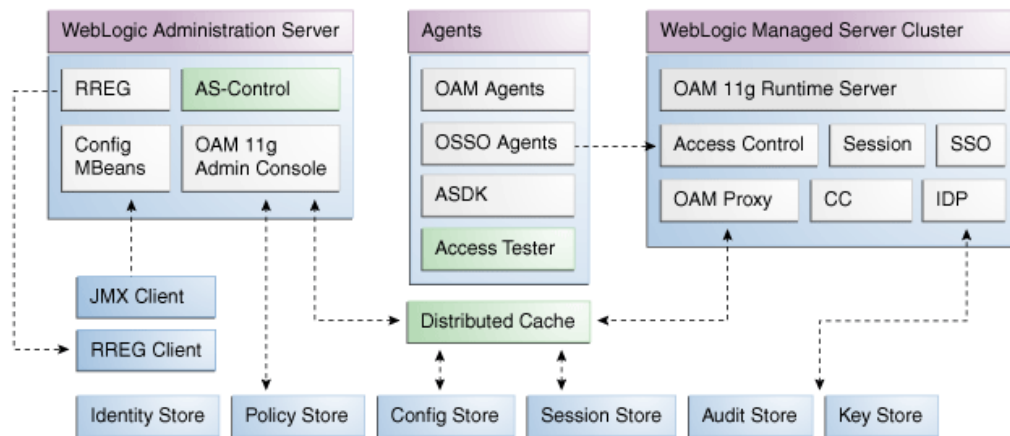
Enterprise Manager Grid Control is an independently licensed product that provides additional capabilities not found in Fusion Middleware Control (primarily, the ability to collect and maintain data for historical purposes and trending).

Oracle Access Management 12c is deployed as a Java EE application in a WebLogic container. For high availability and failover, Oracle Access Management 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 Access Manager and 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 Management on the WebLogic Administration Server, as illustrated in [Figure 12-1](#) (and described in the *Installing and Configuring Oracle Identity and Access Management*).

Figure 12-1 Fusion Middleware Control (AS-Control) Deployment Architecture



Using Fusion Middleware Control for targets is supported through the Oracle Dynamic Monitoring Systems instrumentation within Oracle Access Management. 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 Access Manager with Security Token Service

12.2 Logging In to and Out of 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. This section provides the following topics:

- [Logging In To Fusion Middleware Control](#)

- [Logging Out of Fusion Middleware Control](#)

12.2.1 Logging In To Fusion Middleware Control

Only Fusion Middleware Control Administrators log in to Fusion Middleware Control.

See Also:

Oracle Fusion Middleware Administrator's Guide for details about getting started using Fusion Middleware Control

1. In a browser window, enter the URL to Fusion Middleware Control. For example:
`http://host.example.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 Management, 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.

12.2.2 Logging Out of Fusion Middleware Control

You can 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.

12.3 Displaying Menus and Pages in Fusion Middleware Control

Fusion Middleware Control displays OAM Farm page and associated information, summary, and cluster or server pages.

This section provides the following topics for Access Manager and Security Token Service:

- [Farm Page in Fusion Middleware Control](#)
- [Context Menus and Pages in Fusion Middleware Control](#)
- [Displaying Context Menus and Target Details in Fusion Middleware Control](#)

See Also:

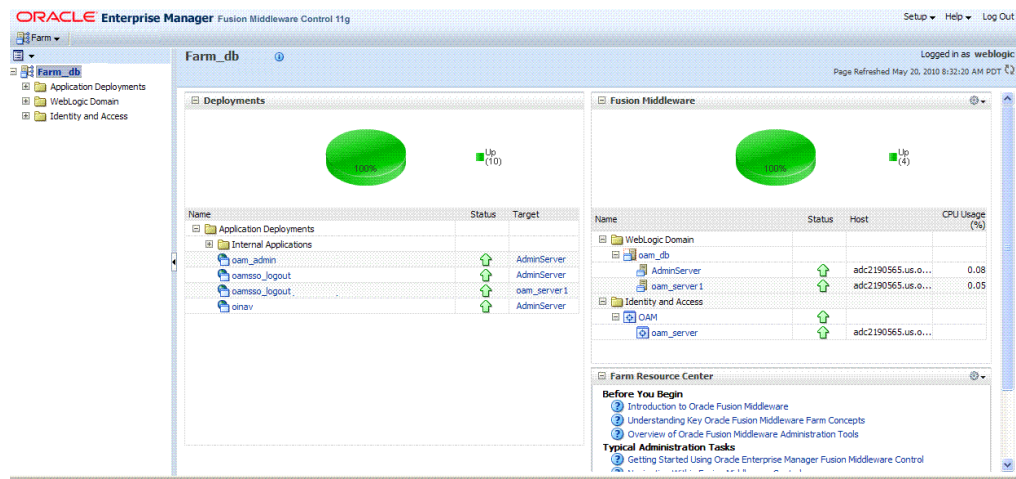
Oracle Fusion Middleware Administrator's Guide for details about getting started using Fusion Middleware Control

12.3.1 Farm Page in Fusion Middleware Control

Each OAM Farm page in Fusion Middleware Control includes similar information.

Figure 12-2 illustrates the OAM Farm page in Fusion Middleware Control. The Farm Resource Center provides immediate access to online information.

Figure 12-2 OAM Farm Page in Fusion Middleware Control

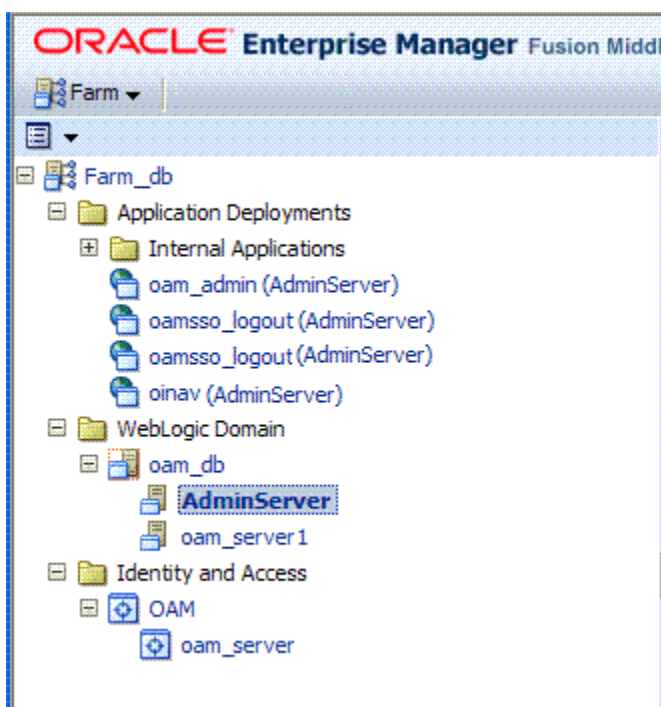


Sections on the Farm page are described in Table 12-1.

Table 12-1 Farm Page Sections

Farm Page Sections	Description
Deployments	<p>Within the farm, this section displays the Status and Target of each Internal Application within the Application Deployment.</p> <p>Clicking any link in the Deployments section (or in the navigation tree) displays a page containing more information.</p>
Fusion Middleware	<p>Within the farm, this section displays the status, host, and CPU usage for server instances in the:</p> <ul style="list-style-type: none"> • WebLogic Server domain • Identity and Access <p>Clicking any link on the page (or in the navigation tree) displays a page containing a more detailed summary.</p>
Farm Resource Center	<p>Provides a wealth of online information in the following categories:</p> <ul style="list-style-type: none"> • Information that is useful before you begin using Fusion Middleware Control • Administrator tasks using Fusion Middleware Control • Other resources <p>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.</p>

The navigation tree on the left side of the page, like the one in Figure 12-3, 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 12-3 Farm Navigation Tree in Fusion Middleware Control

For more information, see "[Logging In To Fusion Middleware Control](#)".



See Also:

"[Displaying Menus and Pages in Fusion Middleware Control](#) "

12.3.2 Context Menus and Pages in Fusion Middleware Control

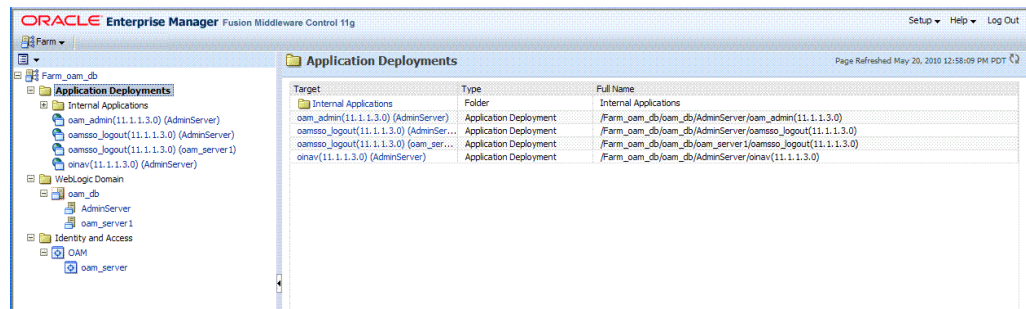
For Oracle Access Management, Farm details in Fusion Middleware Control are divided into four nodes within the navigation tree.

The nodes are:

- Application Deployments
- Internal Applications (includes logout page and other details for the OAM AdminServer and OAM Server instances)
- WebLogic Server domains (WebLogic Server details, including the OAM Farm)
- Identity and Access (includes OAM Cluster or individual OAM Server instances)

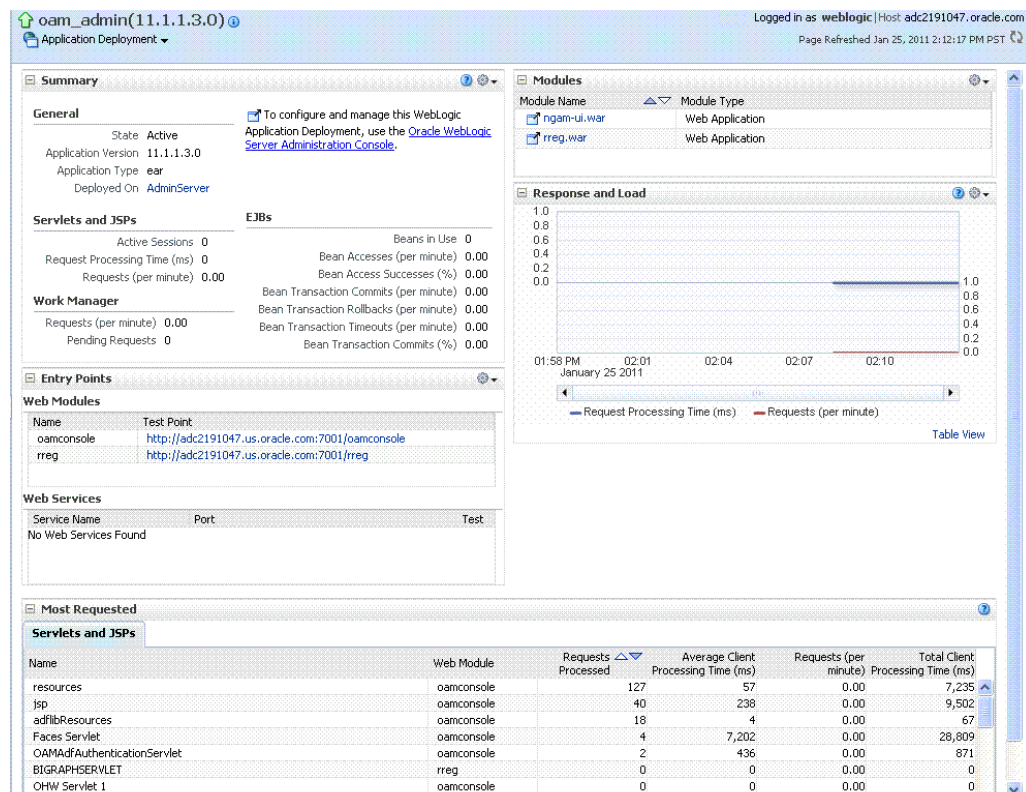
Clicking a node in the navigation tree displays an information page with individual links and a description of the Target, Type, and Full Name, as shown in [Figure 12-4](#) for Application Deployments.

Figure 12-4 Node Information Page in Fusion Middleware Control



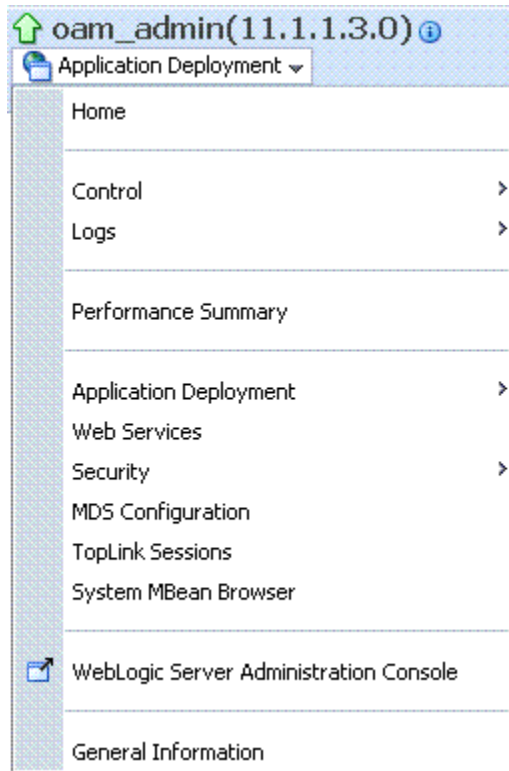
Clicking an instance (target) name (from either the navigation tree or a page), displays a context menu and a more detailed summary page. The Internal Application target is highlighted in the navigation tree and a page of the same name is displayed on the right. The context menu is available beneath the target name at the top of the page, as shown in Figure 12-5.

Figure 12-5 Application Deployment Summary for the Selected Internal Application



The Application Deployment menu is shown in Figure 12-6.

Figure 12-6 Application Deployment Menu



WebLogic Server domain: The WebLogic Server domain page is shown in [Figure 12-7](#) with the corresponding menu displayed. The Oracle WebLogic Server domain Resource Center, with links to online documentation, is visible in the bottom-left corner. This page more closely resembles the Farm landing page.

Figure 12-7 WebLogic Server Domain Summary with Context Menu Exposed

The screenshot shows the Fusion Middleware Control console interface. At the top, it indicates the user is logged in as 'weblogic' and the page was refreshed on Jan 25, 2011 2:28:20 PM PST. The main content area is titled 'base_domain' and 'WebLogic Domain'. A context menu is open over the 'oam_server1' instance, listing various management options. The background page shows a table of server instances and a 'Deployments' section with a table of application deployments.

Name	Listen Port	Active Sessions	Request Processing Time (ms)	Bea Accesses (per minute)
oam_server1	14100	0	0	6.1

Name	Status	Target
oam_admin(11.1.1.3.0)	Up	AdminServer
oamssso_logout(11.1.1.3.0)	Up	AdminServer
oamssso_logout(11.1.1.3.0)	Up	oam_server1

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 12-5](#).

For more information, see "[Displaying Context Menus and Target Details in Fusion Middleware Control](#)".

See Also:

"[Viewing Performance in Fusion Middleware Control](#)" for information about the Identity and Access node and related pages.

12.3.3 Displaying Context Menus and Target Details in Fusion Middleware Control

Fusion Middleware Control Administrators can view context menus and target pages.

Note:

From the Farm Resource Center on the OAM Farm page, choose topics of interest (or click Help in the upper-right corner of the page) to get more information.



See Also:

["Context Menus and Pages in Fusion Middleware Control"](#)

1. Log in as described in ["Logging In To Fusion Middleware Control"](#).
2. Expand the Farm containing Oracle Access Management, 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 12-5](#) and [Figure 12-6](#)).
5. **Cluster or Server Pages:** See ["Viewing Performance in Fusion Middleware Control"](#).

12.4 Viewing Performance in Fusion Middleware Control

Fusion Middleware Control provides various performance metrics for administrators.

The following sections provide more details:

- [Performance Overview Pages in Fusion Middleware Control](#)
- [Metrics Palette and the Performance Summary Page](#)
- [Displaying Performance Metrics in Fusion Middleware Control](#)
- [Displaying Component-Specific Performance Details](#)
- [Resulting Pages for Selected Nodes and Targets](#)

12.4.1 Resulting Pages for Selected Nodes and Targets

Using Fusion Middleware Control, you can view performance metrics for live sessions in a variety of formats.

Fusion Middleware Control provides Administrators with:

- A cluster-wide view of performance for Access Manager with Security Token Service
- A per-server drill-down of key performance metrics
- The ability to quickly add or remove performance metrics

[Table 12-2](#) summarizes the pages for selected nodes and target instances.

Table 12-2 Resulting Pages for Selected Nodes and Targets

Node	Target	Information Summary Page	Performance Overview	Performance Summary w/ Metrics
Application Deployment	...AdminServer	Yes	No	Yes
Internal Applications	oamsso_logout AdminServer	Yes	No	Yes
	oamsso_logout 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 (Cluster)	No	Yes	Yes
	oam_server (Server)	No	Yes	Yes



Note:

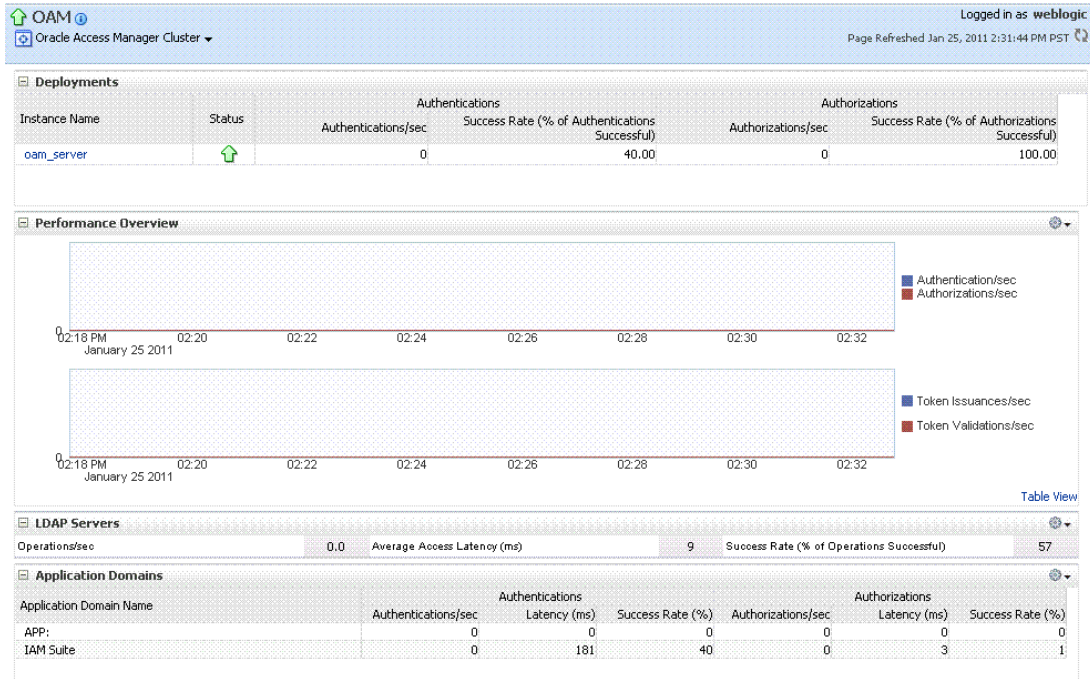
Security Token Service performance is included with relevant OAM Cluster and Server pages.

12.4.2 Performance Overview Pages in Fusion Middleware Control

The Fusion Middleware Control Performance Overview can be used to reflect WebLogic cluster information down to specific performance metrics for individual Cluster and Server targets.

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 12-8](#), the Cluster is selected in the navigation tree, beneath the Identity and Access node. [Figure 12-8](#) illustrates the Cluster Deployments and Performance Overview sections. This page includes a table for Token Issuance and Token Validations.

Figure 12-8 Cluster Page



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 Cluster Deployment section. Figure 12-9 illustrates the OAM Server instance Key Metrics, which include Token Issuance and Token Validations per second. The Token Validation success rate is included.

Figure 12-9 Key Metrics for Server Page

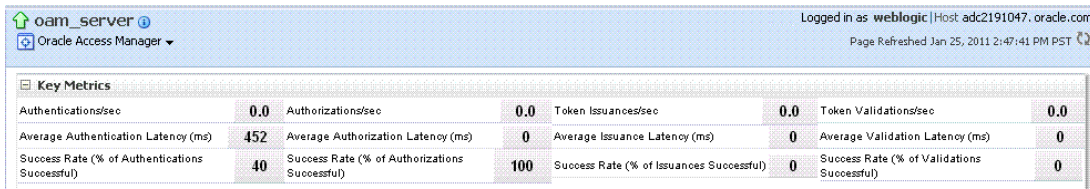


Table 12-3 describes the elements of the Performance Overview for Clusters and OAM Server instances in Fusion Middleware Control. There are only a few differences.

Table 12-3 Summary of Performance Overviews in Fusion Middleware Control

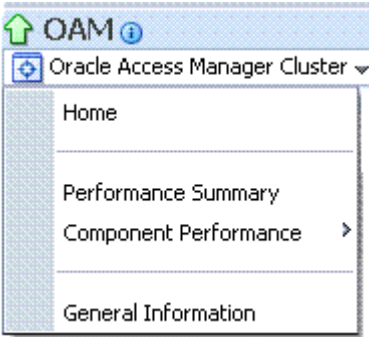
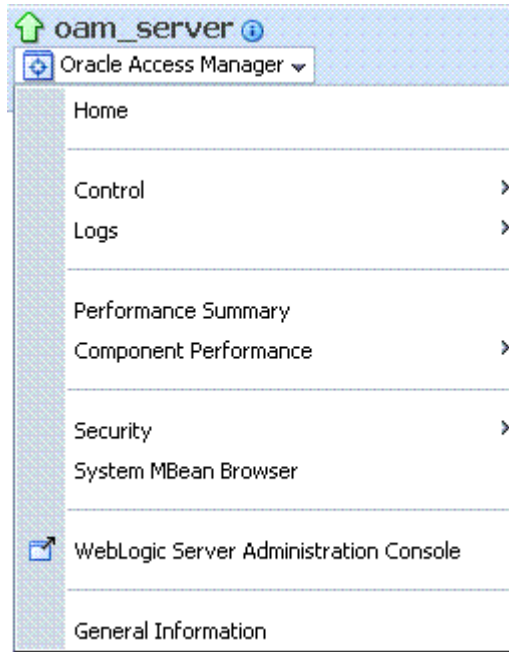
Section or Column Name	Description
Cluster Menu	<p>Dynamic context menus provide functions related to the selected target (also available when you right-click a target in the navigation tree). This menu is available for the selected Cluster.</p> 
Deployments, OAM Cluster pages	<p>The Component Performance command enables you to choose between displaying Access Manager or Security Token Service metrics. See Access Manager Component Pages.</p> <p>This section appears only on OAM Cluster pages. It describes the status of each instance in the cluster. The following information is included:</p> <ul style="list-style-type: none"> • Instance Name • Status • Authentications • Authorizations
Instance Name	<p>This column includes the name of each OAM Server instance in the cluster. For example: <i>OAM_server_name</i></p>
Status	<p>This column identifies the status of each OAM Server instance in the cluster with either a:</p> <ul style="list-style-type: none"> • Green Up Arrow (running) • Red Down Arrow (not running)
Authentications	<p>Authentications columns identify:</p> <ul style="list-style-type: none"> • 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	<p>This column identifies the number of authorizations per second for each OAM Server instance in the cluster. Authorizations columns identify:</p> <ul style="list-style-type: none"> • 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 12-3 (Cont.) Summary of Performance Overviews in Fusion Middleware Control

Section or Column Name	Description
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 server instance.



The Component Performance command enables you to choose between displaying specific Access Manager or Security Token Service metrics. See [Access Manager Component Pages](#).

Key Metrics, OAM Server Page	<p>This table provides a summary of statistics for only the selected OAM Server instance. Key metrics include details for both Access Manager and Security Token Service:</p> <ul style="list-style-type: none"> • 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
------------------------------	--

Table 12-3 (Cont.) Summary of Performance Overviews in Fusion Middleware Control

Section or Column Name	Description
Performance Overview, OAM Cluster and OAM Server Pages	<p>This section provides a graphic representations of Access Manager authentication and authorization operations and 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.</p> <p>Whether you have an OAM Cluster or OAM Server instance selected, the Performance Overview includes:</p> <ul style="list-style-type: none"> • Authentications/sec and Authorizations/sec • Token Issuances/sec and Token Validations/sec <p>Within each table:</p> <ul style="list-style-type: none"> • 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 occurred during the time period.
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	<p>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:</p> <ul style="list-style-type: none"> • LDAP operations/sec • LDAP Latency (milliseconds) • LDAP Success Rate
Application Domains, OAM Cluster and OAM Server Pages	<p>This section of the OAM Cluster and OAM Server pages provides information for all Application Domains that were used during authentication and authorization processing.</p> <p>Columns in this section provide the:</p> <ul style="list-style-type: none"> • 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

12.4.2.1 Access Manager Component Pages

The Component Performance command on both the Cluster and Server instance menus enables you to display Access Manager-specific metrics.



Cluster component-specific metrics are aggregated across the cluster, illustrated in [Figure 12-10](#). Details follow in [Table 12-4](#).

Figure 12-10 Aggregated Access Manager Component Metrics for the Cluster

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 12-11 illustrates the Access Manager component metrics for a single OAM Server instance.

Figure 12-11 Access Manager Component Metrics for a Single OAM Server Instance

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 12-4 describes the component-specific metrics for Access Manager.

Table 12-4 Access Manager Component Metrics

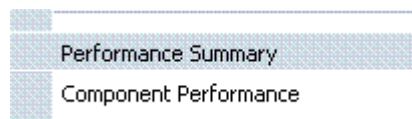
Access Manager Metrics	Description
Access Manager Clients	Based on your selection (Cluster or Server instance), this page provides information for all active Access Clients in a cluster (or for the active Access Clients of an individual OAM Server). Details include: <ul style="list-style-type: none"> Client ID Type Authentications Authorizations
Client ID	Displays the name of the Agent, as defined in the Agent registration in the Oracle Access Management Console.
Type	Displays the Agent. type For example: OAM Webgate

Table 12-4 (Cont.) Access Manager Component Metrics

Access Manager Metrics	Description
Authentications	<p>Authentications columns identify:</p> <ul style="list-style-type: none"> • 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	<p>Authorizations columns identify:</p> <ul style="list-style-type: none"> • 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

12.4.3 Metrics Palette and the Performance Summary Page

The Performance Summary command on the Cluster or Server menu displays metrics charts for the selected target.

Figure 12-12 Performance Summary Command

On the Performance Summary page, a chart is displayed for each selected metric. An OAM Server Performance Summary page. [Figure 12-13](#) 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 12-13 Performance Summary Page with Metric Palette

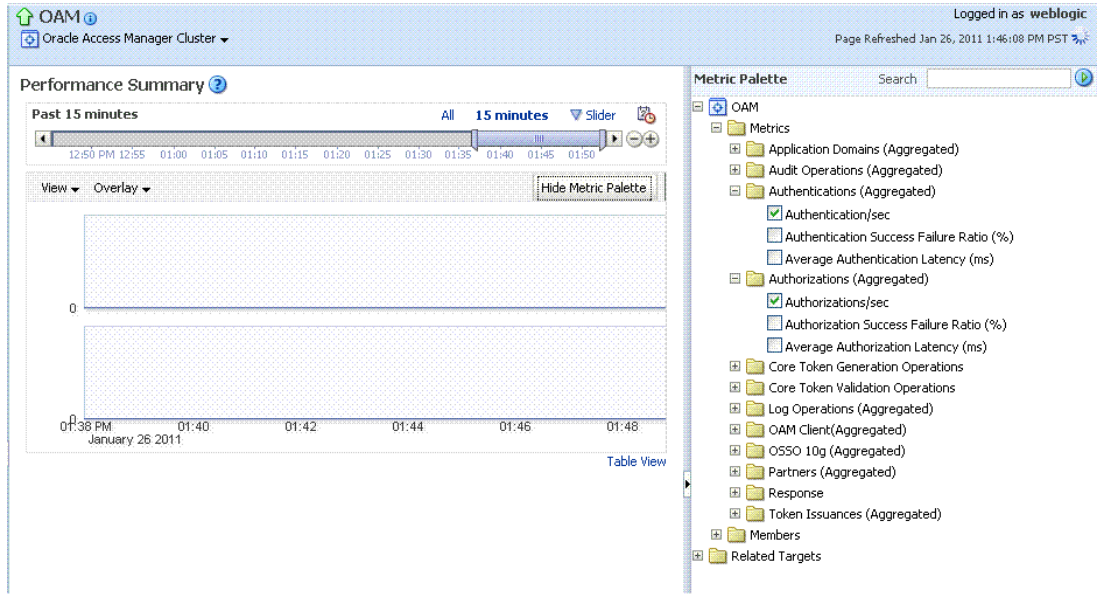


Table 12-5 describes the status and controls available on the Performance Summary page.

Table 12-5 Status and Controls on Performance Summary Pages

Status or Control	Description
Past <i>n</i> minutes	Status is based on the specified time period, which can be adjusted using the slider.
All	
<i>n</i> Minutes	The specified time period, which can be adjusted using the slider.
Slider	The tool you use to adjust the time period.
Chart Set	A list from which you can choose the set of saved charts to view.

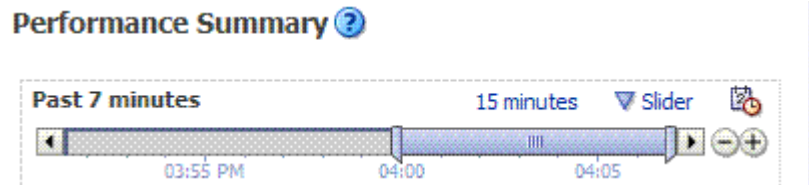


Table 12-5 (Cont.) Status and Controls on Performance Summary Pages

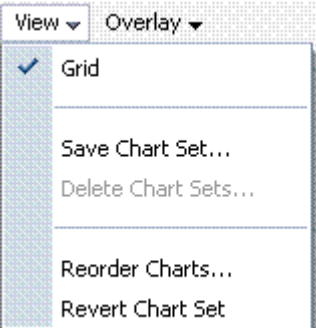
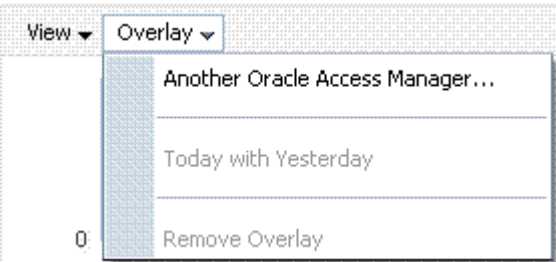
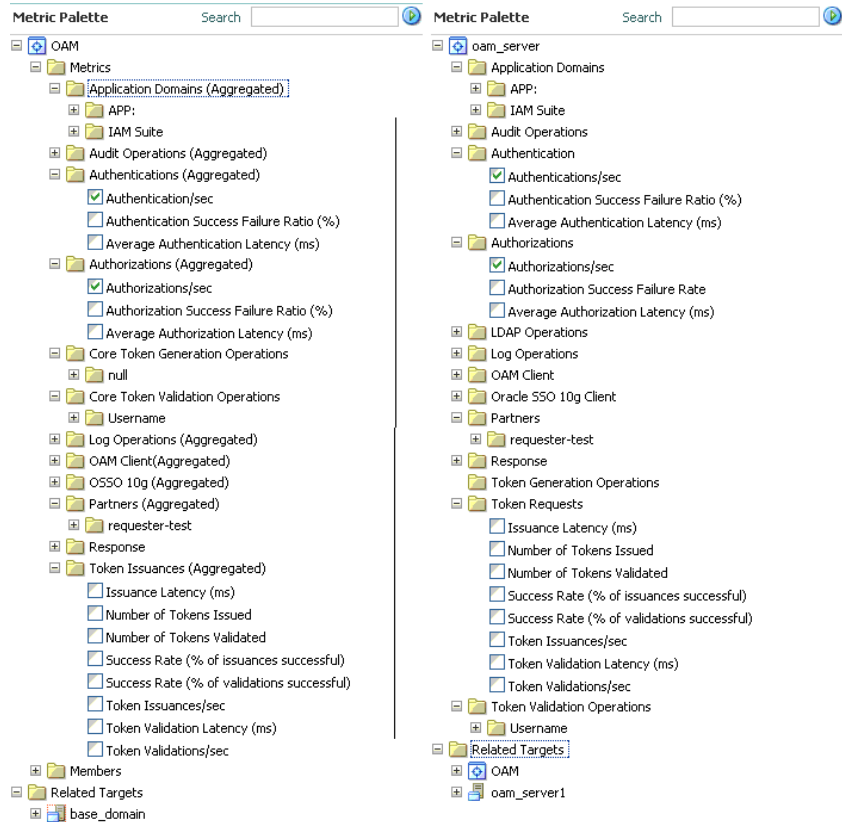
Status or Control	Description
View	A menu that enables you to add a grid, save a chart, and order information on the page.
	
Overlay	A menu that enables you to search for and view another instance of the same type and compare this against the instance in the summary.
	

Table 12-5 (Cont.) Status and Controls on Performance Summary Pages

Status or Control	Description
Metric Palette	A listing from which you can select performance metrics to chart. Items unique to Access Manager and Security Token Service are shown here. Left: Metric Palette for the Cluster Right: Metric Palette for a Single OAM Server



12.4.4 Displaying Performance Metrics in Fusion Middleware Control

Fusion Middleware Control Administrators can add or change the metrics that are displayed in the Performance Summary.

 **See Also:**

- ["Performance Overview Pages in Fusion Middleware Control "](#)
- ["Metrics Palette and the Performance Summary Page"](#)

1. Log in as described in "[Logging In To Fusion Middleware Control](#)".
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, 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 **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 [Validating Connectivity and Policies Using the Access Tester](#)).
- b. In Fusion Middleware Control, check performance metrics.

12.4.5 Displaying Component-Specific Performance Details

Fusion Middleware Control Administrators can use the following procedure to view and compare component-specific performance data.



See Also:

[Access Manager Component Pages](#)

1. Log in as described in "[Logging In To Fusion Middleware Control](#)".
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 [Validating Connectivity and Policies Using the Access Tester](#)).
 - b. In Fusion Middleware Control, check performance metrics.

12.5 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 [Logging Component Event Messages](#) .

Topics in this section include:

- [Dynamic Log Level Changes in Fusion Middleware Control](#)
- [Setting Log Levels Dynamically Using Fusion Middleware Control](#)

12.5.1 Dynamic Log Level Changes in Fusion Middleware Control

Using Fusion Middleware Control, Administrators can change log levels dynamically for **Access Manager** (or **Security Token Service**).

[Table 12-6](#) outlines log availability and functions in Fusion Middleware Control.

Table 12-6 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 AdminServer	Yes	Yes
	oamsso_logout oam_server	Yes	Yes
WebLogic Server domain	oam_bd (Cluster name)	Yes	No
	AdminServer	Yes	Yes
	oam_server	Yes	Yes
Identity and Access	OAM (Cluster)	No	No
	oam_server (Server)	Yes	Yes

[Figure 12-14](#) 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 Security Token Service there is only one logger that affects the log levels for Security Token Service: `oracle.security.fed`.

Figure 12-14 Access Manager Log Levels on the Log Configuration Tab

The screenshot shows the 'Log Configuration' page for Oracle Access Manager. The 'Log Levels' tab is active. The page title is 'oam_server' and the user is logged in as 'weblogic'. The page was refreshed on Jan 26, 2011 3:28:34 PM PST. The main content area contains a table of loggers with their configured levels and log files.

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.

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 12-15 Log Levels for Security Token Service

The screenshot shows the 'Log Configuration' page for Oracle Access Manager, specifically for the Security Token Service (STS). The 'Log Levels' tab is active. The page title is 'oam_server' and the user is logged in as 'weblogic'. The page was refreshed on Jan 26, 2011 3:28:34 PM PST. The main content area contains a table of loggers with their configured levels and log files.

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.

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.CreateV	NOTIFICATION:1 (INFO) [Inherit]	odl-handler	
oracle.security.fed.eventhandler.sts.mapping.wss.MapW5	NOTIFICATION:1 (INFO) [Inherit]	odl-handler	
oracle.security.fed.eventhandler.sts.mapping.wstrust.Mag	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.Validat	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.incomingrequest	NOTIFICATION:1 (INFO) [Inherit]	odl-handler	
oracle.security.fed.plugins.sts.diagnostics.token.validator	NOTIFICATION:1 (INFO) [Inherit]	odl-handler	
oracle.security.fed.plugins.sts.diagnostics.tokenprocessing	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

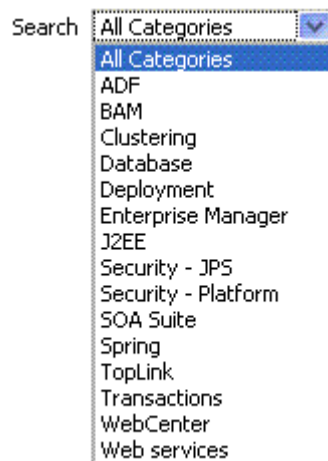
The Log Levels tab on the Log Configuration page allows you to configure the log level for both persistent loggers and active runtime loggers:

- 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 12-7 explains the configuration status and options for log levels.

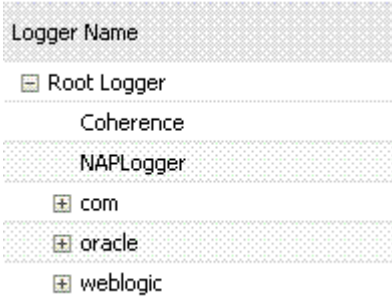
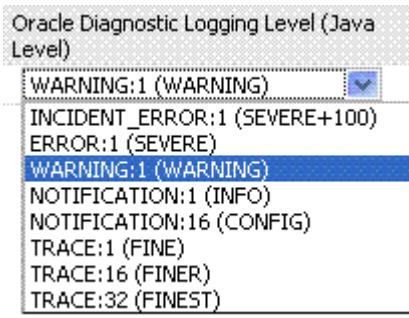
Table 12-7 Log Levels Tab on Log Configuration Page

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. <ul style="list-style-type: none"> • Runtime Loggers • Loggers with Persistent Log Level State
Search	Use this list to specify the categories you would like to search.



Table

Table 12-7 (Cont.) Log Levels Tab on Log Configuration Page

Element	Description
Logger Name	<p>The name of the loggers found during the search. You can expand names in the list to see any loggers beneath the top node.</p> 
Oracle Diagnostic Logging Level (Java Level)	<p>Choose the logging level for the corresponding logger; c.</p>  <p>Click Apply and review confirmation messages displayed in a pop-up window:</p> <ul style="list-style-type: none"> 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	<p>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.</p> <p>See Also: "Managing Log File Configuration from Fusion Middleware Control".</p>
Persistent Log Level State	<p>Identifies the persistent state for this specific logger, which is set when you create or edit the value using the Log Files tab.</p>

12.5.2 Setting Log Levels Dynamically Using Fusion Middleware Control

Fusion Middleware Control Administrators can set the log level dynamically.



See Also:

["Dynamic Log Level Changes in Fusion Middleware Control"](#)



Note:

Administrators can also set logger levels using custom WLST commands as described in [Logging Component Event Messages](#) .

1. Log in as described in "[Logging In To Fusion Middleware Control](#)".
2. Expand the desired node, and select a target. For example:
 Identity and Access
 oam_server
3. From the 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](#)"

12.6 Managing Log File Configuration from Fusion Middleware Control

Fusion Middleware Control Administrators can create, edit, or view the log file.

This section provides the following information:

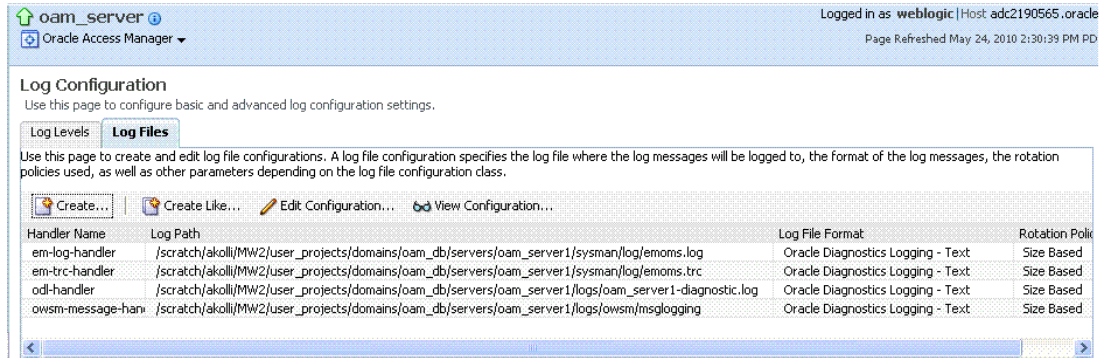
- [Log File Configuration Page in Fusion Middleware Control](#)
- [Managing Log Files with Fusion Middleware Control](#)

12.6.1 Log File Configuration Page in Fusion Middleware Control

Use the Log Files Configuration 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.

[Table 12-7](#) shows the Log Files Configuration.

Figure 12-16 Log Files Configuration Page

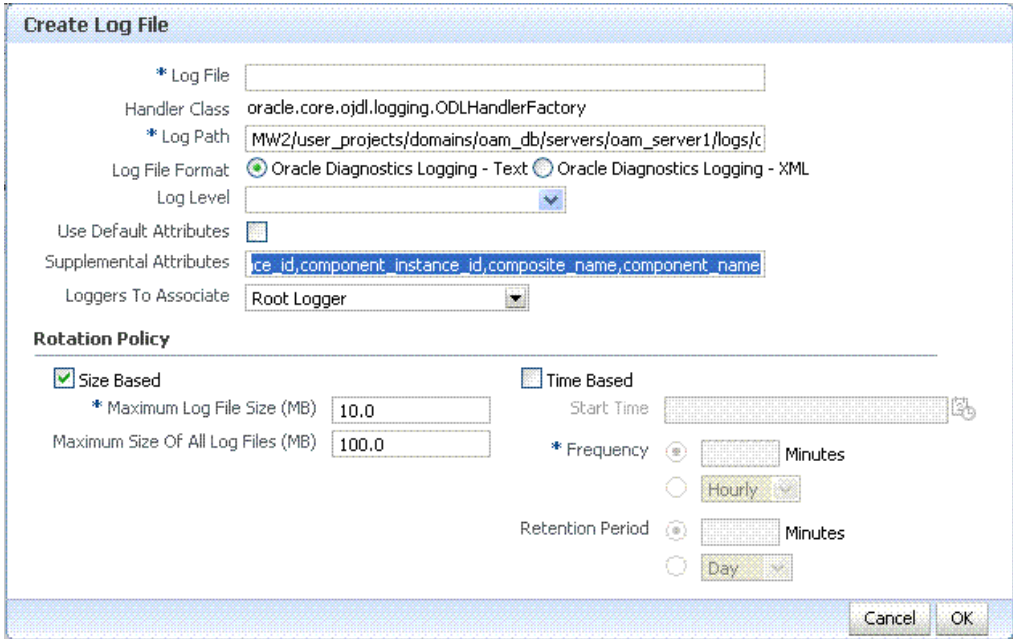


[Table 12-8](#) describes the log files configuration parameters for **Access Manager** (or **Security Token Service**).

Table 12-8 Log Files Elements

Element	Description
Create	Click this button to display the fresh form to create a new file for logged messages. <ul style="list-style-type: none"> 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.

Table 12-8 (Cont.) Log Files Elements

Element	Description
Create Like	Click this button to display a partially filled-in form to create a new file for logged messages.
	
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.
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.

12.6.2 Managing Log Files with Fusion Middleware Control

Fusion Middleware Control Administrators can create a log file, edit the configuration, or view a read-only version of the log file configuration.



See Also:

["Log File Configuration Page in Fusion Middleware Control"](#)

1. Log in as described in "[Logging In To Fusion Middleware Control](#)".
2. Expand the desired node, and select a target. For example:
 - Identity and Access
 - oam_server
3. From the Access Manager menu, select Logs and then Log Configuration.
4. **Create a Log File:** From the Log Files tab ([Table 12-8](#)):
 - 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
 - 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.

Contents are greyed out when opened for viewing configuration.

8. Proceed to "[Viewing Log Messages in Fusion Middleware Control](#)".

12.7 Viewing Log Messages in Fusion Middleware Control

Fusion Middleware Control Administrators can locate, view and export log messages for the target.

This section includes the following topics:

- [About Finding, Viewing, and Exporting Log Messages](#)
- [Viewing Logged Messages With Fusion Middleware Control](#)

12.7.1 About Finding, Viewing, and Exporting Log Messages

By using the context menu for an OAM Server instance in Fusion Middleware Control, Administrators can locate, view, and export key log information.

The key log information can be managed for:

- Application Deployment targets, including the WebLogic (and OAM) AdminServer and the OAM SSO logout pages on both AdminServer and OAM Servers
- WebLogic Server domain targets, including the OAM Farm, AdminServer, and OAM Servers
- Identity and Access targets, including the OAM Farm, 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. See [About Diagnostic Log Files in *Securing Applications with Oracle Platform Security Services*](#)
- 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

12.7.1.1 Log Messages Page in Fusion Middleware

[Figure 12-17](#) shows the Log Messages page for **Access Manager** and **Security Token Service** in Fusion Middleware Control.

Figure 12-17 Typical Log Messages Page in Fusion Middleware Control

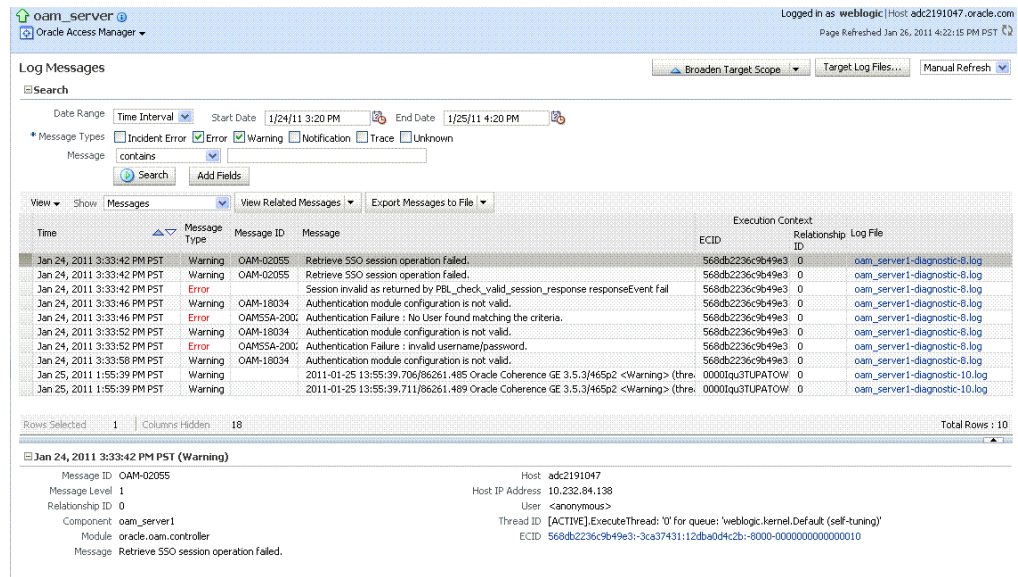


Table 12-9 describes elements on the Log Messages page in Fusion Middleware Control, which you can use to locate and view messages.

Table 12-9 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: <ul style="list-style-type: none"> Oracle WebLogic Server domain OAM 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: <ul style="list-style-type: none"> Manual Refresh 30 Second Refresh 1 Minute Refresh
Search Options	
Date Range	The period during which the desired set of messages was logged: <ul style="list-style-type: none"> Most Recent Minutes Hours Days Time interval Date Range Start Date End Date

Table 12-9 (Cont.) OAM Log Message Search Controls in Fusion Middleware Control

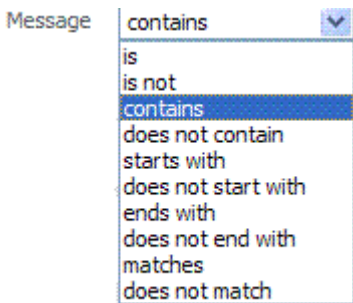
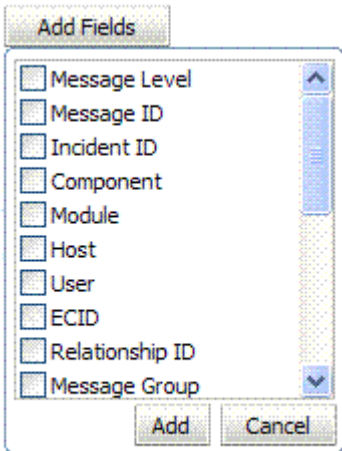
Element	Description
Message Types	<p>Check all message types that apply for this search:</p> <ul style="list-style-type: none"> • Incident Error • Error • Warning • Notification • Trace • Unknown
Message	<p>Choose an identifier from this list and add a value in the blank field beside it to refine your search criteria:</p> 
Add Fields	<p>Click this button to display a list of additional search criteria you can include.</p> 
Search	<p>Click this button to initiate a search using the specified criteria.</p>
Viewing Options	

Table 12-9 (Cont.) OAM Log Message Search Controls in Fusion Middleware Control

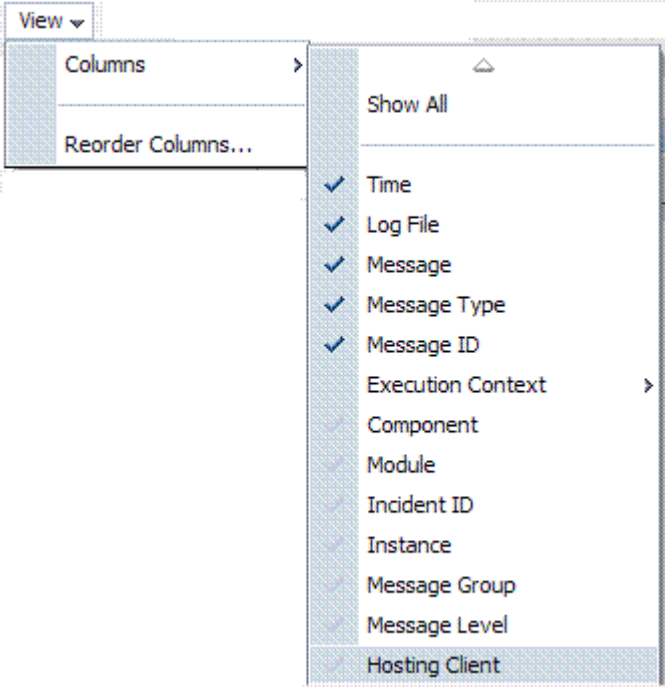
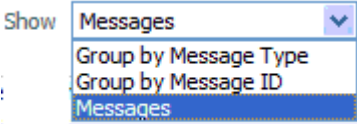
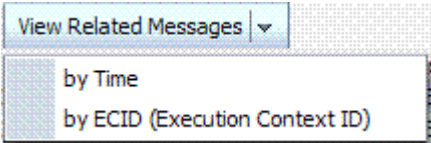
Element	Description
View	Choose items from this menu to view or reorder columns in the search results table:
	
Show	Select the entity to view:
	
View Related Messages	This menu is available when at least one message is listed in the search results.
	

Table 12-9 (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: <div data-bbox="402 457 852 646" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Export Messages to File ▾</p> <ul style="list-style-type: none"> As Oracle Diagnostic Log Text (.txt) As Oracle Diagnostic Log XML (.xml) As Comma-Separated List (.csv) </div>																												
Results Table Columns	These are based on selections in the View menu on the Log Messages page. <div data-bbox="402 821 1377 888" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <table border="1"> <thead> <tr> <th>Time</th> <th>Log File</th> <th>Message</th> <th>Message Type</th> </tr> </thead> <tbody> <tr> <td>May 19, 2010 9:44:17 AM PDT</td> <td>oam_server1-diagn</td> <td>Message received from client. Message OpCode = 1 [IsResrcOpProtected], Seq#</td> <td>Notification</td> </tr> <tr> <td>May 19, 2010 9:44:17 AM PDT</td> <td>oam_server1-diagn</td> <td>Master Controller: processing Event:is_resource_protected.</td> <td>Notification</td> </tr> </tbody> </table> </div>	Time	Log File	Message	Message Type	May 19, 2010 9:44:17 AM PDT	oam_server1-diagn	Message received from client. Message OpCode = 1 [IsResrcOpProtected], Seq#	Notification	May 19, 2010 9:44:17 AM PDT	oam_server1-diagn	Master Controller: processing Event:is_resource_protected.	Notification																
Time	Log File	Message	Message Type																										
May 19, 2010 9:44:17 AM PDT	oam_server1-diagn	Message received from client. Message OpCode = 1 [IsResrcOpProtected], Seq#	Notification																										
May 19, 2010 9:44:17 AM PDT	oam_server1-diagn	Master Controller: processing Event:is_resource_protected.	Notification																										
Message Area	Displays details for the selected message in the search results table. <div data-bbox="402 1081 1377 1283" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>May 19, 2010 9:44:17 AM PDT (Notification)</p> <table border="0"> <tr> <td>Message ID</td> <td>OAM-02086</td> <td>Module</td> <td>oracle.oam.controller</td> </tr> <tr> <td>Message Level</td> <td>1</td> <td>Host</td> <td>adc2190565</td> </tr> <tr> <td>dcid</td> <td>1257aa20b86f75d:-6e3af23f:1288ec14b03:-8000-0000000000000010</td> <td>Host IP Address</td> <td>10.232.82.164</td> </tr> <tr> <td>Relationship ID</td> <td>0</td> <td>User</td> <td><anonymous></td> </tr> <tr> <td>Argument 1</td> <td>Master Controller</td> <td>Thread ID</td> <td>[ACTIVE].ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'</td> </tr> <tr> <td>Argument 2</td> <td>is_resource_protected</td> <td>ECID</td> <td>00001Y5sEg9LeWlHyt1f1BunDp0000dg</td> </tr> <tr> <td>Component</td> <td>oam_server1</td> <td></td> <td></td> </tr> </table> <p>Message Master Controller: processing Event:is_resource_protected.</p> </div>	Message ID	OAM-02086	Module	oracle.oam.controller	Message Level	1	Host	adc2190565	dcid	1257aa20b86f75d:-6e3af23f:1288ec14b03:-8000-0000000000000010	Host IP Address	10.232.82.164	Relationship ID	0	User	<anonymous>	Argument 1	Master Controller	Thread ID	[ACTIVE].ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'	Argument 2	is_resource_protected	ECID	00001Y5sEg9LeWlHyt1f1BunDp0000dg	Component	oam_server1		
Message ID	OAM-02086	Module	oracle.oam.controller																										
Message Level	1	Host	adc2190565																										
dcid	1257aa20b86f75d:-6e3af23f:1288ec14b03:-8000-0000000000000010	Host IP Address	10.232.82.164																										
Relationship ID	0	User	<anonymous>																										
Argument 1	Master Controller	Thread ID	[ACTIVE].ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'																										
Argument 2	is_resource_protected	ECID	00001Y5sEg9LeWlHyt1f1BunDp0000dg																										
Component	oam_server1																												

12.7.2 Viewing Logged Messages With Fusion Middleware Control

Fusion Middleware Control Administrators can 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.

1. Log in as described in "Logging In To Fusion Middleware Control".
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 12-9):

- 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 [Validating Connectivity and Policies Using the Access Tester](#)).
 - 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](#)".

**See Also:**

"[About Finding, Viewing, and Exporting Log Messages](#) "

12.8 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:

- [Fusion Middleware Control System MBean Browser](#)
- [Managing Mbeans](#)

12.8.1 Fusion Middleware Control System MBean Browser

The Fusion Middleware Control System Mbean Browser can be used to view System MBean Browser for Nodes and Targets.

[Table 12-10](#) details about the System MBean Browser..

Table 12-10 System MBean Browser

Node	Target	System Mbean Browser
Application Deployment	...AdminServer	Yes
Internal Applications	oamssso_logout(11.1.1.3.0)	Yes
	AdminServer	Yes
	oamssso_logout(11.1.1.3.0) oam_server	
WebLogic Server domain	oam_bd (Cluster name)	Yes
	AdminServer	Yes
	oam_server	Yes
Identity and Access	OAM (Cluster)	No
	oam_server (Server)	Yes



Note:

Security Token Service MBeans are also available as described here.

[Table 12-11](#) describes the MBeans that **Access Manager** and **Security Token Service** deploy on the AdminServer on the domain runtime server (OAM Server).

Table 12-11 MBeans that Access Manager and Security Token Service Deploy

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 12-18 Shows the System MBean Browser and the related Attributes tab displaying information for the Security Token Service CertRevocationListConfig: oracle.sts:Location=oam_server1,type=CertRevocationListConfig.

Figure 12-18 System MBean Browser and Attributes Tab

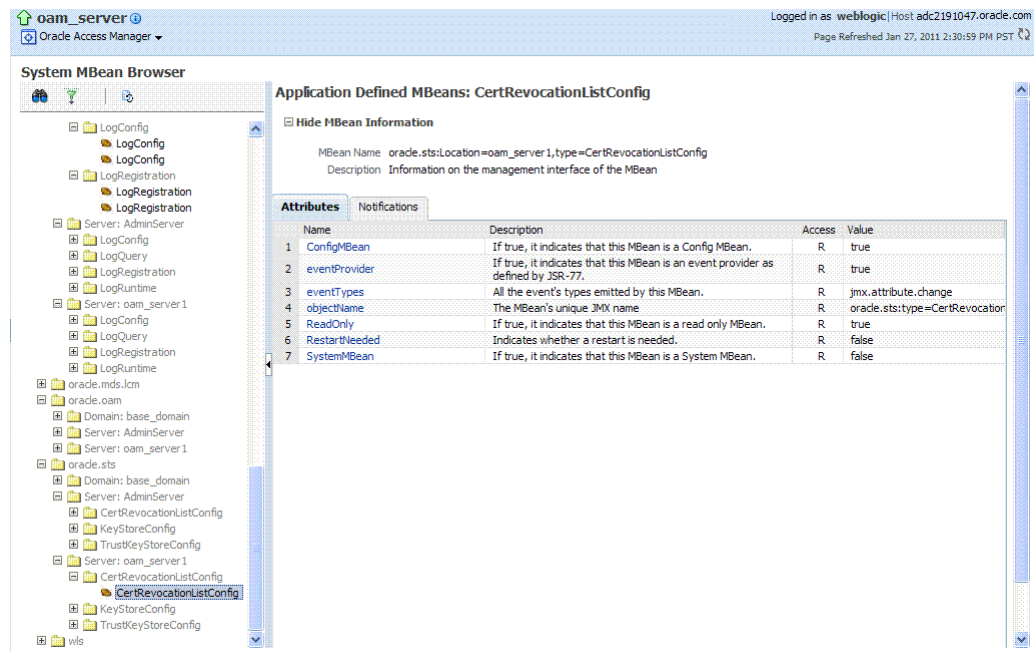


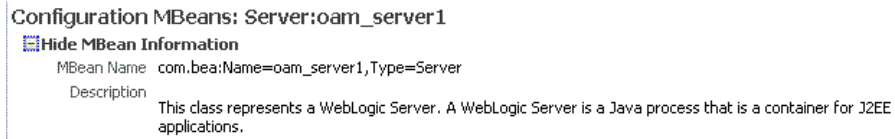
Table 12-12 describes the System MBean Browser and associated tab in greater details.

Table 12-12 System MBean Browser

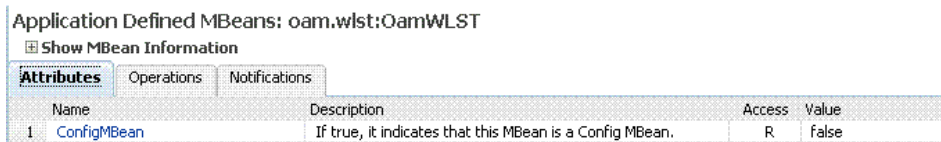
System MBean Browser	Description
System MBean Browser	Expand items in this section to display Mbeans for the selected target. Under Application Defined Beans, find <code>oracle.oam</code> and <code>oracle.sts</code> .



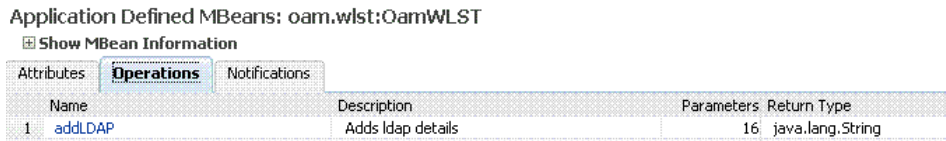
MBean Information Details for Attributes and Operations related to the MBean for the selected target are displayed on the right.



Attributes This tab describes MBean attributes for the selected target.



Operations This tab describes MBean operations for the selected target.



Notifications This tab lists any notifications resulting from the invocation of an MBean.

Table 12-12 (Cont.) System MBean Browser

System MBean Browser	Description
Controls	<p>The following controls are available from these pages:</p> <ul style="list-style-type: none"> • 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

12.8.2 Managing Mbeans

Fusion Middleware Control Administrators can view, edit or invoke MBeans for **Access Manager** and **Security Token Service**. Additionally, you can apply values (or revert the change) and invoke MBeans.

1. Log in as described in "[Logging In To Fusion Middleware Control](#)".
2. Expand the desired node and select a target. For example:
 - Identity and Access
 - oam_server
3. From the 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.

Part IV

Managing Access Manager Settings and Agents

Administrators enable SSO across enterprise applications through agents registration and protect specific applications using Access Manager policies.

Administrators manage low-level Access Manager settings, agents, and sessions. This section contains the following chapters:

- [Configuring Access Manager Settings](#)
- [Introduction to Agents and Registration](#)
- [Registering and Managing OAM Agents](#)
- [Maintaining Access Manager Sessions](#)

13

Configuring Access Manager Settings

The Access Manager Settings provide configuration options for a number of specific Access Manager service operations.

This chapter describes these Access Manager-specific settings.

- [Oracle Access Management Overview](#)
- [Managing Load Balancing](#)
- [Managing 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](#)

13.1 Oracle Access Management Overview

Familiarize with the Oracle Access Management, Serve registration and management before you start off with configuration options and service operations.

Be sure to review the following topics:

- [Getting Started with Oracle Access Management](#)
- [Managing Server Registration](#)

13.2 Managing Load Balancing

Configure two or more Managed Servers to operate as a cluster and use Oracle Access Management Console for Access Manager load balancing settings.

This section describes the following topics:

- [About Common Load Balancing Settings](#)
- [Managing OAM Server Load Balancing Settings](#)

13.2.1 About Common Load Balancing Settings

For production environments that require increased application performance, throughput, or high availability, you can configure two or more Managed Servers to operate as a cluster. A cluster is a collection of multiple WebLogic Server server instances running simultaneously and working together to provide increased scalability and reliability.

In a cluster, most resources and services are deployed identically to each Managed Server (as opposed to a single Managed Server), enabling failover and load balancing. A single domain can contain multiple WebLogic Server clusters and multiple Managed Servers that are not configured as clusters. The key difference between clustered and non-clustered

Managed Servers is support for failover and load balancing. These features are available only in a cluster of Managed Servers.

By default, Access Manager has a single OAM Server to which all login and logout requests are sent. In a high-availability deployment, you must change this setup so that login and logout requests are first sent to the load balancer.



See Also:

High Availability Guide, "Access Manager High Availability Configuration Steps" for high-level instructions to set up a high availability Access Manager deployment.

Figure 13-1 shows the Load Balancing Settings section of the Access Manager Settings page. In earlier releases this was part of the SSO Engine settings; the SSO Engine being the controller for sessions.

Figure 13-1 Access Manager Settings: Load Balancer

Configuration >

Access Manager Settings
The following settings apply to the Access Manager service.

▲ Load Balancing

* OAM Server Host * OAM Server Protocol

* OAM Server Port * Server Error Mode

Table 13-1 describes each element and how it is used. Settings are global and common to all OAM Servers in the WebLogic administration domain.

Table 13-1 Access Manager Settings: Load Balancer

Element	Description
OAM Server Host	The virtual host name that represents the OAM Server Cluster, which might be exposed by a load balancer in front of an OAM Server Cluster.
OAM Server Port	The virtual host port associated with the OAM Server Cluster. Values between 1 and 65535 are supported.
OAM Server Protocol	The protocol, either HTTP or HTTPS, that is used to access the virtual host that represents the OAM Server Cluster. See Also: " About Security Modes and X509Scheme Authentication "

13.2.2 Managing OAM Server Load Balancing Settings

Users with valid Administrator credentials can modify Access Manager load balancing settings using Oracle Access Management Console.



See Also:

["About Common Load Balancing Settings "](#)

1. From the Access Manager Settings, open Load Balancing:
2. Expand the Load Balancing area:
 - View Only: Close the page when you finish.
 - Modify: Edit Load Balancing settings for your deployment ([Table 13-1](#)).
3. Click Apply to submit the changes (or close the page without applying changes).
4. Dismiss the Confirmation window.

13.3 Managing Secure Error Modes

A custom error page is packaged as part of the custom login application. An out-of-the-box custom Web application archive file is provided that you can use as a starting point to develop customized login and password pages.

Server Error Mode settings are global and common to all OAM Servers in the WebLogic administration domain. This section provides the following topics:

- [OAM Server Error Modes](#)
- [Viewing or Editing OAM Server Secure Error Modes](#)

13.3.1 OAM Server Error Modes

The OAM Server Error Mode appears on the Load Balancing Settings area of the Access Manager Settings page.

[Figure 13-1](#) shows the Server Error Mode function.

Figure 13-2 Access Manager Settings: Server Error Mode

[Configuration >](#)

Access Manager Settings
The following settings apply to the Access Manager service.

▲ **Load Balancing**

* OAM Server Host	<input type="text" value="slc01mqd.us.oracle.c"/>	* OAM Server Protocol	<input type="text" value="http"/>
* OAM Server Port	<input type="text" value="14100"/> ▲ ▼	* Server Error Mode	<input type="text" value="External"/>

Table 13-2 describes the options you can choose to configure Server Error Mode for your deployment.

Table 13-2 Server Error Mode

Element	Description
Server Error Mode	<p>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)).</p> <p>Choose one of the following settings to configure error messages with varying degrees of security for your custom login pages:</p> <ul style="list-style-type: none"> • 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. Recommended for Password Policy validation, as described in "Managing Global Password Policy". <p>See Also: "Viewing or Editing OAM Server Secure Error Modes"</p>

Table 13-3 shows the error triggering condition and message codes for each of the three modes.

Table 13-3 Error Triggering Condition, Modes, and Message Codes

Error Triggering Condition	Internal Mode	External Mode	Secure Mode
Invalid login attempt	OAM-1	OAM-2	OAM-8
Processing submitted credentials fails. For example: In WNA mode, the SPNEGO token is not received.	OAM-3	OAM-3	OAM-8
An authentication exception is raised.	OAM-4	OAM-4	OAM-9
User account gets locked based on certain conditions (exceeded invalid attempts, for instance).	OAM-5	OAM-5	OAM-8
User account disabled.	OAM-5	OAM-5	OAM-9
User has exceeded the maximum number of allowed sessions (a configurable attribute).	OAM-6	OAM-6	OAM-9
Default error message, which is displayed when no other specific messages propagate up. This is not propagated to the user level. Cause could be multiple conditions.	OAM-7	OAM-7	OAM-9
Password expired.	OAM-10	OAM-10	OAM-9

Table 13-4 identifies the error codes, trigger conditions, and recommended messages.



See Also:

Developing Custom Error Pages in the *Developing Applications with Oracle Access Management*

Table 13-4 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).	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 OIG 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.
OAM-7	Failure could be due to multiple reasons; the exact reason is not propagated to the user level for security reasons. For instance: <ul style="list-style-type: none"> • 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.
OAM-8	See Table 13-3	Authentication failed.
OAM-9	System error. Please re-try your action. If you continue to get this error, please contact the Administrator.	System error. Please re-try your action. If you continue to get this error, please contact the Administrator.
OAM-10	Password expired.	The password has expired.

13.3.2 Viewing or Editing OAM Server Secure Error Modes

Users with valid Administrator credentials can view or edit Access Manager secure error mode settings for OAM Servers using the Oracle Access Management Console.



See Also:

["About Common Load Balancing Settings "](#)

To view or edit:

1. In the Configuration console, select **Access Manager** from the **View** menu in the **Settings** section.
2. On the Access Manager Settings page, expand the **Load Balancing** section.
3. Server Error Mode:
 - **Modify**: Choose the desired **Server Error Mode** for your deployment ([Table 13-2](#) and [Table 13-4](#)).
 - **View Only**: Close the page when you finish.
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"](#).

13.4 Managing WebGate Traffic Load Balancer

This section describes the following topics:

- [About WebGate Traffic Load Balancer](#)
- [Viewing or Editing WebGate Traffic Load Balancer](#)

13.4.1 About WebGate Traffic Load Balancer

The WebGate Load Balancer settings are used to initialize the WebGate profile parameters whenever a new profile is created.

The following figure shows the WebGate Traffic Load Balancer section of the Access Manager Settings page:

Figure 13-3 Access Manager Settings: WebGate Traffic Load Balancer

WebGate Traffic Load Balancer

* OAM Server Host * OAM Server Protocol

* OAM Server Port

The following table describes each element and how it is used. Settings are global and common to all OAM servers in the Weblogic administration domain:

Table 13-5 Access Manager Settings: WebGate Traffic Load Balancer

Element	Description
OAM Server Host	The managed server or load balancer host information.
OAM Server Port	The managed server or load balancer port information.
OAM Server Protocol	The protocol that is used to communicate between the WebGate and the managed servers. Values: HTTP, HTTPS

13.4.2 Viewing or Editing WebGate Traffic Load Balancer

Users with valid Administrator credentials can view or edit WebGate Load Balancer settings for OAM Servers using the Oracle Access Management Console.

To view or edit:

1. In the Configuration console, select **Access Manager** from the **View** menu in the **Settings** section.
2. On the Access Manager Settings page, expand the WebGate Traffic Load Balancer section.
3. Edit Settings as needed for your configuration.
4. Click **Apply** to submit the changes.
5. Navigate to Application Security Console, select **Create Webgate** from the **Agents** menu and create a new agent. Verify the WebGate created is populated with the modified values in the **User Defined Parameters** field.

13.5 Managing SSO Tokens and IP Validation

Use Oracle Access Management Console for modifying Access Manager SSO settings, IP Validation and SSO token version.

This section provides the following topics:

- [Access Manager SSO Tokens and IP Validation Settings](#)
- [Viewing or Editing SSO Tokens and IP Validation](#)

13.5.1 Access Manager SSO Tokens and IP Validation Settings

The Access Manager Settings page include information such as **IP Validation** and **SSO token version**.

[Table 13-6](#) describes each element and how it is used.

Table 13-6 Access Manager Settings: SSO

Element	Description
IP Validation	<p>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.</p> <p>Check the box to enable IP Validation.</p> <p>Clear the box to disable IP Validation if and only if IP Validation is disabled on all the configured WebGates. See IP Address Validation for WebGates.</p>
SSO Token Version	<p>SSO token version is the version of the SSO_ID token (cookie) created by the OAM server.</p> <p>SSO token versions are mainly used for compatibility between data centers in a Multi-Data Center (MDC) setup. Ensure that all the data centers in the MDC setup have SSO token version 5 selected.</p> <p>From OAM 12c onwards, SSO token version 5 is selected by default.</p>

13.5.2 Viewing or Editing SSO Tokens and IP Validation

Users with valid Administrator credentials can view or edit Access Manager SSO settings using the Oracle Access Management Console.



See Also:

["About Common Load Balancing Settings"](#)

To view or edit:

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. In the Configuration console, select **Access Manager** from the **View** menu in the **Settings** section.
3. Expand the **SSO** section:
 - View Only: Close the page when you finish.
 - Modify: Perform remaining steps to edit the configuration.
4. Edit settings as needed for your deployment, based on details in [Table 13-6](#).
5. Click **Apply** to submit the changes (or close the page without applying changes).
6. Dismiss the Confirmation window.
7. Proceed to ["Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security"](#).

13.6 Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security

Configure secure server communication modes and manage through the settings for the common OAM Proxy.

This section describes the following topics:

- [OAM Proxy Simple and Cert Mode Transport Security](#)
- [Configuration Settings of Common OAM Proxy Page for Secure Server Communications](#)
- [Viewing or Editing Simple or Cert Settings for OAM Proxy](#)
- [Configuring 64-bit WebGate in Cert Mode](#)
- [Tuning the Simple Mode WebGate](#)

13.6.1 OAM Proxy Simple and Cert Mode Transport Security

Open, Simple, and Cert are the modes of secure communication and there are similarities between these modes.

[Table 13-7](#) outlines the similarities between Simple and Cert modes.


 **See Also:**
[Securing Communication](#)

Table 13-7 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 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: <ul style="list-style-type: none"> • password.xml 	PEM format: <ul style="list-style-type: none"> • Keystore Alias • Key KEYSTOREStore Alias Password 	N/A

Table 13-7 (Cont.) Summary: Simple and Cert Mode

Artifact or Process	Simple Mode	Cert Mode	Open Mode
During OAM Agent or OAM Server registration, the communication mode is propagated to the Oracle Access Management 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 Server. 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: <pre>openssl rsa -in aaa_key.pem - passin pass: -out aaa_key.pem - passout pass: <i>passphrase</i> -des</pre>	N/A	X	N/A
Agent Key Password	N/A	Enter a password during agent registration in Cert Security mode (see Table 15-1).	N/A
During Agent registration, ObAccessClient.xml is generated in: \$DOMAIN_HOME/ output/\$Agent_Name/	ObAccessClient.xml Copy to: 11g Webgate: \$OHS_Instance_Dir	ObAccessClient.xml Copy to: 11g Webgate: \$OHS_Instance_Dir	ObAccessClient.xml Copy to: 11g Webgate: \$OHS_Instance_Dir
During Agent registration, password.xml is generated in: \$DOMAIN_HOME/ output/\$Agent_Name/ See Also: Securing Communication	password.xml Copy to: 11g Webgate: \$OHS_Instance_Dir	password.xml Copy to: 11g Webgate: \$OHS_Instance_Dir	N/A
During Agent registration, aaa_key.pem is generated in: \$DOMAIN_HOME/ output/\$Agent_Name/ See Also: Securing Communication	aaa_key.pem Copy to: 11g Webgate: \$OHS_Instance_Dir	aaa_key.pem Copy to: 11g Webgate: \$OHS_Instance_Dir	N/A

13.6.2 Configuration Settings of Common OAM Proxy Page for Secure Server Communications

You can the configure settings of Common OAM Proxy Page for Secure Server Communications.

[Table 13-8](#) describes the settings required for Simple or Cert mode configurations.

Table 13-8 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: <ul style="list-style-type: none"> • PEM Keystore Alias • PEM Keystore Alias Password <p>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.</p> <p>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".</p>

13.6.3 Viewing or Editing Simple or Cert Settings for OAM Proxy

Administrators can use view or edit Simple or Cert mode settings for the common OAM Proxy.



See Also:

- "[Registering an OAM Agent Using the Console](#)"
- "[Securing Communication](#)"

To view or edit:

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. In the Configuration console, select **Access Manager** from the **View** menu in the **Settings** section.
3. Expand the **Access Protocol** section.
4. **Simple Mode Configuration:** Add or alter a Global Passphrase if you are using OAM-signed X.509 certificates.

5. **Cert Mode Configuration:** Specify the following details.
 - PEM Keystore Alias
 - PEM Keystore Alias Password
6. Click **Apply** to submit the changes and dismiss the Confirmation window (or close the page without applying changes).
7. Update Agent registration pages as needed to regenerate artifacts, and then replace the earlier artifacts as described in [Introduction to Agents and Registration](#) or [Registering and Managing OAM Agents](#).

13.6.4 Configuring 64-bit WebGate in Cert Mode

64-bit WebGates now support SHA2 (256,384 & 512 bit) certificates.

Run the following command to configure a 64-bit WebGate in cert mode.

```
<Oracle Middleware Home>/oracle_common/bin/orapki wallet add  
-wallet $DOMAIN_HOME/output/$Agent_Name/cwallet.sso -trusted_cert  
-cert <Root CA path .i.e. aaa_chain.pem> -auto_login_only
```

13.6.5 Tuning the Simple Mode WebGate

If using a simple mode WebGate, you can improve the response time of the OAM login page by changing the `aaaTimeoutThreshold` time parameter in the WebGate profile from -1 to 10.

For detailed information about the AAA Timeout Threshold configuration element, see [Table 15-3](#) in [Registering and Managing OAM Agents](#).

13.7 Managing Run Time Policy Evaluation Caches

Access Manager common run time policy evaluation cache settings are managed by administrators and policy evaluation caches are required during policy evaluation at run time.

This section describes the following topics:

- [Settings for Run Time Policy Evaluation Caches](#)
- [Managing Run Time Policy Evaluation Caches](#)



See Also:

["Run Time Resource Evaluation "](#)

13.7.1 Settings for Run Time Policy Evaluation Caches

The Resource Matching Cache and the Authorization Result Cache are set and required during policy evaluation at run time.

[Figure 13-4](#) illustrates the Policy section of the Access Manager Settings page.

Figure 13-4 Common Policy Evaluation Caches

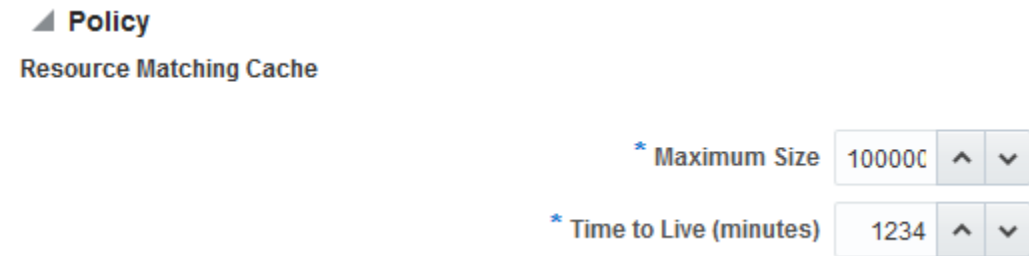



Table 13-9 outlines these global settings that apply to all servers and requests.

Table 13-9 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: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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: <i>Tuning Performance</i>

 **See Also:**

[Polling Interval for System and Policy Configuration](#)

13.7.2 Managing Run Time Policy Evaluation Caches

Administrators manage the Access Manager common run time policy evaluation cache settings.

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. In the Configuration console, select **Access Manager** from the **View** menu in the **Settings** section.
3. Expand the **Policy** section.
4. **Resource Matching Cache:** Specify details and click apply (Table 13-9).
5. **Authorization Result Cache:** Specify details and click apply (Table 13-9).
6. Click **Apply** to submit the changes and dismiss the Confirmation window (or close the page without applying changes).

 **See Also:**

- *High Availability Guide*
- *Tuning Performance*

13.8 Configuring Policy Cache Parameters

The parameters for Policy Cache can be set in oam-config.xml file under OAMPolicyProvider or in System Properties.

Following table lists the Policy Cache parameters with their default values:

Table 13-10 Polciy Cache Parameters

Parameter Name	Default value	Description
PolicyCacheRefreshIntervalMilliseconds	30000 (30 Sec)	Configured in runtime server to check for new version in store after this configured time. It is configured in milliseconds.
PolicyCacheRefreshTimeoutSeconds	300 (5 min)	Configured in runtime server to wait for 300 seconds before stopping the execution of each query. It is configured in seconds.
PolicyCacheCountInDb	5	Configured in Admin/Policy manager server for number of latest versions of policy cache to be stored in the database. Older version of caches after this threshold limit will be deleted.
PolicyCacheIgnoreChecksum	FALSE	Configured in Admin/Policy manager runtime server to ignore the policy check sum calculation and read the cache from the store.
PolicyCacheThreadMaxRetryCount	5	Configured in runtime server for number of times the policy version thread will retry in case of DB timeout.
PolicyCacheReadThreadBlockTimeInSec	1800(30 Mins)	Configured in runtime server after reaching the number of retrials specified by PolicyCacheThreadMaxRetryCount , the policy version thread will avoid checking for the configured amount of time. It is configured in seconds.
PolicyCacheCheckTimeoutLimit	10mins	Configured in runtime server to wait for the runtime server cache initialization. It is configured in minutes.

Table 13-10 (Cont.) Polciy Cache Parameters

Parameter Name	Default value	Description
<code>oracle.oam.EntityRefreshIntervalMillis</code>	60000 Millisecond	Configured in Admin server or Policy Manager as system property to increase the frequency to check the changes in the policy on Admin server or Policy Manager. It is default to 60000ms.

 **Note:**

The `oracle.oam.EntityRefreshIntervalMillis` parameter can be configured only in System properties and not in `oam-config.xml` file.

14

Introduction to Agents and Registration

An agent (also known as a single sign-on agent or policy-enforcement agent) is any front-ending entity that acts as an access client to enable single sign-on across enterprise applications.

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

This chapter includes the following topics to give you an overview of agents, their registration and management, processing, and tools.

- [Introduction to Policy Enforcement Agents](#)
- [Introduction to Agent Registration](#)
- [OAM Remote Registration](#)

14.1 Introduction to Policy Enforcement Agents

An agent is a software plug-in that can be installed on a Web server (such as Oracle HTTP Server) where the application resides. To secure access to protected resources, a Web server, Application Server, or third-party application must be associated with an agent that is registered with Access Manager. To spare users from re-authenticating when accessing multiple resources, the application delegates the authentication function to the single sign-on (SSO) provider: Access Manager.

During agent registration, the application can be automatically registered and basic policies automatically generated. Alternatively, you can turn off automatic policy generation during Agent registration and manually create policies.

After registration, the Agent acts as a filter for HTTP/HTTPS requests, communicating between the OAM Server and its services. The Agent intercepts requests for resources protected by Access Manager and works with Access Manager to fulfill access requirements. The following sections introduce the types of agents.

- [Agent Types and Runtime Processing for OAM Agents](#)
- [About OAM WebGate Configured as a Detached Credential Collector](#)

14.1.1 Agent Types and Runtime Processing for OAM Agents

With Access Manager, each Agent acts as a filter for requests.

Your deployment can include the agent types described in [Table 14-1](#), in any combination.

Table 14-1 Agent Types

Agent Type	Description
OAM Agents	OAM Agents must be installed independently, following Oracle Access Management installation. After registering the agent with Access Manager, the agent communicates directly with registered OAM Servers and Access Manager services. OAM Agents communicate with Access Manager using the OAM Proxy to "sanitize" the request and respond identically for all agents. The following OAM Agents types are available:
Note: Unless explicitly stated, the terms Webgate and Access Client are used interchangeably.	<ul style="list-style-type: none"> Webgate: An out of an box Web server access client that intercepts HTTP requests for Web resources and forwards these to the OAM Server. WebGates for various Web servers are shipped with Access Manager. <ul style="list-style-type: none"> WebGates provide: <ul style="list-style-type: none"> Oracle Universal Installer for platform Host-based cookie Individual WebGate OAMAuthnCookie_<host:port> Resource to Authorization Policy Authorization Result Webgate Authorization Caching Diagnostic page to tune parameters Capability to act as a detached credential collector <p>See Also: "About OAM WebGate Configured as a Detached Credential Collector" <i>Tuning Performance</i></p> <ul style="list-style-type: none"> Custom, Programmatic Access Clients: Access Manager provides a pure Java software developer kit (SDK). Use this SDK to create custom Access Clients and extensions for Access Manager authentication and authorization functionality (and custom tokens). An Access Client processes requests for Web and non-Web resources (non-HTTP) from users or applications. See details in the <i>Developing Applications with Oracle Access Management</i>.

[Table 14-2](#) introduces Access Manager features that support agent registration, configuration, management, and single-sign on. Links to topics providing more information are included.

Table 14-2 Agent Registration and SSO Support

Oracle Provides	Description
Oracle Access Management Console	Agent Registration, Configuration, Management. See Also: Registering an OAM Agent Using the Console
oamreg tool	Remote Agent Registration and Management See Also: Acquiring and Setting Up the Remote Registration Tool .
SSO Implementations	Access Manager supports numerous SSO scenarios. See Also: Access Manager Single Sign-On Components
Protocols that secure information exchange on the Internet	This depends on the credential collector you choose. See Also: Table 22-4

Table 14-2 (Cont.) Agent Registration and SSO Support

Oracle Provides	Description
Login and Logout Forms	The location of the login and logout forms depends on the credential collector. See Also: Table 22-4 and Configuring Centralized Logout for Sessions Involving OAM WebGates
Cryptographic keys	One key is generated and used per registered Webgate. See Also: #unique_372/unique_372_Connect_42_BHCGCCJG
Keys storage	<ul style="list-style-type: none"> • Agent side: A per agent key is stored locally in the Oracle Secret Store in a wallet file. • OAM Server side: A per agent key, and server key, are stored in the credential store on the server side.

[Table 14-3](#) provides run time processing information for OAM Agents.



See Also:

[Understanding Credential Collection and Login](#)

Table 14-3 Run Time Processing Overview for Access Manager

Agent Type	Description
WebGates	After installation and registration, WebGates communicate with Access Manager using the OAM Proxy to "sanitize" the request and respond identically for all agents.
Access Clients	<p>Process overview, Authentication Request without OAMAuthnCookie: When a request for a resource protected by Basic authentication scheme comes without an authorization header (credentials)</p> <ol style="list-style-type: none"> 1. WebGate redirects through the front channel to either Embedded or Detached Credential Collector (depending on scheme configuration) to collect credentials. 2. Credential Collector collects user credentials based on the challenge method defined for the authentication scheme. 3. User is authenticated; OAM Proxy (Embedded Collector) or Detached Collector itself (DCC) communicates with the OAM Server through the back channel protocol for the token and returns a response through the front channel with a token issued by the OAM Server. 4. WebGate validates the response, extracts the authentication token issued by the OAM Server, and sets a token in OAMAuthnCookie. 5. WebGate is redirected to the requested resource, with the newly set OAMAuthnCookie attached. 6. WebGate validates the OAMAuthnCookie, performs authorization through the back channel, and serves the page when authorization is successful. <p>Process overview, Basic Authentication: When a request for a resource protected by Basic authentication scheme comes without an authorization header (credentials)</p> <ol style="list-style-type: none"> 1. WebGate responds with <code>WWW-Authenticate</code> header containing the realm mentioned in the authentication scheme with status code 401 (authorization required). 2. Browser client interprets the <code>WWW-Authenticate</code> header and collects credentials from user. 3. Browser client performs request again with authorization header containing credentials. <p>See Also: "About OAM WebGate Configured as a Detached Credential Collector"</p>

14.1.2 About OAM WebGate Configured as a Detached Credential Collector

With Oracle Access Manager, the Embedded Credential Collector (ECC) is the default. The ECC was and is integrated with the OAM Server.

Access Manager also supports the ECC by default. However, Access Manager also enables you to configure a WebGate to use a detached credential collector (DCC). The DCC is considered more secure when compared to the default ECC.

A WebGate configured to act as a DCC is known as an Authenticating WebGate. WebGates that protect resources are known as Resource WebGates.



See Also:

["Configuring OAM WebGate and Authentication Policy for DCC"](#)

14.2 Introduction to Agent Registration

You can use either the Oracle Access Management Console or the remote registration tool for Agent registration and updates. Unless explicitly stated, information in this section applies to agent registration using either of these tools.

This section provides the following details.

- [Keys and Policies Generated during Agent Registration](#)
- [File System Changes and Artifacts for Registered Agents](#)

14.2.1 Keys and Policies Generated during Agent Registration

Administrators must register each Agent to operate with Access Manager. Only registered agents can communicate with an OAM Server, and process information for a user attempting to access a protected resource.

The agent is presumed to reside on the computer hosting the application to be protected. However, it can reside on a proxy Web server and the application on a different host.

An agent key and partner key are created during registration. You can also create policies to protect the application during agent registration. If you choose to automatically create policies during agent registration, a host identifier and Application Domain are created with basic policies and resource definitions. Later on, you can view and manage the Application Domain and policies.

Note:

You can register multiple WebGates or Access Clients under a single host identifier, with the same Application Domain and policies, as follows:

1. When you register a WebGate, allow the process to create a host identifier (a name of your choice), and enable "Auto Create Policies".
2. Register a second WebGate with the same host identifier as Step 1, and clear the "Auto Create Policies" box to eliminate policy creation.

Following a successful registration (using either the console or remote registration tool), the full agent registration appears in the Oracle Access Management Console and is propagated to all Managed Servers in the cluster. [Table 14-4](#) identifies the keys and policies generated during agent registration.

Table 14-4 Keys and Policies Generated During Agent Registration

Keys and Policies	Accessible to	Accessible through
One key per WebGate Agent See Also: "Key Use, Generation, Provisioning, and Storage"	<ul style="list-style-type: none"> • OAM Server 	<ul style="list-style-type: none"> • Client-side: Secure local storage on the client host (a local wallet file) • Server side: The Java Keystore
Partner key for the application	<ul style="list-style-type: none"> • WebGate 	Client-side

Table 14-4 (Cont.) Keys and Policies Generated During Agent Registration

Keys and Policies	Accessible to	Accessible through
<p>Application Domain and default Policies are generated during Agent registration on demand:</p> <ul style="list-style-type: none"> Named for the Agent Populated with default authentication and authorization policies (but not Token Issuance Policies) Identified by the same host identifier that was specified for the Agent during registration 	<ul style="list-style-type: none"> Administrators can view, modify, or remove a registered agent using either the Oracle Access Management Console or custom WLST commands for Access Manager All agent types at run time monitor attempts to access a Web site and use OAM Servers to provide authentication and authorization services before completing the request 	<p>Oracle Access Management Console</p> <p>Policy Configuration Application Domains <i>DomainName</i></p>

14.2.2 File System Changes and Artifacts for Registered Agents

When you register an agent using the Oracle Access Management Console, a new file system directory is created for the Agent on the Oracle Access Management Console host (AdminServer).

This new directory includes generated files for the registered agent, as described in [Table 14-5](#).

Table 14-5 Artifacts Associated with Agent Registration

Registration Artifact	Generated for ...
All WebGates or Access Client ObAccessClient.xml	<p>All WebGates/Access Clients on the console host (AdminServer). During run time, periodic update checks are made. ObAccessClient is updated automatically when a change is discovered.</p> <p>See Also: Properties files generated on the client in this table.</p>
cwallet.sso WebGate only	<p>WebGates, regardless of the transport security mode.</p>
Certificate and password files for secure communication	<p>All WebGates/Access Clients. For example:</p> <ul style="list-style-type: none"> password.xml (nominally encrypted file for Simple Mode Global passphrase) aaa_cert.pem (reserved name for WebGate certificate file, which cannot be changed) aaa_key.pem (reserved name for WebGate key file, which cannot be changed) <p>Cert Mode:</p> <ul style="list-style-type: none"> PEM keystore Alias PEM keystore Alias Password <p>Note: When editing a 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.</p> <p>See: Configuring Access Manager Settings for details about Simple and Cert mode transport security)</p>

Generated or updated artifacts must be copied from the console host (AdminServer) into the agent's installation directory, as shown in [Table 14-6](#).

Table 14-6 Copying Generated Artifacts

Agent Type & Artifacts	Copy Generated Artifacts to Agent Installation Directory ...
ObAccessClient.xml (and WebGate cwallet.sso)	Before agent startup, copy the ObAccessClient file (and cwallet.sso) from the generated location (AdminServer (Console) host) to the agent installation directory.
WebGate or Access Client	See: Registering and Managing OAM Agents

14.3 OAM Remote Registration

As an alternative to using the Console for agent registration, you can use the remote registration utility, oamreg, with Oracle-provided templates.

The user of the remote registration script can be a part of any group that is mapped against the Administrator's Role in the primary user-identity store for Access Manager ([Managing Data Sources](#)).

Secure registration and creation of an Application Domain (as well as Symmetric key generation) is supported using either remote registration mode described in [Table 14-7](#).

Table 14-7 Remote Registration Methods

Method	Description
In-band mode	For Administrators within the network who manage the Web server that hosts the agent can use this mode or the Oracle Access Management Console.
Out-of-band mode	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 Application: One key is generated and used per registered WebGate.

The functionality that are not supported with remote registration are as follows:

- Persistence of the Key and Agent Information
- Generation of Keys used by internal components
- API support for reading Agent information

For more information on the registration modes, see the following sections:

- [Performing In-Band Remote Registration](#)
- [Performing Out-of-Band Remote Registration](#)
- [Updating Agent Configuration Files](#)

[Registering and Managing OAM Agents](#) has additional details.

14.3.1 Performing In-Band Remote Registration

Using the remote registration tool, an in-band Web server Administrator can perform tasks for provisioning an application. Unless explicitly stated, tasks are the same regardless of the type of agent you have protecting resources.

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 Management.

1. Acquire the registration tool as described in "[Acquiring and Setting Up the Remote Registration Tool](#)".
2. Update the input file with unique values for the agent and Application Domain as described in "[Creating Your Remote Registration Request](#)".
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](#)".
4. Validate the configuration as described in "[Validating Remote Registration and Resource Protection](#)".
5. Perform access checks to validate that the configuration is working, as described in "[Verifying Authentication and Access After Remote Registration](#)".

14.3.2 Performing 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.

Following is a brief overview of out-of-band remote registration (when the Agent is 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 registration tool as described in "[Acquiring and Setting Up the Remote Registration Tool](#)".
 - Copy and edit a template to input unique values for the agent and Application Domain as described in "[Creating Your Remote Registration Request](#)".
 - 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 registration tool as described in "[Acquiring and Setting Up the Remote Registration Tool](#)".
 - 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 Out-of-Band Remote Registration](#)" for details:
 - *agentName_Response.xml* is generated for the out of band Administrator to use in Step 3.

- OAM Agents: A modified ObAccessClient.xml file is created (and the WebGate cwallet.sso file), which the out-of-band Administrator can use to bootstrap the WebGate.

WebGates: SSO wallet creation.

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.

Note:

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.

4. **In-band Administrator:** Validates the configuration as described in "[Validating Remote Registration and Resource Protection](#)".
5. **Out-of-band Administrator:** Performs several access checks to validate that the configuration is working, as described in "[Verifying Authentication and Access After Remote Registration](#)".

See Also:

- "[Updating Agent Configuration Files](#)"
- "[Remote Registration Tool, Modes, and Process](#)"

14.3.3 Updating Agent Configuration Files

After a successful registration (or update), you must locate the Agent configuration files on the AdminServer (console) host and copy these to the Agent host.

The artifacts for Agent's registration or update are described in [Table 14-8](#).

Table 14-8 Agent Registration and Configuration Update Artifacts

Artifacts For ...	Description
Simple or Cert mode	If Simple or Cert mode is used, certificate artifacts must also be copied to the Agent host following registration. See Also: Securing Communication
OAM Agents (WebGate/ Access Client)	See Also: Registering and Managing OAM Agents

15

Registering and Managing OAM Agents

You can register and manage WebGates (and the programmatic equivalent, Access Clients) using either the Oracle Access Management Console or the remote registration command-line utility. During registration, you can identify specific applications to be protected by Access Manager policies.

The following topics describe how to register and manage OAM Agents:

- [Before Registering and Managing Agents](#)
- [OAM Agent Registration Parameters in the Console](#)
- [Registering an OAM Agent Using the Console](#)
- [Configuring and Managing Registered OAM Agents Using the Console](#)
- [Remote Registration Tool, Modes, and Process](#)
- [Remote Registration Templates: OAM Agents](#)
- [Performing Remote Registration for OAM Agents](#)
- [Remote Agent Update Modes and Templates](#)
- [Updating Agents Remotely](#)
- [Validating Remote Registration and Resource Protection](#)

15.1 Before Registering and Managing Agents

Ensure that the Oracle Access Management Console host (AdminServer) and a managed OAM Server are running.



See Also:

The following, as needed for your environment.

- [Introduction to Agents and Registration](#)
- [Managing Policies and Application Domains Remotely](#)

15.2 OAM Agent Registration Parameters in the Console

Unless explicitly stated, the information here applies to WebGates, including programmatic Access Clients.

OAM Agent registration parameters topics include:

- [Creating OAM WebGate Page and Parameters](#)
- [User-Defined WebGate Parameters](#)

- [IP Address Validation for WebGates](#)

15.2.1 Creating OAM WebGate Page and Parameters

The `Create OAM ... WebGate` page requests minimal information to streamline registration.

Required details are identified by the asterisk (*).

Figure 15-1 Create OAM WebGate Page

The screenshot shows the 'Create Webgate' page in the Oracle Access Manager console. The page is titled 'Create Webgate' and includes an 'Apply' button in the top right corner. Below the title, there is a brief instruction: 'Use the following screen to register an OAM Agent. Before you register, ensure that at least one OAM Server is running in the same mode as the Agent to be registered.'

The form contains several input fields and options:

- Name:** A required text input field (marked with an asterisk).
- Description:** A text input field.
- Base URL:** A text input field.
- Access Client Password:** A text input field.
- Host Identifier:** A text input field.
- User Defined Parameters:** A large text area for entering parameters.
- Security:** Radio buttons for 'Open' (selected), 'Simple', and 'Cert'.
- Virtual host:** A checkbox.
- Auto Create Policies:** A checked checkbox.
- IP Validation:** An unchecked checkbox.

At the bottom of the page, there are two sections for resource lists:

- Protected Resource List:** Includes an 'Add' and 'Delete' button and a 'Relative URI' input field containing '/**'.
- Public Resource List:** Includes an 'Add' and 'Delete' button and a 'Relative URI' input field with the text 'No data to display'.

Copyright © 2000, 2017, Oracle and/or its affiliates. All rights reserved.

[Table 15-1](#) describes the Create page for OAM WebGates (or Access Clients).

Table 15-1 Elements on Create Pages for OAM Agents

OAM WebGate Element	Description
Name	<p>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.</p> <p>A unique identifying name for each Agent registration is preferred. However:</p> <ul style="list-style-type: none"> • If the Agent Name exists, no error occurs and the registration does not fail. Instead, 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.
Description	A meaningful description of this Agent registration.
Base URL Optional	<p>The host and port of the computer on which the Web server for the WebGate is installed. For example, <code>http://example_host:port</code> or <code>https://example_host:port</code>. The port number is optional.</p> <p>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 Host Identifier element). However, one domain can have multiple Base URLs.</p>
Access Client Password Optional	<p>An optional, unique password for this WebGate, which can be assigned during this registration process.</p> <p>When a registered WebGate connects to an OAM Server, the password is used for authentication to prevent unauthorized WebGates from connecting to OAM Servers and obtaining policy information.</p>
Security	<p>Level of communication transport security between the Agent and the OAM Server (this must match the level specified for the OAM Server):</p> <ul style="list-style-type: none"> • 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 (<code>aaa_key.pem</code>) is encrypted using DES algorithm. The Agent Key Password is saved in obfuscated format in <code>password.xml</code> and is required by the server to generate <code>password.xml</code>. However, this password is not retained by the server. When editing an WebGate registration, <code>password.xml</code> is updated only when the mode is changed from Open to Cert or Simple to Cert. In Cert mode, once generated, <code>password.xml</code> cannot be updated. Editing the Agent Key Password does not result in creation of a new <code>password.xml</code>. <p>Note: For more information on Simple and Cert modes, and private key encryption, see Securing Communication.</p>
Host Identifier	<p>This identifier represents the Web server host. This is automatically seeded with the value in the agent Name field.</p> <p>Note: You can register multiple OAM WebGates (or Access Clients) under a single host identifier with the same Application Domain and policies, as follows:</p> <ol style="list-style-type: none"> 1. When you register a WebGate, allow the process to create a host identifier (a name of your choice), and enable "Auto Create Policies". 2. Register a second WebGate with the same host identifier as Step 1, and clear the "Auto Create Policies" box to eliminate policy creation. <p>See Also: "About Virtual Web Hosting".</p>
User-defined Parameters	<p>Parameters you can enter to enable specific WebGate behaviors:</p> <p>See Also: "User-Defined WebGate Parameters".</p>

Table 15-1 (Cont.) Elements on Create Pages for OAM Agents

OAM WebGate Element	Description
Virtual Host	<p>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.</p> <p>See Also: "About Virtual Web Hosting".</p>
Auto Create Policies	<p>During agent registration, you can have authentication and authorization policies created automatically. This option is checked (enabled) by default.</p> <p>Default: Enabled</p> <p>Shared Registration and Policies: Multiple WebGates (or Access Clients) installed on different Web servers can share a single registration and policies to protect the same resources. This is useful in a high-availability failover environment. To do this:</p> <ol style="list-style-type: none"> 1. WebGate1: Register the first WebGate and enable Auto Create Policies to generate a host identifier (named as you like) and policies. 2. WebGate2: Register the second WebGate, specify the same host identifier as the first WebGate, and disable Auto Create Policies. <p>After registering the second agent, both WebGates use the same host identifier and policies.</p>
IP Validation	<p>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.</p> <p>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 re-authenticate.</p> <p>Default: Disabled</p> <p>See Also: "IP Address Validation for WebGates".</p>
Agent Key Password	<p>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.</p> <p>Note: The Agent Key Password has no relationship to the Access Client Password described earlier within this table.</p> <p>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 <code>openssl</code> or other third-party tools to be placed inside <code>aaa_key.pem</code>. At runtime, WebGate retrieves the key from <code>password.xml</code>, and uses it to decrypt the key in <code>aaa_key.pem</code>.</p> <ul style="list-style-type: none"> • If the key is encrypted, WebGate internally invokes the call back function to obtain the password. • If the key is encrypted and <code>password.xml</code> does not exist, WebGate cannot establish connections with the OAM Server. • If the key is not encrypted, there is no attempt to read <code>password.xml</code>. <p>For more information, see Securing Communication .</p>

Resource Lists

Table 15-1 (Cont.) Elements on Create Pages for OAM Agents

OAM WebGate Element	Description
Protected Resource (URI) List	<p>URIs for the protected application: <code>/myapp/login</code>, for example. Each URI for the protected application should be specified in a new row of the table for the Protected Resource List.</p> <p>Default: <code>/**</code></p> <p>The default matches any sequence of characters within zero or more intermediate levels spanning multiple directories.</p> <p>Add Resources: Each URI should be specified in a new row of the table for the Protected Resource List. Click the + button to add a resource to the Protected Resource List. For instance, if you add <code>/financial</code> (and repeat to add <code>/myfinancial</code>) the following URLs are seeded into the designated policies of the Application Domain when Auto Create Policies is selected):</p> <p style="padding-left: 40px;"><code>/financial</code> yields Resource URL <code>/financial/**</code> <code>/myfinancial</code> yields Resource URL <code>/myfinancial/**</code> <code>/**</code></p> <p>See Also: "Resource URL, Prefixes, and Patterns".</p>
Public Resource (URI) List	<p>Each public application should be specified in a new row of the table for the Public Resource List.</p> <p>Add Resources: Each URI should be specified in a new row of the table for the Public Resource List. Click the + button to add a resource to the Public Resource List. For instance, if you add <code>/people</code> the following URLs are included here and in the Application Domain (when Auto Create Policies is selected):</p> <p style="padding-left: 40px;"><code>/people</code></p> <p>See Also: "Resource URL, Prefixes, and Patterns".</p>
See Also:	Table 15-3

To help streamline WebGate registration, some elements are concealed during the create operation and default values are applied.



Note:

All changes made using the Oracle Access Management Console are taken up without restarting the application server. Changes are reflected automatically after the re-configuration timeout period.

15.2.2 User-Defined WebGate Parameters

Certain supported parameters can be defined by Administrators entering values directly on the WebGate registration page or within the OAM Agent remote registration request template.

[Table 15-2](#) describes supported user-defined parameters. Each parameter can have only one value.

Table 15-2 User-Defined WebGate Parameters

User-Defined WebGate Parameter	Description
ChallengeRedirectMethod	<p>Configure this user-defined authentication POST data preservation parameter for both the embedded credential collector (ECC) and the detached credential collector (DCC).</p> <p>Value: GET POST DYNAMIC</p> <p>Note: Preference is given first to the Authentication Scheme containing this parameter; second to the WebGate providing this user defined parameter. Otherwise, default behavior is Dynamic.</p> <p>See Also: Table 22-25</p>
ChallengeRedirectMaxMessageBytes	<p>Configure this user-defined WebGate parameter to limit the size of the message data received as obrareq.cgi and obrar.cgi. Message data is comprised of query string length, if present (or POST data length, if POST data is present). If message size exceeds this limit, the message is not processed and the existing message is shown in the browser. The event is logged as usual.</p> <p>Default: 8192 bytes</p> <p>Notes:</p> <p>obrareq.cgi is the authentication request in the form of a query string redirected from WebGate to OAM Server.</p> <p>obrar.cgi is the authentication response string redirected from the OAM Server to WebGate.</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>
MaxPostDataBytes	<p>Authentication post-data preservation parameter for both the embedded credential collector (ECC) and the detached credential collector (DCC). This parameter requires a positive integer value that restricts the maximum number of bytes of POST data that is submitted as user credentials and sent to the OAM Server.</p> <p>Default: 8192 bytes</p> <p>Assigning MaxPostDataBytes to a Resource WebGate gives preference to restricting the size of the post data received from the application before forwarding the post data to be preserved.</p> <p>See Also: "Configuring the PasswordPolicyValidationScheme" "Configuring Authentication POST Data Handling" Table 22-25</p>
MaxPreservedPostDataBytes	<p>Configure this user-defined WebGate parameter (or user-defined Authentication Scheme challenge parameter) for authentication POST-data preservation.</p> <p>Default: 8192 bytes</p> <p>Note: Preference is given first to the Authentication Scheme containing this parameter; second to the WebGate providing this user-defined parameter. Otherwise, default behavior is 8192 bytes.</p> <p>This parameter defines the maximum length of POST data that WebGate can preserve. If the size of inbound raw user POST data (or encrypted post data after processing), crosses this limit, POST data is dropped and the existing authentication flow continues. The event is logged as usual.</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 22-25</p>

Table 15-2 (Cont.) User-Defined WebGate Parameters

User-Defined WebGate Parameter	Description
PostDataRestoration	<p>Authentication post-data preservation parameter for both the embedded credential collector (ECC) and the detached credential collector (DCC). This parameter requires a value of <code>true</code> or <code>false</code>.</p> <p>Default: <code>false</code></p> <p>When set to <code>true</code>, WebGate initiates POST data preservation.</p> <p>See Also: "Configuring Authentication POST Data Handling"</p>
serverRequestCacheType ECC Only	<p>Authentication post-data preservation parameter by the embedded credential collector (ECC).</p> <p>This OAM Server parameter in <code>oam-config.xml</code> indicates mechanism to be used to remember the request context. Possible values are <code>FORM</code>, <code>COOKIE</code>, or <code>CACHE</code>.</p> <p>Default: <code>COOKIE</code></p> <p><code>FORM</code> is the required value for POST data preservation.</p> <p>See Also: <code>TempStateMode</code> in Table 22-25. "Configuring Authentication POST Data Handling"</p>
UrlInUTF8Format=true	<p>In an environment that uses Oracle HTTP Server 2, this parameter must be set to <code>true</code> to display latin-1 and other character sets.</p>
ProxySSLHeaderVar=IS_SSL	<p>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 <code>HTTPS</code> rather than <code>HTTP</code>. The proxy ensures that URLs are stored in <code>HTTPS</code> format by setting a custom header variable indicating whether it is servicing an SSL or non-SSL client connection.</p> <p>The value of the <code>ProxySSLHeaderVar</code> parameter defines the name of the header variable the proxy must set. The value of the header variable must be <code>"ssl"</code> or <code>"nonssl"</code>.</p> <p>If the header variable is not set, the SSL state is decided by the SSL state of the current Web server.</p> <p>Default: IS_SSL</p>
client_request_retry_attempts=1	<p>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.</p> <p>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.</p> <p>Default: 1</p> <p>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 <code>client_request_retry_attempts</code> parameter.</p>
InactiveReconfigPeriod=10	<p>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).</p> <p>Default: 10 (minutes)</p>

Table 15-2 (Cont.) User-Defined WebGate Parameters

User-Defined WebGate Parameter	Description
logoutRedirectUrl=	Default = <code>http://OAMServer_host:14200/oam/server/logout</code>
maxAuthorizationResultCacheElements	<p>Max Authorization Results Cache Elements—Number of elements maintained in the Authorization Result Cache. This cache maintains information about authorization results for associated sessions. For example:</p> <pre>maxAuthorizationResultCacheElements=10000</pre> <p>Default = 100000</p>
authorizationResultCacheTimeout	<p>Authorization Results Cache Timeout—Number of elements maintained in the Authorization Result Cache. This cache maintains information about authorization results for associated sessions. For example:</p> <pre>authorizationResultCacheTimeout=60</pre> <p>Default, if no time is specified = 15 (seconds)</p> <p>Note: Authorization Results Cache Timeout is not set by default.</p> <p>With the cache enabled, the first request result persists for the cache duration. This magnifies the effect causing a brief time delay. For example suppose you set an authentication policy Response and set a custom session attribute <i>exmpl:sample</i>. The corresponding authorization policy Response returns this as <code>HEADER SESSION_ATTR_EXMPL=sample</code>. When a user access the URL protected by these policies, the header comes after a few refreshes. Initially, however, the value might not be found.</p> <p>A value of 0 disables the cache. With no cache, it takes two requests for the header response to be filled. The first sets the session variable used, the second uses the session variable. Oracle recommends that you do not set a Response value in the same authorization request that triggers it.</p>
UniqueCookieNames	<p>Controls WebGate cookie name format:</p> <ul style="list-style-type: none"> • Legacy: Backward compatible and supports the colon (:) character. The format is <code><prefix>_<host>:<port>_<suffix></code> • Enabled: Default and enable rfc2109-compliant cookie name restriction. The format is <code><prefix>_<host>_<port>_<suffix></code> • Disabled: no <code><host>[:/<_>]<port></code> is added in the cookie name. The format is <code><prefix>_<suffix></code>. • Any other value is treated as the default enabled format. <code><prefix>_<host>_<port>_<suffix></code>

Table 15-2 (Cont.) User-Defined WebGate Parameters



User-Defined WebGate Parameter	Description
EnableFIPSMODE=true	<p>Enables FIPS mode for WebGates in CERT security mode.</p> <p>In FIPS mode, a Federal Information Processing Standard (FIPS) 140-2 certified security module is used by the WebGates to establish SSL/TLS connections.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> Note:</p> <p>For WebGates, only OAP over TCP is FIPS compliant.</p> </div> <p>By default, FIPS mode is disabled. From WebGate 12c PS1 release onwards, you can enable the FIPS mode by setting <code>EnableFIPSMODE=true</code>.</p> <p>Ensure the root certificate CA that is added in <code>cwallet.sso</code> is not MD5. If the root CA is MD5, the TLS handshake fails and the following error is logged: <code>Error: TLS Handshake unsuccessful</code></p> <p>If the FIPS mode is initialized successfully, the following message is logged in the <code>webgate.log</code> file: "Fips Mode Initialized Successfully...."</p> <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> Note:</p> <p>If the FIPS mode fails to initialize due to missing or corrupted libraries, the WebGate switches to non-FIPS Mode and the following message is logged in the <code>webgate.log</code> file: <code>Missing/Corrupted FIPS Libraries at the specified location path. Hence proceeding without FIPS Mode.</code></p> </div>
OAM WebGate only	
SetKeepAlive	<p>By default, SetKeepAlive is ON. In this case, a first keep-alive message will be sent after the default idle time of 2 minutes. To change this behavior, set a new value for the parameter. If SetKeepAlive=Off, the feature is disabled and no keep-alive messages will be sent. If SetKeepAlive=x (where x is some positive integer value), the keep-alive message will be sent after the channel is idle for x minutes. Any firewall or load balancer should be configured to forward the TCP/IP keep-alive messages to the actual end parties (front-ending Access Manager server).</p> <p>A programmatic way to change the idle time is implemented for Linux64, Linux32, and Windows32 WebGates. This is not possible on SPARC Solaris platforms; in that case, SetKeepAlive is enabled and the idle time out for Keep alive must be set manually by the system administrator.</p>
filterOAMAuthnCookie	<p>For WebGate, a user-defined parameter (<code>filterOAMAuthnCookie</code> (default <code>true</code>)) 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 <code>filterOAMAuthnCookie</code> parameter to <code>false</code>.</p>

Table 15-2 (Cont.) User-Defined WebGate Parameters

User-Defined WebGate Parameter	Description
ssoCookie	<p>Controls the OAMAuthnCookie cookie.</p> <p>Default: ssoCookie=httponly ssoCookie=Secure</p> <p>Disable either setting: ssoCookie=disablehttponly ssoCookie=disableSecure</p> <p>Note: These parameters are configured differently depending on your credential collector configuration.</p> <ul style="list-style-type: none"> For detached credential collector-enabled OAM WebGates, set these parameters directly in the agent registration page. For non-DCC agents (Resource WebGates), these parameters are configured through user-defined challenge parameters in authentication schemes. <p>See Also: Table 22-25 "Configuring Challenge Parameters for Encrypted Cookies" "Configuring OAM WebGate and Authentication Policy for DCC"</p>
miscCookies	<p>Controls other miscellaneous Access Manager internal cookies. By default, httponly is enabled for all other (miscellaneous) cookies.</p> <p>Default: miscCookies=httponly miscCookies=Secure</p> <p>Disable either setting: miscCookies=disablehttponly miscCookies=disableSecure</p> <p>Note: These parameters are configured differently depending on your credential collector configuration.</p> <ul style="list-style-type: none"> For detached credential collector-enabled WebGates, set these parameters directly in the agent registration page. For non-DCC agents (Resource WebGates), these parameters are configured through challenge parameters of the same name. <p>See Also: Table 22-25 "Configuring Challenge Parameters for Encrypted Cookies" "Configuring OAM WebGate and Authentication Policy for DCC"</p>
OAMAuthAuthenticationServiceLocation WebGate non-browser client functionality	<p>Activates non-browser client functionality and defines the location of the authentication service.</p> <p>OAMAuthUserAgentPrefix=<i>prefix string that acts as the prefix for the "user-agent" HTTP header value.</i></p> <p>For example, to activate this functionality for Identity Connect:</p> <pre>OAMAuthAuthenticationServiceLocation=https://login.example.com/nbc</pre> <p>Non-browser client functionality is deactivated if the parameter is omitted (or is provided with no value).</p>

Table 15-2 (Cont.) User-Defined WebGate Parameters

User-Defined WebGate Parameter	Description
OAMAuthUserAgentPrefix WebGate non-browser client functionality	<p>Activates non-browser client functionality and defines the string that acts as a prefix for the "user-agent" http header value.</p> <p><code>OAMAuthAuthenticationServiceLocation=full URL location of the NBC authentication service.</code></p> <p>For example, to activate this functionality for Identity Connect:</p> <pre>OAMAuthUserAgentPrefix=NBC</pre> <p>Non-browser client functionality is deactivated if this parameter is omitted (or is provided with no value).</p>
RequestContextCookieExpTime	<p>Controls the time (in seconds) to expire OAMRequestContext cookie. Configuring the cookie lifetime is an optional control for deployments with a critical need to handle situations where the cookies could proliferate.</p> <p>Default: not set</p> <p>In the Resource WebGate registration, add this parameter to expire the OAMRequestContext cookie in the configured number of seconds using the "Max-Age" directive on all but IE browsers (default 5 minutes).</p> <p>Note: For Internet Explorer only, this parameter requires a time sync between the browser and Web server hosts because IE uses the "Expires" directive to expire the cookie with an absolute time. However, on IE browsers, when this parameter is not set, OAMRequestContext cookie is a transient session cookie. On other (non-IE) browsers, the cookie is persistent and expires based on the time set using the "Max-Age" directive.</p> <p>See Also: OAMRequestContext in Table 21-6</p>
ProxyTrustedIPList	Multi-valued parameter that holds the list of IP addresses of the trusted proxies or load balancers. See ProxyTrustedIPList .
ProxyRemoteIPHeaderVar	Specifies the name of the HTTP header that contains the list of IP addresses. See ProxyRemoteIPHeaderVar .

15.2.3 IP Address Validation for WebGates

IP address validation is a function that determines if a client's IP address is the same as the IP address stored in the cookie generated for single sign-on. The `IPValidation` parameter turns IP address validation on and off; it is a WebGate specific parameter found in the WebGate profile.

If `IPValidation` is `true`, the IP address stored in the cookie must match the client's IP address, otherwise, the SSO cookie (`#unique_372/unique_372_Connect_42_BHCGCCJG`) is rejected and the user must reauthenticate. By default, `IPValidation` is `false`. The following is true in regards to enabling and disabling IP Validation.

- Enabling IP Validation on the WebGate automatically enables it on the OAM server side; this can be verified in the Access Manager settings.
- Disabling IP Validation on the WebGate will not disable it on the OAM server.
- IP Validation on the OAM server side should be disabled manually, if and only if it is disabled on all the WebGates.

- When IP Validation is enabled on the WebGate side, server side IP Validation should never be turned off.

**Note:**

Access Manager now supports Internet Protocol version 6 (IPv6) as well as IPv4.

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 SSO cookie. However, Oracle recommends that you keep IP validation on whenever possible. For WebGate profile configuration information, see [Viewing or Editing an OAM Agent Registration Page in the Console](#). Additional details are in the following sections.

- [IP Validation Exceptions List](#)
- [IP Validation in Load Balanced Environments](#)

15.2.3.1 IP Validation Exceptions List

The IP Validation parameter can cause problems with certain Web application deployments.

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 from using the cookie. The IP Validation Exceptions parameter lists IP addresses that are exceptions to this process. When `IPValidation` is `true`, the IP address is compared to the IP Validation Exceptions List. If the address is found on the list, it does not need to match the IP address stored in the cookie.

You can add as many IP addresses as needed to the Exceptions list - the actual IP addresses of the client and not the IP addresses stored in the `ObSSOCookie` SSO cookie. If an SSO cookie is from one of the exception IP addresses, the Access System ignores the address stored in the SSO cookie for validation. (The IP addresses in the IP Validation Exceptions List can be used when the IP address in the cookie is for a reverse proxy.)

15.2.3.2 IP Validation in Load Balanced Environments

In the case of (proxy servers or) a load balancer, Oracle Access Manager cannot enforce true IP validation because an attacker can use the IP address defined in the exception list. Web applications managed thusly typically change the user's IP address (substituting the IP address of the proxy or load balancer). This can prevent single sign-on using the SSO cookie.

A load balancer adds an "X-forwarded-for" header variable to incoming HTTP requests, containing a comma-space-separated list of the original IP number of the requester. Consider the following example in which the request passed proxy1, proxy2 and proxy3 (proxy3 appears as the remote address of the request). The last IP address is always the IP address that connects to the last proxy.

```
X-Forwarded-For: client1, proxy1, proxy2
```

The trust list will be referenced to look up each IP address from the header, starting with the right-most value. The left-most IP address being the farthest downstream client and each successive proxy that passed the request (adding the IP address from which it received the request).

Within the specified order, the first IP address that does not match any of those in the trusted list is treated as an apparent client IP (defined as the IP address of the initiator of the connection to the furthest node along the communication path that can be trusted). Additionally:

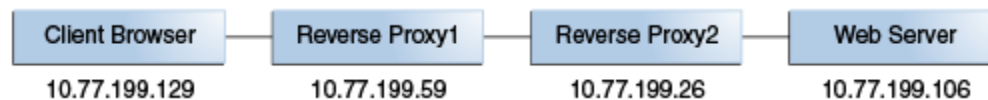
- When all IP addresses from the header (starting from the right side) match with entries in the trusted list, WebGate chooses the end client IP (the left most IP address in the header).
- When the IP address is determined, WebGate obtains a session token that contains the apparent client IP address and IP validation is evaluated by comparing the IP address against the address in the session token.
- When the IP validation feature is enabled within a load balanced deployment, authentication (session creation) and authorization is done by the WebGate with this feature; otherwise the authenticated user must re-authenticate. When WebGate searches for the particular HTTP header, the search is case-insensitive. For example, "X-Forwarded-For" and "X-FORWARDED-FOR" are treated the same.

15.2.3.2.1 ProxyTrustedIPList

`ProxyTrustedIPList` is a user defined, multi-valued WebGate parameter that holds the list of IP addresses for the trusted proxies or load balancers.

The values are space separated. The IP addresses in the IP Validation Exceptions List can be used when the IP address in the cookie is for a reverse proxy.

Figure 15-2 Load Balanced Deployment



In [Figure 15-2](#), the end user's HTTP request passes through REVERSEPROXY1 and REVERSEPROXY2 to reach the actual Web server. In this case, the IP addresses of REVERSEPROXY1 and REVERSEPROXY2 should be added in the `ProxyTrustedIPList` list as follows:

```
ProxyTrustedIPList=10.77.199.59 10.77.199.26
```

 **Note:**

In a centralized authentication deployment, if any Resource WebGate (RWG) or Authentication WebGate (AWG) is behind a proxy, the IP addresses of all intermediaries must be configured (in the `ProxyTrustedIPList` parameter) in the profile of the WebGate behind the proxy. Otherwise, IP validation failures can occur.

15.2.3.2.2 ProxyRemoteIPHeaderVar

The ProxyRemoteIPHeaderVar parameter specifies the name of the HTTP header that contains the list of IP addresses.

If this parameter is not provided, the default header X-Forwarded-For is used. This parameter can be configured like any other user-defined parameter in a WebGate profile. For example, in the deployment described in [ProxyTrustedIPList](#), "X-FORWARDED-FOR" and other headers that come to the Web server take the following form.

```
HTTP_X_FORWARDED_FOR="10.77.199.129, 10.77.199.59"  
REMOTE_ADDR="10.77.199.26"
```

15.3 Registering an OAM Agent Using the Console

The registration procedure for WebGate or programmatic Access Client is the same. You can register an OAM-type agent before you deploy it.

Users with valid Administrator credentials can perform the following task to register a WebGate using the Oracle Access Management Console.

When the Oracle Access Manager (OAM) WebGate component is required to be used in a Highly Available (HA) environment to eliminate a single point of failure and distribute the workload via a load balancer (LBR), the OAM WebGate component only has to be registered once. The Resulting Artifacts will be used by all of the OAM WebGates behind the LBR.

See Also:

- [OAM Agent Registration Parameters in the Console](#)
- [Configuring Oracle HTTP Server WebGate for Oracle Access Manager in *Installing WebGates for Oracle Access Manager*](#)

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 [Securing Communication](#) .

Note:

You use the same procedure to register a programmatic Access Client. The version is the same as the SDK used to create the Access Client.

Before you begin, confirm that at least one OAM Server is running in the same mode as the agent to be registered.

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.

2. In the Application Security console, select **Create WebGate** from the **Agents** menu.
3. On the **Create WebGate** page, enter required details (those with an *) to register this Agent.
4. **Protected Resource List:** In this table, enter individual resource URLs to be protected by this Agent, as shown in [Table 15-1](#).
5. **Public Resource List:** In this table, enter individual resource URLs to be public (not protected), as shown in [Table 15-1](#).
6. **Auto Create Policies:** Check to create a fresh Application Domain and policies (or clear and use the same host identifier as another WebGate and share policies ([Table 15-1](#))).
7. Click **Apply** to submit the registration.
You may also close the page without applying changes, if applicable.
8. Click the **Download** button to download the generated artifacts.
Downloaded artifacts are located in the `$DOMAIN_HOME/output/$Agent_name` folder.
9. Copy the artifacts as follows (or install WebGate with the same specifications, then copy artifacts), including any Simple or Cert mode files. For example, Open mode files include:

Agent & Artifacts	Artifacts
WebGate/Access Client ObAccessClient.xml and cwallet.sso	From the AdminServer (Console) host: <code>\$DOMAIN_HOME/output/\$Agent_Name/</code> To the Agent host: <code>\$1gWG_install_dir/WebGate/config</code>

10. **Verify Registration:** These are similar to steps in "[Validating Agent Registration using the Oracle Access Management Console](#)".
 - a. Under Agents in Application Security, search and confirm the Agent name is listed.
 - b. Confirm the Agent's page contains the appropriate information.
 - c. **Auto Create Policies:** Confirm the Application Domain was generated, the host identifier was created for the application, and that resources were created in the Application Domain and associated with the host identifier.
 - d. Perform further tests, as described in "[Verifying Authentication and Access After Remote Registration](#)".
11. Proceed as needed for your deployment:
 - [Configuring and Managing Registered OAM Agents Using the Console](#)
 - [Managing Access Manager SSO, Policies, and Testing](#)

15.4 Bulk Updates to WebGates

Multiple WebGate profiles can be updated simultaneously by grouping common parameters of WebGates into a WebGate template.

Topics

- [Updating Multiple WebGate Profiles](#)
- [WLST Commands for Bulk Updates to WebGate Profiles](#)

15.4.1 Updating Multiple WebGate Profiles

Use WebGate template to group WebGates sharing common parameters and values. You can update the WebGate template using WLST commands. The command updates the parameters across all WebGates associated with the specified WebGate template.

To update multiple WebGates at the same time:

1. Create a WebGate template and map all WebGates with common parameters to that template. See [Creating a WebGate Template and Mapping WebGates to that Template](#) for more information.
2. Run the [updateWebgateTemplateParams](#) command with the required parameters. For example, to update `ipValidation` exceptions across multiple WebGates, specify the parameter and its corresponding values in the command:

```
updateWebgateTemplateParams (webgateTemplateName="TemplateA",
webgateParamsList="ipValidation,ipValidation exceptions",
webgateValuesList="1,192.168.1.3:1.2.4.6:255.255.255.1:234.12.12.13",
batchSize="20")
```

This command updates the `ipvalidation` exception parameter with the IP addresses provided in the command across all WebGates that are associated with `TemplateA`.

The `updateWebgateTemplateParams` command does not remove existing parameters and values. To remove any of the parameters, use the [removeWebgateTemplateParams](#) command.

15.4.1.1 Creating a WebGate Template and Mapping WebGates to that Template

Use the `createWebgateTemplate` and `updateWebgateTemplateToWebgateMapping` WLST commands to create a WebGate template and map WebGates to that template.

1. Create a WebGate template by running the [createWebgateTemplate](#) WLST command.

```
createWebgateTemplate (webgateTemplateName="TemplateA",webgateParamsList="cook
ieSessionTime,maxConnections",webgateValuesList="24,5")
```

2. Associate WebGates to this template using the [updateWebgateTemplateToWebgateMapping](#) command.

```
updateWebgateTemplateToWebgateMapping( webgateTemplateName="TemplateA",
webgateIdsList="IAMSuiteAgent", isWebgateIdsListAFilePath = "False")
```

To use a file containing a list of WebGate Ids, set the `isWebgateIdsListAFilePath` to `true` and specify the file path in the `isWebgateIdsListAFilePath` argument.

```
updateWebgateTemplateToWebgateMapping( webgateTemplateName="TemplateA",
webgateIdsList="/tmp/test.txt", isWebgateIdsListAFilePath="True",
batchSize="20")
```

15.4.2 WLST Commands for Bulk Updates to WebGate Profiles

The following WebLogic Scripting Tool (WLST) commands support bulk updates to WebGate profiles. More information in the following sections.

- [createWebgateTemplate](#)
- [updateWebgateTemplateToWebgateMapping](#)
- [updateWebgateTemplateParams](#)
- [showWebgateTemplate](#)
- [removeWebgateTemplateParams](#)
- [rollbackWebgatesToPreviousState](#)

15.4.2.1 createWebgateTemplate

The `createWebgateTemplate` command enables you to create a WebGate template with the parameters and values specified within the command.

Description

The `createWebgateTemplate` command creates a WebGate template with the specified parameters and values.

Syntax

```
createWebgateTemplate(webgateTemplateName="<NameOfTemplate>",  
webgateParamsList="<webgateParamsList>", webgateValuesList="<webgateValuesList>")
```

Arguments	Definition
<code>webgateTemplateName</code>	Specifies the name of the WebGate template to be created.
<code>webgateParamsList</code>	A comma-separated list of WebGate parameters that needs to be added to the WebGate template.
<code>webgateValuesList</code>	Specifies the values corresponding to the <code>webgateParamsList</code> .

Example

This example illustrates the use of the `createWebgateTemplate` command.

```
createWebgateTemplate(webgateTemplateName="TemplateA",  
webgateParamsList="cookieSessionTime, maxConnections", webgateValuesList="24,5")
```

15.4.2.2 updateWebgateTemplateToWebgateMapping

The `updateWebgateTemplateToWebgateMapping` command enables you to associate WebGates with a WebGate template.

Description

The `updateWebgateTemplateToWebgateMapping` command enables you to associate all the WebGates having common parameters with a WebGate template, using WebGate IDs. This command overrides the existing parameters and values.

Syntax

```
updateWebgateTemplateToWebgateMapping (webgateTemplateName="<NameOfTemplate>",
webgateIdsList="<webgateIdsList>", webgateIdsExclusionList="<Webgate1,
Webgate2, ...>", [isWebgateIdsListAFilePath="False"], [batchSize="<batchSize>"])
```

Argument	Definition
webgateTemplateName	Specifies the name of the WebGate template.
webgateIdsList	A comma-separated list of WebGate IDs that needs to be associated with the WebGate template. Supports a file containing a list (separated by new-line) of WebGate IDs.
webgateIdsExclusionList	Specifies the IDs of the WebGates that need to be excluded from the WebGate template.
isWebgateIdsListAFilePath	[Optional] Defaults to <code>False</code> . Set this value to <code>True</code> to specify a file containing the WebGate IDs.
batchSize	[Optional] Specifies the number of WebGate instances to be processed per thread. Default value is 10.

Example

These examples illustrates the use of `updateWebgateTemplateToWebgateMapping` command.

```
updateWebgateTemplateToWebgateMapping (webgateTemplateName="TemplateA",
webgateIdsList="/tmp/test.txt", webgateIdsExclusionList="Webgate1,Webgate2",
isWebgateIdsListAFilePath = "True", batchSize="20")
```

```
updateWebgateTemplateToWebgateMapping (webgateTemplateName="TemplateA",
webgateIdsList="IAMSuiteAgent", isWebgateIdsListAFilePath = "False")
```

15.4.2.3 updateWebgateTemplateParams

The `updateWebgateTemplateParams` command enables you to update an existing WebGate template with the common parameters and values specified within the command.

Description

The `updateWebgateTemplateParams` command updates the WebGate template with the specified parameters and values that are common across the WebGates of the specified template type. You can use this command to update the WebGate template with additional parameters. This command does not remove the existing parameters. You must use the `removeWebgateTemplateParams` command to remove the parameters from the template.

Syntax

```
updateWebgateTemplateParams (webgateTemplateName="<NameOfTemplate>",
webgateParamsList="<webgateParamsList>",
webgateValuesList="<webgateValuesList>", [batchSize="<batchSize>"])
```


Argument	Definition
webgateTemplateName	[Mandatory] Specifies the name of the WebGate template that needs to be updated. If the template does not exist, a new WebGate template is created with the specified name.
webgateParamsList	List of WebGate parameters associated with the WebGate template. Specify the parameters in the format: <code>webgateParamsList="Parameter1, Parameter2, ..."</code> . Specify the user defined parameters in the format: <code>userdefinedparameters/[parameters]</code> .
webgateValuesList	Specifies the values corresponding to the parameters in the <code>webgateParamsList</code> . Specify the values for the parameters in the format: <code>webgateValuesList="ParameterValue1, ParameterValue2, ..."</code> . Specify the IP for <code>IPValidationExceptions</code> in the following format, for example: <code>192.168.1.1:255.255.255.1</code> . Use a colon (:) to separate the IP values.
batchSize	[Optional] Specifies the number of WebGate instances to be processed per thread of the <code>ThreadPool</code> . Default value is 10. Specify the size in the following format, for example <code>batchSize="20"</code>

Example

This example illustrates the use of `updateWebgateTemplateParams` command.

```
updateWebgateTemplateParams (webgateTemplateName="TemplateA",
webgateParamsList="idleSessionTimeout", webgateValuesList="3650", batchSize="20")
```

15.4.2.4 removeWebgateTemplateParams

The `removeWebgateTemplateParams` command removes the specified parameters from the WebGate template.

Description

The `removeWebgateTemplateParams` command removes the specified parameters from the WebGate template.

Syntax

```
removeWebgateTemplateParams (webgateTemplateName="<NameOfTemplate>",
webgateParamsList="<webgateParamsList>")
```

Arguments	Definition
webgateTemplateName	Specifies the name of the WebGate template.
webgateParamsList	A comma-separated list of WebGate parameters that needs to be removed from the WebGate template.

Example

This example illustrates the use of the `removeWebgateTemplateParams` command.

```
removeWebgateTemplateParams (webgateTemplateName="webgateTemplateName1", webgateParamsList="idleSessionTimeout")
```

15.4.2.5 rollbackWebgatesToPreviousState

The `rollbackWebgatesToPreviousState` command returns the WebGate profiles to its previous successful state.

Description

If an update to the WebGate template fails, you can revert all the WebGate information to its previous values (successful state before running the command) using the `rollbackWebgatesToPreviousState` command.

Syntax

```
rollbackWebgatesToPreviousState()
```

There are no arguments for this command

Example

This example illustrates the use of the `rollbackWebgatesToPreviousState` command.

```
rollbackWebgatesToPreviousState()
```

15.4.2.6 showWebgateTemplate

The `showWebgateTemplate` command displays the metadata of the specified WebGate template.

Description

The `showWebgateTemplate` command can be used to view the current property and values of an existing Webgate Template. This command also lists the WebGates associated with that template.

Syntax

```
showWebgateTemplate (webgateTemplateName="<NameOfTemplate>")
```

Arguments	Definition
webgateTemplateName	[Mandatory] Specifies the name of the WebGate template. The command displays the properties and values for this template.

Example

This example illustrates the use of the `showWebgateTemplate` command.

```
showWebgateTemplate (webgateTemplateName="DemoTemplate")
```

15.5 Configuring and Managing Registered OAM Agents Using the Console

This section provides the following topics to help you manage registered WebGates:

- [Registered OAM Agent Configuration Parameters in the Console](#)
- [WebGate Search Controls](#)
- [Viewing or Editing an OAM Agent Registration Page in the Console](#)
- [Deleting OAM Agent Registration Using the Console](#)

15.5.1 Registered OAM Agent Configuration Parameters in the Console

Whether you registered the agent using the Oracle Access Management Console or the remote registration utility, you can view the full agent configuration page in the console

[Figure 15-3](#) shows the OAM WebGate Page with the default values.

Figure 15-3 Expanded OAM WebGate Page with Defaults

Launch Pad SSO Agents x accessgate-oi x

Access Manager >

accessgate-oi Webgate Apply Download

Version: OAM Webgate
 Name: accessgate-oi
 Description:
 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: 80
 * Failover Threshold: 1
 * AAA Timeout Threshold: 5
 * Preferred Host: IAMSuiteAgent
 Logout URL:
 Logout Callback URL: /oam_logout_success
 Logout Redirect URL: http://slc12bx.us.oracle.com

Logout Target URL:
 Deny On Not Protected:
 User Defined Parameters: maxSessionTimeUnits=minutes, client_request_retry_attempts=1, proxySSLHeaderVar=IS_SSL, inactiveReconficPeriod=10
 * Sleep for (Seconds): 60
 Cache Pragma Header: no-cache
 Cache Control Header: no-cache
 Debug:
 IP Validation:
 Allow Management Operations:
 Allow Token Scope Operations:
 Allow Master Token Retrieval:
 Allow Credential Collector Operations:
 IIS Impersonation User:
 IIS Impersonation Password:

Server List:

Primary Server List + Add X Delete

Access Server	Host Name	Host Port	Max Connections
oam_ser	slc12bx.us...	5575	1

Secondary Server List + Add X Delete

Access Server	Host Name	Host Port	Max Connections
No data to display			

Copyright © 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Note:

Most elements on the agent's page are the same as those you define when using the remote registration tool with the expanded OAM template. ObAccessClient.xml is populated with values after agent registration or modification, regardless of the method you use.

Table 15-3 describes elements on an expanded registration. Additional settings revealed here are used by the OAM Proxy.

Table 15-3 Elements on Expanded OAM WebGate/Access Client Registration Pages

Element	Description
Name	See: Table 15-1 .
Version	
Description	See Also: " User-Defined WebGate Parameters "
Access Client Password Security	See Also: " IP Address Validation for WebGates ".
User-defined Parameters	
IP Validation	
State	Specifies whether this registration is enabled or disabled.
Only in the console.	Default = Enabled
Max Cache Elements	<p>Number of elements maintained in the cache. Caches are the following:</p> <ul style="list-style-type: none"> • 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. • (OAM 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. <p>The value of this setting refers to the maximum consolidated count for elements in these caches.</p> <p>Default = 100000</p>
Cache Timeout (seconds)	<p>Amount of time cached information remains in the WebGate caches (Resource to Authentication Scheme, Authentication Schemes, and OAM WebGate only Resource to Authorization Policy) when the information is neither used nor referenced.</p> <p>Default = 1800 (seconds)</p>
Max Connections	<p>The maximum number of connections that this WebGate 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.</p> <p>Default = 1</p>
Max Session Time (hours)	<p>Maximum time to keep network connections from this WebGate to the OAM Server alive. After elapsed time, all the WebGate to OAM Server network connections will be shutdown and replaced with new ones. The unit is based on the <code>maxSessionTimeUnits</code> user-defined parameter which can be 'minutes' or 'hours'. When <code>maxSessionTimeUnits</code> is not defined, the unit is defaulted to 'hours'.</p>
Failover Threshold	<p>Number representing the point when this WebGate opens connections to a Secondary OAM Server.</p> <p>Default = 1</p> <p>For example, if you type 30 in this field and the number of connections to primary OAM Server falls to 29, this Agent opens connections to secondary OAM Server.</p>

Table 15-3 (Cont.) Elements on Expanded OAM WebGate/Access Client Registration Pages

Element	Description
AAA Timeout Threshold	<p>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.</p> <p>Default = -1 (default network TCP/IP timeout is used)</p> <p>If using a simple mode WebGate, you can improve the response time of the OAM login page by changing the <code>aaaTimeoutThreshold</code> time parameter in the WebGate profile from -1 to 10.</p> <p>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.</p> <p>For example, suppose a WebGate 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 WebGate 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 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.</p> <p>When finding new connections, WebGate 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 Agent still tries the primary OAM Server first. As a result, the Agent cannot send requests to an OAM Server for a period greater than twice the setting in the OAM Server Timeout Threshold.</p> <p>If the OAM Server takes longer to service a request than the value of the timeout threshold, the 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 Agent can continue to retry the request until the OAM Server is shut down.</p>
ServerConnectionReadTimeout	<p>This parameter can be configured in the ASDK Agent User Defined Parameters section, for further timeout fine-tuning. This setting can be configured for TCP read timeout if required. The read timeout is the timeout on waiting to read data. Specifically, if the server fails to send a byte <i>n</i> seconds after the last byte, a read timeout error will be raised.</p>
poolTimeOut	<p>This parameter can be configured in the ASDK Agent User Defined Parameters section. <code>poolTimeout</code> is the maximum time a request thread will wait to get a connection from the connection pool, before throwing an exception. The default is 30 seconds.</p>

Table 15-3 (Cont.) Elements on Expanded OAM WebGate/Access Client Registration Pages

Element	Description
Preferred Host	<p>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.</p> <p>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.</p> <p>Defaults to Name (of WebGate registration)</p>
User Defined Parameters	See Also: " User-Defined WebGate Parameters " and Tuning Database Parameters in the <i>Fusion Middleware Performance and Tuning Guide</i> .
Logout URL OAM WebGates	<p>The Logout URL triggers the logout handler, which removes the cookie (OAMAuthCookie for OAM WebGates) and requires the user to re-authenticate the next time the user accesses a resource protected by Access Manager.</p> <p>Default = [] (not set)</p>
Additional Logout for OAM WebGate Only	<p>For OAM WebGate single sign-off behavior, specific logout elements and values automate the redirect to a central Logout URL, callback URL, and end_URL.</p> <p>See Also: Table 27-2</p>
Logout Callback URL OAM WebGate only	<p>The URL to <code>oam_logout_success</code>, which clears cookies during the callback. This can be a URI format without <code>host:port</code> (recommended), where the OAM Server calls back on the <code>host:port</code> of the original resource request. For example:</p> <p>Default = <code>/oam_logout_success</code></p> <p>This can also be a full URL format with a <code>host:port</code>, where OAM Server calls back directly without reconstructing callback URL.</p> <p>Note: In the remote registration template this parameter is named <code>logoutCallbackUrl</code> (Table 15-10).</p> <p>See Also: Table 27-2</p>
Logout Redirect URL OAM WebGate only	<p>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:</p> <p>Default = <code>http://OAMServer_host:14200/oam/server/logout</code></p> <p>See Also: Table 27-2</p>
Logout Target URL OAM WebGate only	<p>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.</p> <p>Default: <code>end_url</code></p> <p>Note: The <code>end_url</code> value is configured using <code>param.logout.targeturl</code> in <code>jps-config.xml</code>.</p> <p>See Also: Table 27-2</p>
Sleep for (seconds)	<p>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.</p> <p>Default: 60 (seconds)</p>

Table 15-3 (Cont.) Elements on Expanded OAM WebGate/Access Client Registration Pages

Element	Description
Cache Pragma Header Cache Control Header WebGate only (not Access Clients)	<p>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.</p> <p>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.</p> <p>Defaults: no-cache</p> <p>Note: Browsers may store a local cached copy of content served by an OAM protected resource. Some browsers, including Internet Explorer, cache content accessed via HTTPS which may be retrieved by other users who have access to the same computer at a future time. Please ensure that the Cache Directives are set based on the sensitivity of the application content.</p> <p>See Also: Tuning Oracle HTTP Server in <i>Tuning Performance</i></p>
Debug	Debugging can be enabled or not.
Deny on Not Protected WebGates only (not Access Clients)	<p>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.</p> <ul style="list-style-type: none"> • OAM WebGate: Always enabled, and cannot be changed <p>Important: Deny on Not Protected overrides Host Identifiers and Preferred Host. Oracle recommends enabling Deny on Not Protected. Otherwise security holes can occur in large installations with multiple Host Identifiers, virtual hosts, and other complex configurations.</p>
Allow Management Operations	<p>This Agent Privilege function enables the provisioning of session operations per agent, as follows:</p> <ul style="list-style-type: none"> • Terminate session • Enumerate sessions • Add or Update attributes for an existing session • List all attributes for a given session ID or read session <p>Default: Disabled</p> <p>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.</p>
OAM WebGate only	
Allow Credential Collector Operations	<p>Activates WebGate detached credential collector functionality for simple-form or dynamic multi-factor authentication.</p> <p>Default: Disabled</p> <p>See Also: "Configuring OAM WebGate and Authentication Policy for DCC"</p>
Allow Master Token Retrieval	Allows the ASDK code to retrieve the OAM_ID cookie.

Table 15-3 (Cont.) Elements on Expanded OAM WebGate/Access Client Registration Pages

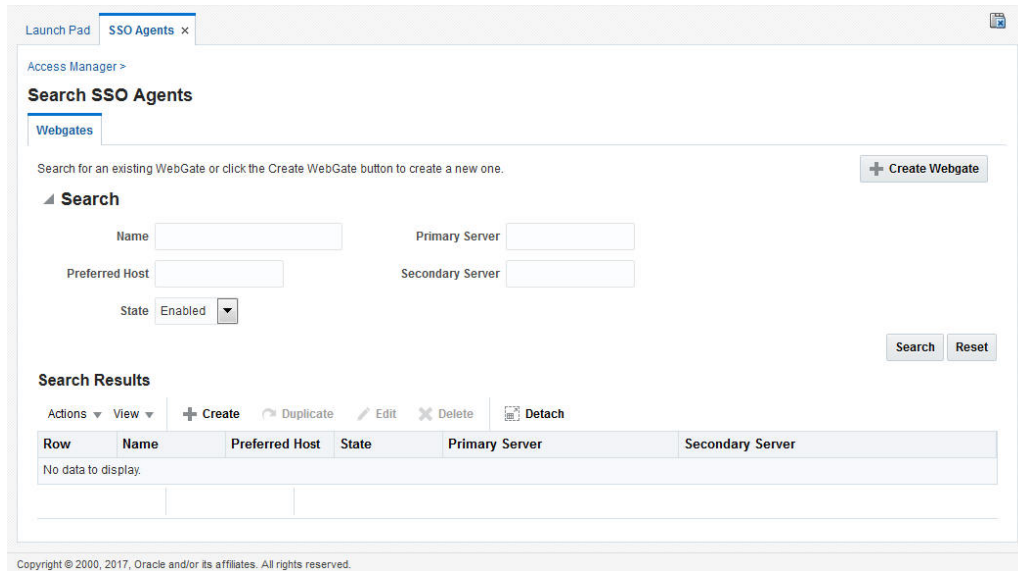
Element	Description
Allow Token Scope Operations	Allows the ASDK code to scope the OAM_ID cookie to the domain level instead of host level.
Primary Server List	Identifies Primary Server details for this Agent. The default is based on the OAM Server: <ul style="list-style-type: none"> • Server Name • 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: <ul style="list-style-type: none"> • 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).)
Token Validity Period	The duration in seconds that the WebGate token (OAMAuthnCookie) will be valid for this agent. When the validity period is reached, the OAMAuthnCookie will be invalidated by WebGate. For resource WebGates, this means the user will be redirected to the credential collector, where either their overall session token will be validated, or they will need to reauthenticate if their overall session token is invalidated. The Token Validity Period controls the length of the user session for DCC WebGates, overriding the session lifetime configured on the OAM server. Default = 3600 (seconds)

15.5.2 WebGate Search Controls

You can create a new WebGate registration, or search for a specific WebGate or group of WebGates (all OAM WebGates, for instance).

[Figure 15-4](#) shows the WebGates Search controls, defaults, and the empty Search Results table.

Figure 15-4 WebGate Search Controls and Create Button



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 15-4](#).

Table 15-4 Agent Search Controls

Control	Description
Create WebGate	Click to open a fresh WebGate registration page.
Name	Enter the name (or partial name and wild card (*)) as defined on the registration page.
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 state to narrow the search and results: <ul style="list-style-type: none"> 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.

15.5.2.1 Searching for an OAM Agent Registration

Before you begin, the Agent must be a registered agent of Access Manager.

1. In the Oracle Access Management Console, click Application Security at the top of the window.

2. In the Application Security console, click **Agents**.
3. If not already displayed, select the desired agent type tab.
4. **Find:**
 - **All Enabled:** Select **Version All**, **State All**, and click the **Search** button.
 - **An Agent/WebGate Name:** In the text field, enter the exact name of the instance you want to find and click the **Search** button. For example:
`my_OAM_WebGate`
5. Click the **Search Results** tab to display the results table, then:
 - **Edit or View:** Click the **Edit** command button in the tool bar to see the configuration page.
 - **Delete:** Proceed to "[Deleting OAM Agent Registration Using the Console](#)".
 - **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 you finish.

15.5.3 Viewing or Editing an OAM Agent Registration Page in the Console

Users with valid Administrator credentials can change any setting for registered WebGates and programmatic Access Clients using the Oracle Access Management Console. The procedure is the same whether you are editing a WebGate or Access Client registration.

For example, you might want to revise the time-out 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 the WebGate configuration area. (Artifacts need only be copied to the WebGate directory path if the agent name, access client password, or security mode is changed.)

Note:

All changes made using the Oracle Access Management Console are taken up without restarting the application server, and are reflected automatically after the reconfiguration time-out period.

Before you begin, the agent must be registered and available in the Oracle Access Management Console.

See Also:

- [Creating OAM WebGate Page and Parameters](#)

1. From the Oracle Access Management Console, click **SSO Agents**.

- a. Double-click OAM Agents node to display the Search page.
- b. **Find the Registration:** See "[WebGate Search Controls](#)".
- 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 15-1](#), [Table 15-3](#)).
3. **User-Defined Parameters:** Add or modify these as desired ([Table 15-2](#)).
4. Click **Apply** to submit changes and dismiss the Confirmation window (or close the page without applying changes).
5. Copy the artifacts as follows (or install WebGate with the same specifications, then copy artifacts), including any Simple or Cert mode files. For example, Open mode files include:

Agent & Artifacts	Artifacts
OAM WebGate/Access Client	From the AdminServer (Console) host:
ObAccessClient.xml and cwallet.sso	$\$DOMAIN_HOME/output/\$Agent_Name/$ To the Agent host: $\$11gWG_install_dir/WebGate/config$.

6. Proceed as needed for your deployment:
[Managing Access Manager SSO, Policies, and Testing](#).

15.5.4 Deleting OAM Agent Registration Using the Console

Users with valid Administrator credentials can delete a registered WebGate or Access Client from the Oracle Access Management Console. Deleting an agent registration removes only the registration and not the associated host identifier, Application Domain, resources, or the agent itself.



See Also:

[OAM Agent Registration Parameters in the Console](#)

Before you begin, evaluate the Application Domain, resources, and policies associated with this agent and ensure that these are configured to use another agent (or be removed).

- In the Oracle Access Management Console, click **Application Security** at the top of the window.
 1. In the Application Security console, click **Agents** to display the Search page.
 2. **Find the Registration:** See "[WebGate Search Controls](#)".
 3. Select the desired registration from the results table, and open it to confirm it is the right agent to remove, close the page.
 4. Select the name in the results table, click the Delete (X) button, check the Confirmation dialog and then close the page.
 5. Confirm the Agent name is no longer listed in the navigation tree.

15.6 Remote Registration Tool, Modes, and Process

As an alternative to using the console for agent registration, you can use the remote registration utility, `oamreg`, with Oracle-provided templates.

Administrators using the Oracle Access Management Console or remote registration utility must have credentials stored in the System Store ([Managing Data Sources](#)).

This section provides details about remote registration in the following topics:

- [Remote Registration Command Arguments and Modes](#)
- [Common Elements within Remote Registration Request Templates](#)
- [Key Use, Generation, Provisioning, and Storage](#)



See Also:

"OAM Remote Registration"

15.6.1 Remote Registration Command Arguments and Modes

Before using the remote registration tool, two environment variables within the script must be set: `OAM_REG_HOME` and `JAVA_HOME`.

[Table 15-5](#) describes the samples, which presume the location of the tool to be `$OAM_REG_HOME` on a Linux system. Your environment might be different.

Table 15-5 Environment Variables to Set within `oamreg`

Environment Variable	Description
<code>OAM_REG_HOME</code>	The directory under which <code>RREG.tar</code> was exploded, followed by <code>/rreg</code> : <code>\$OAM_HOME/oam/server/rreg/client/rreg</code>
<code>JAVA_HOME</code>	The location where Java is located on the client computer. For example: <code>\$WLS_HOME/Middleware/jdk160_11</code> . Note: <code>\$JAVA_HOME</code> should point to JDK 1.6. (JDK 1.7 can also be used in R2PS3.)

Additionally, before using the remote registration tool, you must modify several tags in the request file, as described later ([Table 15-9](#)).

The arguments required to run the remote registration script are listed in [Table 15-6](#).

Table 15-6 Remote Registration Command Arguments: `mode`

Arguments	Description
<code>mode</code>	Either: <ul style="list-style-type: none"> • <code>inband</code> • <code>outofband</code>

Table 15-6 (Cont.) Remote Registration Command Arguments: mode

Arguments	Description
<code>input/filename.xml</code>	Either the absolute path to the input file (<code>*request.xml</code> or an <code>agentName_Response.xml</code>), or the path relative to the value of <code>\$OAM_REG_HOME</code> . The preferred location is <code>\$OAM_REG_HOME/input</code>

The sample commands illustrated in [Table 15-7](#) presume the location of the tool to be `$OAM_REG_HOME` on a Linux system.

Table 15-7 Remote Registration Command Samples

Command Type	Sample (on Linux)
In-band Administrator In-band Request	<code>./bin/oamreg.sh inband input/*Request.xml</code>
In-band Administrator Submitted Request	<code>./bin/oamreg.sh outofband input/starting_request.xml</code>
Out-of-band Administrator Returned Response	<code>./bin/oamreg.sh outofband input/agentName_Response.xml</code>
[prompt_flag] value: [-noprmt]	<p>Optional. When <code>-noprmt</code> is used, <code>oamreg</code> 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 <code>echo</code> command.</p> <p>Examples from <code>\$OAM_REG_HOME</code> location:</p> <pre>(echo username; echo password; echo WebGate_password;) ./bin/oamreg.sh inband input/Request.xml -noprmt config.file</pre> <pre>(echo username; echo password; echo WebGate_password; echo httpscert_trust_prompt;) ./bin/oamreg.sh inband input/Request.xml -noprmt</pre> <pre>(echo username; echo password; echo WebGate_password; echo cert_password;) ./bin/oamreg.sh inband input/Request.xml -noprmt</pre> <pre>(echo username; echo password; echo WebGate_password; echo httpscert_trust_prompt; echo cert_password;) ./bin/oamreg.sh inband input/Request.xml -noprmt</pre> <p>See Also: "Updating Agents Remotely"</p>



Note:

After launching the script, Administrators are prompted for a username and password (unless `-noprmt` is used as shown in [Table 15-7](#).)

After running the script, messages inform you of success or failure. Following a successful registration or update, you must copy the artifacts to the Agent host, as outlined in "[Updating Agent Configuration Files](#)".

15.6.2 Common Elements within Remote Registration Request Templates

Regardless of agent type, global elements are common within all remote registration request files.

[Table 15-8](#), shows the global elements.



Note:

In [Table 15-8](#), descriptions of each element are omitted; see [Table 15-1](#).

Table 15-8 Common Elements in Remote Registration Requests

Element	Example
<code><serverAddress></code>	<code><serverAddress>http://{oam_admin_server_host}:{oam_admin_server_port}</serverAddress></code>
<code><agentName></code>	<code><agentName>RREG_OAM</agentName></code>
<code><hostIdentifier></code>	<code><hostIdentifier>RREG_HostId11G</hostIdentifier></code>

Table 15-8 (Cont.) Common Elements in Remote Registration Requests

Element	Example
<code><agentBaseUrl></code>	<code><agentBaseUrl>http://{web_server_host}:{web_server_port}</agentBaseUrl></code>


 **Note:** Extended Template only

Table 15-8 (Cont.) Common Elements in Remote Registration Requests

Element	Example
<code><autoCreatePolicy></code>	<code><autoCreatePolicy>>true</autoCreatePolicy></code>


 **Note:** Extended Template only

Table 15-8 (Cont.) Common Elements in Remote Registration Requests

Element	Example
<applicationDomain>	<applicationDomain>RREG_OAM11G</applicationDomain>



 **Note:** Extended Template only

Table 15-8 (Cont.) Common Elements in Remote Registration Requests

Element	Example
<virtualhost>	<virtualhost>>false<virtualhost>

 **Note:**
Extended Template only

15.6.3 Key Use, Generation, Provisioning, and Storage

Each registered agent has a symmetric key, regardless of the registration method (Oracle Access Management Console versus remote registration).

Each application will have a symmetric key whether it is protected through 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. The following sections contain details.

- [Key Use](#)
- [Key Generation Process](#)
- [Key Accessibility and Provisioning](#)
- [Key Storage](#)

15.6.3.1 Key Use

Each OAM WebGate agent has its own secret key that is shared between the agent and the OAM Server.

If one WebGate is compromised, other WebGates are unaffected. The following presents an overview:

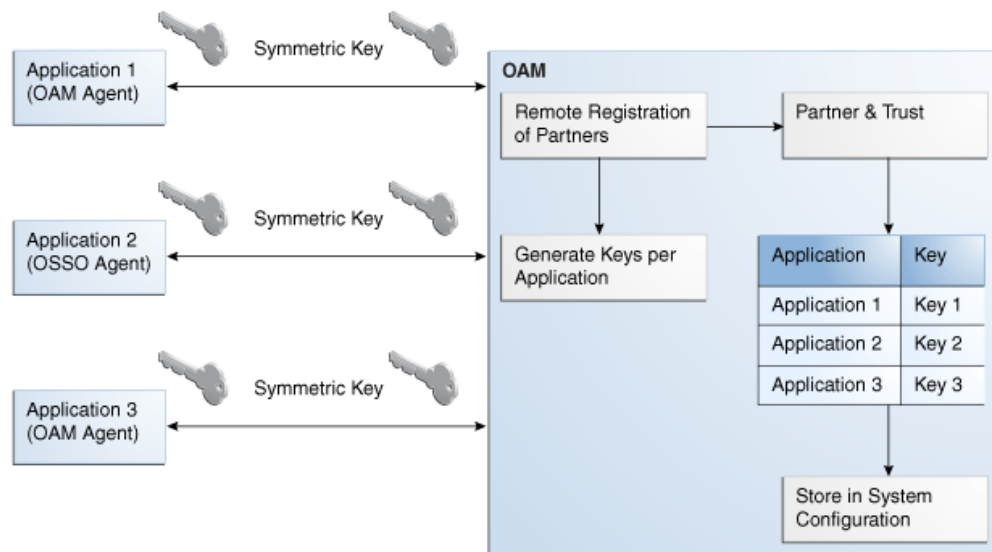
- Encrypt/Decrypt the host-based WebGate-specific OAMAuthnCookie_<host:port>_<random number>.
- Encrypt/Decrypt the data that is redirected between WebGate and OAM Server.

15.6.3.2 Key Generation Process

The key generation occurs automatically when the agent is registered, regardless of the method used (Oracle Access Management Console versus remote registration). There is one symmetric key per agent.

Figure 15-5 illustrates the process of key generation.

Figure 15-5 Key Generation



15.6.3.3 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 WebGates.

 **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 registration.

15.6.3.4 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_MW_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.

15.7 Remote Registration Templates: OAM Agents

Oracle provides both a short and extended registration request template for use with the remote agent registration tool: `oamreg.sh` (Linux) or `oamreg.bat` (Windows).

This topic focuses on OAM Agent templates (WebGates and Access Clients).

Table 15-9 Remote Registration Request Templates for OAM Agents

Template Type	Template Name in <code>\$OAM_REG_HOME/input/</code>
Abbreviated (Short) Form	<ul style="list-style-type: none"> • <code>OAM11GRequest_short.xml</code> (11g WebGates)
Extended (Full) Form	<ul style="list-style-type: none"> • <code>OAM11gRequest.xml</code> (11g WebGates)
Other Templates	For a look at these specialized tasks and templates, see:
Update Agent	<ul style="list-style-type: none"> • "Updating Agents Remotely "
Create Policies, Update Policies	<ul style="list-style-type: none"> • "Managing Policies and Application Domains Remotely"
Out-of-band Response	<ul style="list-style-type: none"> • "Performing Out-of-Band Remote Registration "

15.7.1 OAM Agent Parameters for Remote Registration

Element names in request templates might differ slightly from counterparts in the Oracle Access Management Console.

[Table 15-10](#) describes elements specific to OAM Agent remote registration requests. Protected, public, and excluded resource lists are included in both the short and extended request templates for OAM Agents.



Note:

Descriptions of elements in [Table 15-10](#) are in [Table 15-3](#).

Table 15-10 Elements in Extended OAM Agent Remote Registration Requests

Element	Example
<serverAddress>	See Table 15-8 .
<agentName>	
<hostIdentifier>	
<agentBaseUrl>	
<autoCreatePolicy>	
<applicationDomain>	
<virtualhost>	
<allowCredentialCollectorOperations>	
>	
<allowMasterTokenRetrieval>	
<hostPortVariationsList>	<pre><hostPortVariationsList> <host>host1</host> <port>7777</port> </hostPortVariations> <host>host2</host> <port>7778</port> </hostPortVariations> </hostPortVariationsList></pre>
<protectedResourcesList>	<pre><protectedResourcesList> <resource></resource> </protectedResourcesList></pre>
<publicResourcesList>	<pre><publicResourcesList> <resource>/public/index.html </resource> </publicResourcesList></pre>
<excludedresourcesList>	<pre><excludedresourcesList> <resource>/excluded/index.html </resource> </excludedresourcesList></pre>

Table 15-10 (Cont.) Elements in Extended OAM Agent Remote Registration Requests

Element	Example
<maxCacheElems>	<maxCacheElems>100000 </maxCacheElems>
<cacheTimeout>	<cacheTimeout>1800</cacheTimeout>
<tokenValidityPeriod> OAM WebGate Only	<tokenValidityPeriod>3600 </tokenValidityPeriod>
<maxConnections>	<maxConnections>1</maxConnections>
<maxSessionTime>	<maxSessionTime>24</maxSessionTime>
<failoverThreshold>	<failoverThreshold>1 </failoverThreshold>
<aaaTimeoutThreshold>-	<aaaTimeoutThreshold>-1 </aaaTimeoutThreshold>
<sleepFor>	<sleepFor>60</sleepFor>
<debug>	<debug>>false</debug>
<security>	<security>open</security>
<denyOnNotProtected>	<denyOnNotProtected>1 </denyOnNotProtected>
<allowManagementOperations>	<allowManagementOperations>>false/ <allowManagementOperations>
<cachePragmaHeader> <cacheControlHeader>	<cachePragmaHeader>no-cache </cachePragmaHeader> <cacheControlHeader>no-cache </cacheControlHeader>
<ipValidation>	<ipValidation>0</ipValidation>
<ipValidationExceptions>	<ipValidationExceptions> <ipAddress>10,11,11,11</ipAddress> <ipAddress>10,11,11,12</ipAddress> <ipAddress>10,11,11,13</ipAddress> </ipValidationExceptions>

Table 15-10 (Cont.) Elements in Extended OAM Agent Remote Registration Requests

Element	Example
<logOutUrls>	<pre><logOutUrls> <url>/logout1.html</url> <url>/logout2.html</url> </logOutUrls></pre>
<logoutCallbackUrl> OAM WebGate Only	<pre><logoutCallbackUrl>/oam_logout_success </logoutCallbackUrl></pre>
<logoutTargetUrlParamName> OAM WebGate Only	<pre><logoutTargetUrlParamName>end_url </logoutTargetUrlParamName></pre>
User-Defined Parameter Names	Examples
	<pre><userDefinedParameters> <userDefinedParam> <name>...</name> <value>...</value> </userDefinedParam></pre>
MaxPostDataLength	<pre><userDefinedParameters> <userDefinedParam> <name>MaxPostDataLength</name> <value>750000</value> </userDefinedParam></pre>
maxSessionTimeUnits	<pre><userDefinedParameters> <name>maxSessionTimeUnits</name> <value>hours</value> </userDefinedParam></pre>
useIISBuiltinAuthentication	<pre><userDefinedParameters> <name>useIISBuiltinAuthentication </name> <value>>false</value> </userDefinedParam></pre>
URLInUTF8Format	<pre><userDefinedParameters> <name>URLInUTF8Format</name> <value>>true</value> </userDefinedParam></pre>
inactiveReconfigPeriod Configuration applies to only OAM WebGate.	<pre><userDefinedParameters> <name>inactiveReconfigPeriod</name> <value>10</value> </userDefinedParam></pre>

Table 15-10 (Cont.) Elements in Extended OAM Agent Remote Registration Requests

Element	Example
WaitForFailover	<pre><userDefinedParameters> <name>WaitForFailover</name> <value>-1</value> </userDefinedParam></pre>
proxySSLHeaderVar	<pre><userDefinedParameters> <name>proxySSLHeaderVar</name> <value>IS_SSL</value> </userDefinedParam></pre>
client_request_retry_attempts	<pre><userDefinedParameters> <name>client_request_retry_attempts </name> <value>1</value> </userDefinedParam></pre>
ContentLengthFor401Response	<pre><userDefinedParameters> <name>ContentLengthFor401Response </name> <value>0</value> </userDefinedParam></pre>
SUN61HttpProtocolVersion	<pre><userDefinedParameters> <name>SUN61HttpProtocolVersion </name> <value>1.0</value> </userDefinedParam></pre>
impersonationCredentials	<pre><userDefinedParameters> <name>username:password </name> <value>cred</value> </userDefinedParam></pre>
UseWebGateExtForPassthrough	<pre><userDefinedParameters> <name>UseWebGateExtForPassthrough </name> <value>>false</value> </userDefinedParam></pre>
syncOperationMode	<pre><userDefinedParameters> <name>syncOperationMode</name> <value>>false</value> </userDefinedParam></pre>

Table 15-10 (Cont.) Elements in Extended OAM Agent Remote Registration Requests

Element	Example
filterOAMAuthnCookie <i>OAM WebGate only.</i>	<pre><userDefinedParameters> <name>filterOAMAuthnCookie</name> <value>true</value> </userDefinedParam></pre>

15.8 Performing Remote Registration for OAM Agents

This section includes the following topics describing how to perform remote registration, which is similar regardless of the agent type:

- [Acquiring and Setting Up the Remote Registration Tool](#)
- [Creating Your Remote Registration Request](#)
- [Performing In-Band Remote Registration](#)
- [Performing Out-of-Band Remote Registration](#)

15.8.1 Acquiring and Setting Up the Remote Registration Tool

The oamreg client tool can be used anywhere, not just on the OAM Server.

If the oamreg home is already exploded, you can use the following procedure to acquire and update the oamreg script for your operating system:

Windows: oamreg.bat

Linux: oamreg.sh

Note:

Oracle Recommends using the latest tool and files by applying the latest bundle patch and untarring RREG.tar.gz again as described here.

For remote registration, two variables are required: JAVA_HOME and OAM_REG_HOME, as described in [Table 15-11](#).

Table 15-11 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. (JDK 1.7 can also be used in R2PS3.)

Table 15-11 (Cont.) Variables Required for Remote Registration

Location	Variable	Description
	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 \$ORACLE_IDM_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:**["Remote Registration Command Arguments and Modes"](#)

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 15-11](#)).
JDK 1.7 can also be used in R2PS3.
 - b. Set OAM_REG_HOME to the *exploded_dir_for_RREG.tar/rreg* based on your environment (client side or server side [Table 15-11](#)).
4. Proceed with "[Creating Your Remote Registration Request](#)".

15.8.2 Creating Your Remote Registration Request

You can create an appropriate *Request*.xml file to provide input for the specific agent you want to register.

Before you begin, read [Remote Registration Templates: OAM Agents](#)

1. Locate the required *Request*.xml input file for the agent you want to register:
2. Copy the request file to a new name. For example:

From: OAM11GRequest.xml
To: my11gagent_request.xml

3. In the Request file, modify information to reflect details for your agent and the resources to protect using details in:
 - [Table 15-9](#)
 - [Table 15-10](#)
4. Proceed with task needed for your environment:
 - [Performing In-Band Remote Registration](#)
 - [Performing Out-of-Band Remote Registration](#)

15.8.3 Performing In-Band Remote Registration

The OAM Administrator within the network performs all tasks. Regardless of agent type, you can perform in-band remote registration.

For this example, an OAM Agent is being registered using the short request on a Linux system. Your agent type, request template, and output files will be different.



See Also:

[Configuring Oracle HTTP Server WebGate for Oracle Access Manager in *Installing WebGates for Oracle Access Manager*](#)

Before you begin, read:

- [Acquiring and Setting Up the Remote Registration Tool](#)
 - [Creating Your Remote Registration Request](#)
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.

The following example illustrates a sample rreg registration output.

```
Welcome to OAM Remote Registration Tool!
Parameters passed to the registration tool are:
Mode: inband
Filename: /scratch/work/mw1916/idm1385/oam/server/rreg/input/1.xml
Enter admin username:oamadminuser
Username: oamadminuser
Enter admin password:
Do you want to enter a Webgate password?(y/n):
n
Do you want to import an URIs file?(y/n):
n

-----
Request summary:
OAM Agent Name:RREG_1234
URL String:RREG_1234
Registering in Mode:inband
Your registration request is being sent to the Admin server at: http://
slc01huw.us.example.com:20081
-----
```

Inband registration process completed successfully! Output artifacts are created in the output folder.

The output folder is in the same location where RREG.tar.gz was expanded: `/rreg/output/AgentName/`

3. Review the native configuration file created for the agent in the `/rreg/output/AgentName/` folder.
4. **Finalize Registration:** Perform the following steps to replace the earlier agent configuration file if it is not already replaced:
 - a. Copy artifacts in `/rreg/output/AgentName/` to update the agent configuration. For example:

From the AdminServer (Console) host

```
/rreg/output/Agent_Name/ObAccessClient.xml and cwallet.sso
```

To the Agent host: `$11gWG_install_dir/WebGate/config`. For example:

```
$WebTier_MW_Home/Oracle_WT1/instances/instance1/config/OHS/ohs1/WebGate/config
```
 - b. Restart the OAM Server hosting the agent.
5. Proceed with "[Validating Remote Registration and Resource Protection](#)".

15.8.4 Performing Out-of-Band Remote Registration

This section provides steps for Administrators outside (and inside) the network as they work together to register an agent remotely. During out-of-band remote registration, an administrator outside the network submits a registration request to an Administrator within the network. After processing the request, the in-band Administrator returns the following files to the out-of-band Administrator to configure his environment.

Table 15-12 Files Returned by in-band Administrator to out-of-band Administrator

File	Description
<code>agentName_Response.xml</code>	Returned to, and used by, the out-of-band Administrator. Oracle recommends that you do not open or edit <code>agentName_Response.xml</code> .
Native Web server configuration files	Returned to, and used by, the out-of-band Administrator to update his Web server.
See Also	"Updating Agent Configuration Files"

The steps performed by each Administrator are identified:

- **In-Band Administrator:** Identifies a task performed by the Web server Administrator within the network.
- **Out-of-Band Administrator** Identifies a task performed by the Web server Administrator outside the network

 **See Also:**

Configuring Oracle HTTP Server WebGate for Oracle Access Manager in *Installing WebGates for Oracle Access Manager*

Steps here illustrate registering an OAM Agent on a Linux system. Your templates and output files will be different.

Before you begin, read [Acquiring and Setting Up the Remote Registration Tool](#)

1. **Out-of-Band Administrator:** Create and send your *starting_request.xml* file to the in-band Administrator for processing (see "[Creating Your Remote 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/
```

- 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
```

- b. Copy artifacts generated in */rreg/output/AgentName/* to update the agent configuration (), then restart the OAM Server hosting the agent. For example, *ObAccessClient.xml* and *cwallet.sso*:

From the AdminServer (Console) host */rreg/output/Agent_Name/*
ObAccessClient.xml and *cwallet.sso*

To the Agent host: *\$11gWG_install_dir/WebGate/config*. For example:

```
$WebTier_MW_Home/Oracle_WT1/instances/instance1  
/config/OHS/ohs1/WebGate/config
```

- c. Proceed with "[Validating Remote Registration and Resource Protection](#)".

15.9 Remote Agent Update Modes and Templates

Administrators quickly update, validate, or delete an existing agent registration using remote management modes.

This section provides the following topics:

- [Remote Agent Update Modes](#)
- [Remote OAM Agent Updates Template](#)

15.9.1 Remote Agent Update Modes

To manage an existing agent registration, remote agent management modes can be used.

Table 15-13 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 15-13 Remote Agent Update Modes and Input Files

Mode and Input Files	Description and Syntax
agentUpdate mode <i>OAM11GUpdateAgentRequest.xml</i> <i>OAMUpdateAgentRequest.xml</i>	Allows Administrators to update existing agent attributes, regardless of agent type: ./bin/oamreg.sh agentUpdate input/ *UpdateAgentRequest.xml
agentValidate mode <i>No input file needed.</i>	Validates whether the agent is already provisioned in Oracle Access Manager: ./bin/oamreg.sh agentValidate agentname
agentDelete mode <i>No input file needed.</i>	Allows Administrators to delete the agent registration: ./bin/oamreg.sh agentDelete agentname

15.9.2 Remote OAM Agent Updates Template

You use *OAM11GUpdateAgentRequest.xml* to pass specific Agent-update values to the remote registration tool, oamreg.

The primary differences between the update request and the original registration request is that the update request.

Table 15-14 Delta: OAM Agent Update versus Registration Request

Delta	Element
Adds	<ipValidation>
Omits	<ipValidationExceptions>

Table 15-14 (Cont.) Delta: OAM Agent Update versus Registration Request

Delta	Element
Omits	<hostidentifier>
Omits	<virtualhost>
Omits	<hostportVariations>
Omits	<authCreatePolicy> and application domain-related elements
Omits	<ssoServerVersion>
Omits	<idleSessionTimeout>

**See Also:**

- [Table 15-3](#)

15.10 Updating Agents Remotely

This section provides the following topics for agents registered with Access Manager, regardless of agent type:

- [Updating Agent Registrations Remotely](#)
- [Validating an Agent Registration Remotely](#)
- [Removing an Agent Registration Remotely](#)

15.10.1 Updating Agent Registrations Remotely

Regardless of agent type, you can remotely update agents registered with Access Manager.

This topic provides the steps to update agents registered with Access Manager, regardless of agent type. Before you begin, review [Remote Agent Update Modes](#)

1. Set up the registration tool as described in, "[Acquiring and Setting Up the Remote Registration Tool](#)".
2. Create your update request using one of the following templates:
 - OAM11GUpdateAgentRequest.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. Finalize Agent Registration: Copy the updated ObAccessClient.xml and cwallet.sso.

From the AdminServer (Console) host: /rreg/output/Agent_Name/

To the Agent host: \$11gWG_install_dir/WebGate/config. For example:

```
$WebTier_MW_Home/Oracle_WT1/instances/instance1/ config/OHS/ohs1/WebGate/
config
```

7. Restart the OAM Server that is hosting this agent and proceed to "[Validating an Agent Registration Remotely](#)".

15.10.2 Validating an Agent Registration Remotely

Regardless of agent type, you can remotely validate agent registration.

This topic provides the steps to validate agent registration, regardless of agent type. Before you begin, review [Remote Agent Update Modes](#)

1. Set up the registration tool as described in, "[Acquiring and Setting Up the Remote Registration Tool](#)".
2. On the Agent host, 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!
```

15.10.3 Removing an Agent Registration Remotely

You can remove a registered agent, regardless of agent type.

Before you begin, review [Remote Agent Update Modes](#)

1. Set up the registration tool as described in, "[Acquiring and Setting Up the Remote Registration Tool](#)".
2. On the computer hosting the Agent, run the following agentDelete command. 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!
```

15.11 Validating Remote Registration and Resource Protection

You can validate registration of an agent regardless of the agent type.

You must be an in-band Administrator to perform tasks using the Oracle Access Management Console. Out-of-band Administrators must test authentication and access remotely.

- [Validating Agent Registration using the Oracle Access Management Console](#)
- [Verifying Authentication and Access After Remote Registration](#)

15.11.1 Validating Agent Registration using the Oracle Access Management Console

Only an in-band Administrator can validate agent registration.



See Also:

[Managing Policies to Protect Resources and Enable SSO](#)

1. **Validate Agent Registration** in the Oracle Access Management Console:
 - a. Confirm Agent details under Application Security in the Oracle Access Management Console.
 - b. Confirm the updated Agent configuration files are in the appropriate location, as described in "[Performing Remote Registration for OAM Agents](#)".
2. **Validate Shared Components**, Host identifier: Confirm that the host identifier is defined in the Oracle Access Management Console.
3. **Validate Application Domain**: Under the Policy Configuration tab, confirm there is a new Application Domain named after the registered agent. Resources in the Application Domain should be associated with the host identifier.
4. Proceed with "[Verifying Authentication and Access After Remote Registration](#)".

15.11.2 Verifying Authentication and Access After Remote Registration

After registration, protected resource should be accessible with proper authentication without restarting the AdminServer or OAM Server.

Both in-band and out-of-band Administrators can use the following procedure to validate proper registration and policies.

The procedure here provides several methods for confirming that registration, authentication, and authorization are properly configured and operational. The procedures is nearly identical for all agent types.

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://exampleWebserverHost.sample.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%2FhbItbjT24CBmJoleCIfDIwQ1atdGdnY4mt6kmdSekSFeAAFvFrZZZ
xDfvpkfS3ZLZFbaZU2rAn0YYUM3JUWVYkYFwB%2BBK7V4x%2FeuYHj%2B8gwOyxhNYFna3iSx1MSZBE
y51KTBFsDYoiw6R%2BCxUh008uZDTYHI3s0c7AQsYrEiQTuUV3nv1omaFZ1k1GuZa4J7ycaGbIUyqWx
rM0cKuBJNd6sX1LiRj9HofYQsvUV7ToqeAOpDS7z9qs5LhqU5Vq60bBn12DTX6zNX6Lcc0L5tVvvh7%
2Bn0Akz2%2BoDkLs%2BBTkeGcB3ppgC9;httponly; path=/; domain=.example.com;
```

OAM_ID Cookie:

Set-Cookie:

```
OAM_ID=v1.0~0~E1EBBC9846E09857060A68E79AE6B608~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 Management 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:

[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 State. Start the Web server and OAM Server. Access an application protected by the Agent and confirm that you are asked to authenticate.

15.12 setAllowEmptyHostIdentifier

Enables and disables the use of an empty preferred host parameter.

Description

Enables or disables the use of an empty preferred host parameter. The following parameters (added to the oam-config.xml file) will be set to enable or disable an empty preferred host parameter in the ObAccessClient.xml file.

```
<Setting Name="AutoUpdateHostIdentifier"
  Type="xsd:string">AUTO_UPDATE_HOSTID</Setting>
<Setting Name="AllowEmptyHostIdentifier"
  Type="xsd:boolean">>true</Setting>
```

Syntax

```
setAllowEmptyHostIdentifier (enable="true/false")
```

Argument	Definition
<i>enable</i>	Set to true or false to allow for an empty host identifier or not.

Sample

```
setAllowEmptyHostIdentifier (enable = "true")
```

Maintaining Access Manager Sessions

An Access Manager session is created during authentication and bound to both the user and the client with which the user has authenticated. Access Manager sessions are maintained to provide tracking and policy enforcement (performed either manually by an Administrator or using automated flows) for a given session's lifecycle.

The Access Manager session lifecycle consists of state transitions for session creation, updates, idleness, and expiration.

The following topics describe concepts and procedures for Access Manager sessions:

- [Introducing Access Manager Session Management](#)
- [Understanding Server-Side Session Management](#)
- [Server-Side Session Enforcement Examples](#)
- [Configuring the Server-Side Session Lifecycle](#)
- [Managing Active Server-Side Sessions](#)
- [Validating Server-Side Session Operations](#)
- [Using REST APIs for CRUD Operations on a Session](#)

16.1 Introducing Access Manager Session Management

With this 12.2.1.3.0 release of Oracle Access Management, Access Manager sessions are managed on the server-side backed by a database.

OAM 12.2.1.3.0 server supports database-backed server-side session manager to synchronize session state across multiple nodes of the OAM server cluster. The session is persisted in the AM_SESSION table.

Session manager handles data for an active session uniquely identified by the OAM Server. The lifecycle of the session is managed at the server side by storing a session-id at the client side. The session-id uniquely identifies a user session stored at the OAM 12c server and is used in subsequent requests to fetch the session from server session store and identify the state of client.

The OAM server transforms the session into a token (session token) and sends it in the response. The token contains information to query the database on subsequent access to reconstruct the session at a later time. On subsequent requests, the token is retrieved from the request to identify and fetch the session to get all the details of logged-in user.

As the server-side sessions are backed by a database, it is required that the database is always available so that the sessions being fetched from the database do not fail during Access operations. OAM 12.2.1.3.0 provides Tokenized Session Management that supports access operations when the database is not available.

Database Unavailability Scenarios

In OAM 12.2.1.3.0, only the following scenarios are taken as database-unavailability by the Session Manager.

- Database is down
- Connection pool exhaustion, due to which there are no free connections to connect to the database
- Network latency issues, due to which the database connectivity, and latency of the database operations are degraded

 **Note:**

Database must be available during startup of admin server, policy manager and OAM runtime servers of OAM domain.

The Tokenized Session supports the following features when the database is unavailable:

Table 16-1 Features Supported when the Database is Unavailable

Features and Components Supported when the DB is Unavailable	Scenarios
WebGate 11g	Authentication, Authorization (without session constraints) & SSO flows
Webgate 12c	Authentication, Authorization (without session constraints) & SSO flows
Java ASDK	ASDK Operations
OAuth	Non-session link scenarios

 **Note:**

ASDK operations related to session are not supported.

The Tokenized Session Management does not support the following features when the database is unavailable.

Table 16-2 Features Not Supported when the Database is Unavailable

Features and Components Not Supported (as they rely on server-side sessions) when the DB is Unavailable	Description
Java ASDK	DB availability is mandatory for ASDK session operations to work

Table 16-2 (Cont.) Features Not Supported when the Database is Unavailable

Features and Components Not Supported (as they rely on server-side sessions) when the DB is Unavailable	Description
Session Responses	DB availability is mandatory for session responses to work
Session Snipe	DB availability mandatory. If DB is not available after session snipe, behavior of authorizations from the already established sessions are defined based on the session policy
Remote Session Kill	DB availability mandatory
OAuth MDC Session Link	DB availability mandatory due to OAuth Token and OAM_ID sync issues in the flow.
DCC/NAP Tunneling	Not supported / Not certified
LDAP Prefetch	LDAP Prefetch will not work when DB is down. Additional LDAP queries might be needed during authorization to evaluate identity authz constraints.
oamconsole	Admin operations are not supported when DB is down

**See Also:**[Polling Interval for System and Policy Configuration](#)[SQL Queries to List Sessions and Plugins from Database](#)

16.2 Understanding Server-Side Session Management

OAM 12.2.1.3.0 server supports database-backed server-side session manager to synchronize session state across multiple nodes of the OAM server cluster

The following topics provide an overview of server-side session management:

- [About Securing Access Manager Sessions](#)
- [Access Manager Session Lifecycle, States, and Enforcement](#)

16.2.1 About Securing Access Manager Sessions

Session security begins with a secure installation.

For installation details see the Preparing to Install and Configure Oracle Access Management in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

 **See Also:**

Configuring SSL in Oracle Fusion Middleware in the *Fusion Middleware Administrator's Guide* for details about configuring secure communications between Oracle Fusion Middleware components using SSL.

The HTTPS protocol and database encryption are some of the ways in which Access Manager supports server-side session security. The following list describes how this support can work.


- **HTTPS Protocol**
Access Manager helps prevent session fixation by providing IP address checks by the Proxy. To further help prevent session fixation, be sure to use the secure HTTPS protocol for communication between WebGates and OAM Servers.
- **Database Encryption**
The Session Management Engine does not encrypt data. For security concerns, use an in-database encryption such as Oracle Advanced Security.

16.2.2 Access Manager Session Lifecycle, States, and Enforcement

The session lifecycle refers to a set of states with defined transitions from one state to another that depend on user activity (or lack thereof), and manual (or automated) Administrator activity.

Administrators can define the following global session lifecycle settings:

- **Session Lifetime**

 **Note:**

If a DCC WebGate is used for authentication and its Token Validity Period is shorter than the Session Lifetime value, users will need to re-authenticate when the Token Validity Period expires. The Token Validity Period should be the same as the Session Lifetime when using DCC authentication.

- **Idle Timeout**

 **Note:**

Idle Timeout can also be implemented as application-specific settings, as described later.

- **Maximum number of Sessions**

See [Global Session Lifecycle Settings](#).

[Table 16-3](#) lists the Session lifecycle states.

Table 16-3 Session Lifecycle States

State	Description
Active	Newly-created sessions are active. A session is created when the user is authenticated by Access Manager. The session remains active until Access Manager determines that the session must transition into one of the other states in this table. Note: Administrators can delete only active sessions.
Idle	An active session becomes idle when the user does not access Access Manager-protected content for the period defined by an Administrator. When an active session becomes idle, the user must re-authenticate to proceed. When re-authentication is successful, the session returns to the Active state; session attribute values are preserved through this process.
Expired	An active session expires when the duration of the session exceeds the defined lifetime. An expired session is completely inaccessible and eligible for deletion. When an active session expires, the user must re-authenticate to proceed. When re-authentication is successful, a new session is created; however, session attribute values are not preserved (as they are for Idle states).

See the following topics for more information about:

- [Global Session Enforcement Checks for State Changes](#)
- [Access Manager Session Removal](#)
- [About Step-Up and Step-Down Authentication and Credentials](#)
- [Optional Application-Specific Session Enforcement](#)

16.2.2.1 Global Session Enforcement Checks for State Changes

Each Access Manager session holds two time attributes and applicable values.

Following are the two time attributes:

- Session creation time
- Last access time

The values of these attributes are compared for session enforcement as described in "Table 16-4".

Table 16-4 Session Checks for State Changes

Session Check	Description
Is the Session Idle?	Compares the last access time against the configured Idle Timeout value. Exceeding the configured period triggers a change from the Active to the Idle state.
Is the Session Expired?	Compares the session creation time against the configured Session Lifetime. Exceeding the configured period triggers a change from the Active to the Expired state.

During transitions to the Idle state, underlying session attributes are preserved because the user previously satisfied authentication criteria and the data is trusted. However, continued access to protected resources based on that session, and resulting modification of data

within that session, is not allowed until the user re-authenticates, proving not to be a malicious user with access to an unlocked computer.

16.2.2.2 Access Manager Session Removal

A session can be removed by three actions: expiration, user logout, and termination.

Session removal actions are described in [Table 16-5](#).

Table 16-5 Session Removal

Action	Description
Expiration	Expired sessions are eligible for removal based on their creation time. Actual removal is determined by the storage mechanism. The session is removed from the distributed cache using a background task running on the server; it is removed from the database using a similar background task, an optionally-enabled job within the database itself, or both methods in combination. Once a session has been deleted from storage on all tiers (local and distributed caches, and from the database if enabled), the session is removed.
User Logout	User Logout triggers immediate removal from the distributed cache, subject to present volume of DB session writes and performance.
Termination	Termination is identical to user log out whether the session is interactively terminated through the Administration Console or in an automated way--as part of an Oracle Identity Management user lockout or de-provisioning flow.

16.2.2.3 About Step-Up and Step-Down Authentication and Credentials

On occasion, multiple forms of authentication are required and performed within a single session to complete a step-up flow. A re-authentication level might be a step down from the session.

In a step-up flow, a user authenticates to access protected content and later in the same session, the user requests other, more sensitive content and is required to authenticate again to access it at a more stringent level. In a step-up flow, multiple authentications always occur in order of the increasing authentication level. Each session holds the Authentication Level attribute for step-up authentication enforcement.

If the re-Authentication Level is less than that previously contained in the session, the user has completed a step-down process. Upon successful re-authentication, the session is restored to the Active state with an Authentication Level that is equal to the lower level of the authentication scheme used. If the user later attempts to access content that is protected at a higher level, step-up authentication occurs.

16.2.2.4 Optional Application-Specific Session Enforcement

Access Manager enforces limitations on user access to resources in a more granular way than is possible with a single set of global session timings, or single set of authentication schemes in which access depends solely on a single authentication

level. Access to certain data has more stringent requirements, while access to all other data is configured globally.

Administrators can choose to override global session timeout settings on a per application basis, defined as part of Application Domain settings. Optional application-specific session configuration provides:

- The ability to declare session idle timings on a per-application basis, which is generally more stringent than the global idle timing defined within the deployment as a whole.
- The ability to require the user to re-authenticate after a per-application session inactivity timeout.

[Table 16-6](#) describes session enforcement when you have defined Application Domain-specific overrides to global session settings.

Table 16-6 Application Domain-Specific Overrides

Override	Description
Is the Session Idle?	Compares the last access time against the configured Idle Timeout value for the defined Application Domain only. Exceeding the configured period triggers a change from the Active to the Idle state.
Is the Session Expired?	Compares the session creation time against the configured Session Lifetime. Exceeding the configured period triggers a change from the Active to the Expired state for the defined Application Domain group only.

16.3 Server-Side Session Enforcement Examples

Satisfying the authentication scheme of a given level provides access to all resources protected at lower levels. Additionally, all authentication schemes of a given level are viewed as equivalent.

This section provides a simple session enforcement example based on a single authentication scheme used in two application domains as well as a more complex example based on multiple authentication schemes used in two application domains.

- [Example 1: Single Authentication Scheme](#)
- [Example 2: Multiple Authentication Schemes](#)

16.3.1 Example 1: Single Authentication Scheme

Consider the following configuration:

- A single authentication scheme (S1) defined using Level 2
- Application domains D1 and D2
- All resources within each domain are protected with a single authentication policy, which uses S1, and a single authorization policy.
- Global Session Configuration:
 - Session Lifetime: 90 minutes
 - Idle Session Timeout: 0 (session never idles out)
 - Application Domain Timeout: 30 minutes

Now consider the outcomes in [Table 16-7](#).

Table 16-7 Session Content: Single Authentication Scheme

Time (Delta)	Action	Access Allowed or Denied	Session Content
0	Access to D1	Denied due to no session	null
1	Authentication with S1 and Access to D1	Allowed because Authentication scheme is satisfied	Level 2, authentication time 1
21	Access to D2	Allowed	Level 2, authentication time 1
66	Access to D1	Denied due to Application Domain Timeout (based on the parameters configured)	Level 2, authentication time 1
67	Authentication with S1 and Access to D1 and D2	Both Allowed because the Authentication Scheme is satisfied	Level 2, authentication time 67

16.3.2 Example 2: Multiple Authentication Schemes

Step-down authentication occurs when a session times out as a matter of course — until the user happens to provide new credentials that satisfy a scheme of the same level as the maximum held by the session previously. Otherwise, from the authentication perspective, it is as if the session is new and further step-up is required. Consider this example with two authentication schemes (for step-up and step-down).

- Authentication schemes S1 (Level 2) and S2 (Level 3)
- Application domains D1 and D2
- All resources within each domain are protected with a single authentication policy, and a single authorization policy
- D1 uses S1; D2 uses S2
- Global Session Configuration:
 - Session Life: 240 mins
 - Idle Timeout: 30 mins
 - Appdomain 2 (D2) Timeout: 15 mins (appdomain setting)

When accessing resources from D1, timeout will occur after 30 minutes (global timeout setting); D2 timeout will happen after 15 mins since its timeout value is overridden at the global level. [Table 16-8](#) shows the resulting outcomes.

Table 16-8 Session Outcomes: Multiple Authentication Schemes

Time (Delta)	Action	Access Allowed or Denied	Session Content
0	Access D1 resource (RD1)	Access allowed after successful login	Timeout for D1 will be set to 0+30=30 (30 is default global timeout as D1 has not overridden timeout at the Application Domain level)

Table 16-8 (Cont.) Session Outcomes: Multiple Authentication Schemes

Time (Delta)	Action	Access Allowed or Denied	Session Content
1 (implies after 1 minute)	Access D2 resource (RD2)	Access allowed post credential challenge (user will be prompted for credentials since D2 is protected using a higher authentication scheme)	Timeout of D2 will be set to 1+15=16
t>16 and t<30 (say t=20)	Access RD1 and RD2	Allowed access to RD1 because timeout of D1=30. Allowed access to RD2 after providing credentials since timeout of D2=16	The new timeout of D2 is 16
40	Access RD1	Allowed: D1 resource will be allowed since timeout is 50	
55	Access RD1 and RD2	Allowed to access both resources after user is successfully challenged for credentials.	Timeout of D1 is now 85 (55+30) Timeout of D2 is now 70 (55+15)

The access order does have an impact on the outcome. For instance, the last D1 access could have been allowed if the user had chosen to first pursue access to the D2 application after credentials had expired. For example:

- Authentication S2 with Access to D2 Allowed: L3 scheme satisfied; resulting level of the now (again) active session same as before. Session Content: Level 3, authentication time 51
- Access to D1 Allowed: Level 3 credentials also sufficient for Level 2-protected access. Session Content: Level 3, authentication time 51.

16.4 Configuring the Server-Side Session Lifecycle

Session Lifecycle settings can be defined using the Oracle Access Management Console. When you define either global or application-specific session lifecycle settings, any timing interval set to 0 cancels the corresponding check.

For example if idle timeout is set to 0, sessions never idle out. With a session lifetime of 0, sessions never expire. In all cases, applicable data is tracked and updated in the session, just as if it is being checked on a per-request basis.

This section provides the following topics:

- [Global Session Lifecycle Settings](#)
- [Polling Interval for System and Policy Configuration](#)
- [Application-Specific Session Overrides](#)
- [Viewing or Modifying Global Session Settings](#)
- [Viewing or Modifying Optional Application-Specific Session Overrides](#)

16.4.1 Global Session Lifecycle Settings

Access Manager session lifecycle settings are defined as part of the Common Settings shared by all OAM Servers.

Figure 16-1 shows the lifecycle attributes that you can configure on the Common Settings page.

Figure 16-1 Global Session Details: Common Settings Page

The screenshot shows a configuration page for session settings. It features a section titled "Session" with a dropdown arrow. Below this, there are three configuration items, each with a text input field and up/down arrow buttons:

- * Session Lifetime (minutes): 480
- * Idle Timeout (minutes): 15
- * (Management) Maximum Search Results: 100

On the right side of the configuration area, there is another item: * Maximum Number of Sessions per User: 5.



Table 16-9 describes the global session lifecycle settings and their defaults. Sessions can operate in a disconnected mode. Therefore, changes to the configuration establishing your session rules apply only to new sessions. To apply changes immediately, Oracle recommends that you terminate existing sessions and force users to create new ones that adhere to your new rules.



See Also:

Tuning Performance

Table 16-9 Global Session Settings

Setting	Description
Session Lifetime (minutes)	<p>The amount of time, in minutes, that a user's authentication session remains active. When the lifetime is reached, the session expires.</p> <p>Default = 1440 minutes (24 hours specified in an integer representing minutes)</p> <p>A value of zero (0) disables this setting. Any value between 0 (zero) and 2147483647 is allowed.</p>
	<div style="border: 1px solid #0070C0; padding: 10px;"><p> Note:</p><ul style="list-style-type: none">• The change is effective after the subsequent poll for configuration change (based on the interval configured in <code>oracle.oam.EntityRefreshIntervalMillis</code>. Default value is 30 seconds.)• An expired session is automatically deleted from the in-memory caches (or database).</div>
Idle Timeout (minutes)	<p>The amount of time, in minutes, that a user's authentication session remains active without accessing any Access Manager protected resources. When the user is idle for a longer period, they are asked to re-authenticate.</p> <p>Default = 15 minutes</p> <p>A value of zero (0) disables this setting. Any value between 0 (zero) and 2147483647 is allowed.</p>
	<div style="border: 1px solid #0070C0; padding: 10px;"><p> Note:</p><ul style="list-style-type: none">• The change is effective after the subsequent poll for configuration change (based on the interval configured in <code>oracle.oam.EntityRefreshIntervalMillis</code>. Default value is 30 seconds.)• Timed-out sessions are not deleted from the session manager. Session data could be removed from memory but will still be available in the persistent store (database). After re-authentication, the same session will be re-activated.</div>

See Also: "[Application-Specific Session Overrides](#)"

Table 16-9 (Cont.) Global Session Settings

Setting	Description
Maximum Number of Sessions per User	<p>The exact number of sessions each user can have at one time. Use this setting to configure multiple session restrictions for all users. Any positive integer is allowed.</p> <p>Specifying the count as "1", activates a special mode. If a user who already has a session authenticates using another device (thereby creating a new session), then their existing session is deleted. No error is reported and no warning is given.</p>
(Management) Maximum Search Results	<p>Maximum number of sessions fetched by default for a session query if the result set is large.</p>

 **Note:**

- During authentication, if the database is not available, the Maximum Number of Sessions per User enforcement does not work.
- Too high a number impacts performance and result in a security risk. Oracle recommends less than 20 as a reasonable limit per user. Otherwise there can be performance impact. For tuning information, see *Tuning Performance*

16.4.2 Polling Interval for System and Policy Configuration

OAM system configuration store and policy stores are polled for changes with the following polling intervals.

In OAM 12.2.1.3.0, policy and system configurations are synchronized over a pull-based model. This is driven by the individual OAM server nodes polling for changes in corresponding configurations.

Table 16-10 Default Polling Interval

Configurations	Default Polling Interval (in milliseconds and seconds)
System Configuration Store	every 30000 milliseconds (30 seconds)
Policy Store	every 30000 milliseconds (30 seconds)
OAM Config MBean reloads changes from configuration store	every 30000 milliseconds (30 seconds)

Polling interval can be configured using the `oracle.oam.EntityRefreshIntervalMillis` system property for all the configurations listed in [Table 16-10](#).

For example, to change the policy store polling interval to 60 seconds, set the following:

```
-D oracle.oam.EntityRefreshIntervalMillis=60000
```

16.4.3 Application-Specific Session Overrides

Application-specific access is tracked from the initial application-access time and is updated only as further requests are made of that Application Domain.

In other words, the user's authentication and the authentication state are under control of Access Manager and the Administrator. The current idle time for a given application is shared between Access Manager and the application. The application provisions its own run time data for the user on a per-session basis and needs to remove it as soon as possible to make room for others.

Administrators can add application-specific session overrides on the Summary tab of an Application Domain. [Table 16-11](#) lists application-specific settings that, when specified, override global session settings.

Table 16-11 Application-Specific Session Timing Overrides

Element	Description
Idle timeout	Access Manager previously stored the last access time value within the session. To enforce maximum idle time on a per-application basis, Access Manager includes a new application-specific last access time field to hold it. This is filled with the last access time for each subset of domains visited within the course of a session, on which a per-application idle timeout override has been defined. This is not needed for domains on which an override has not been defined--no checking is done against such data. Default: undefined

 **Note:**

- The change is effective after the subsequent poll for configuration change (based on the interval configured in `oracle.oam.EntityRefreshIntervalMillis`. Default value is 30 seconds.)
- When the app domain timeout value is less than the global session timeout value and it is more restrictive, the app domain idle timeout overrides the global session idle timeout. When it is greater than global session timeout, OAM session timeout is based on the Global Settings value.
- When the DB is down, the app domain session idle timeout does not override the global session idle timeout.

For more information, see "[Viewing or Modifying Optional Application-Specific Session Overrides](#)".

16.4.4 Viewing or Modifying Global Session Settings

Users with valid Administrator credentials can to modify common session lifecycle settings using the Oracle Access Management Console.

For more information, see "[Global Session Lifecycle Settings](#)"

1. In the Oracle Access Management Console, click Configuration at the top of the window.
2. In the Configuration console, select **Common Settings** from the **View** menu in the **Settings** section.
3. On the Common Settings page, expand the **Session** section.
4. Click the arrow keys beside each list to increase or decrease session lifecycle settings as needed ([Table 16-9](#)):
 - Session Lifetime (minutes)
 - Idle Timeout (minutes)
 - Maximum Number of Sessions per User
 - (Management) Maximum Search Results - denotes the number of sessions fetched by default for a session query if the result set is large
5. Click **Apply** to submit the changes (or close the page without applying changes).
6. Close the page when you finish.
7. Proceed to one of the following topics:
 - "[Viewing or Modifying Optional Application-Specific Session Overrides](#)"
 - "[Managing Active Server-Side Sessions](#)"

16.4.5 Viewing or Modifying Optional Application-Specific Session Overrides

Users with valid Administrator credentials can modify optional session settings for one or more application domains in a named group.

For more information, see "[Application-Specific Session Overrides](#)".

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Access Manager section, click **Application Domains**.
3. Find and open the desired domain.
4. On the Summary tab, enter the following information to create (or add) this domain to the group that uses session overrides ([Table 16-11](#)):
 - Idle Timeout
5. Click **Apply** to submit the changes (or close the page without applying changes).
6. Proceed to "[Managing Active Server-Side Sessions](#)".

16.5 Managing Active Server-Side Sessions

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. The Oracle Access Management Console 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:

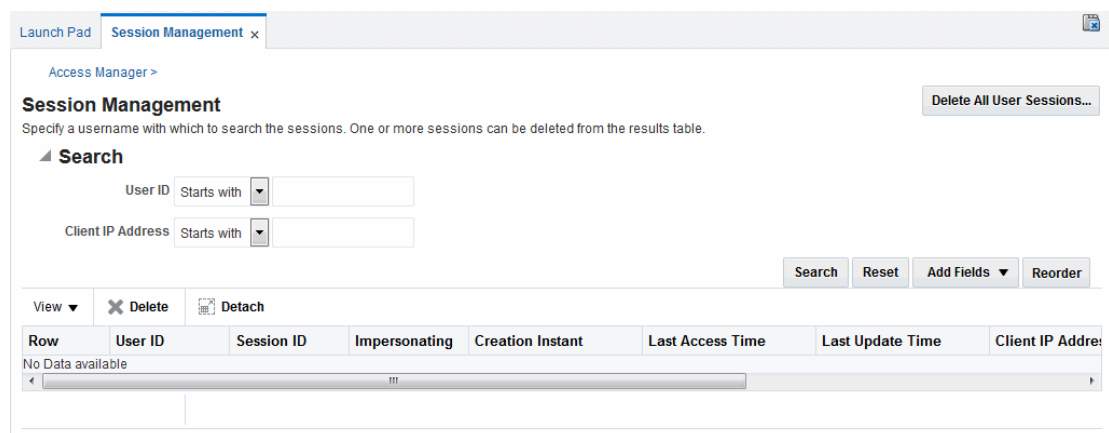
- [Session Management Controls](#)
- [Locating and Managing Active Sessions](#)

16.5.1 Session Management Controls

The Session Management page is accessible from the Configuration section of the Oracle Access Management Console.

[Figure 16-2](#) illustrates the Session Management page. Additional details follow the figure.

Figure 16-2 Common Configuration: Session Management Page



[Table 16-12](#) describes Session Management page and Search controls that enable you to create a query that is based on filter conditions.

Table 16-12 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 16-12 (Cont.) Session Management Controls and the Results Table


Name	Description
Saved Search	<p>Drop down menu that lists any search criteria saved previously for reuse. The list of searches is made available whenever you save search criteria.</p> <p>The drop down menu also has a Personalize ... option in addition to the saved searches. If you choose Personalize, you can change the behavior of the saved search criteria by making new choices such as Set as Default or Run Automatically.</p>
Match All Any	<p>Enables you to match either any of the criteria you have specified or match all of the criteria you have specified during the search.</p> <p>Note: When a resource is protected by AnonymousScheme, it is not displayed in a session search.</p>
User ID	<p>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. A drop down menu includes options like Starts With, Equals, Contains and the like to assist in your search.</p>
Client IP Address	<p>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.</p>
Search	<p>Click this button to initiate a search based on criteria in the form.</p>
<div style="border-left: 2px solid #0070C0; padding-left: 10px;">  Note: The search query is ANDed together and the results are returned. </div>	
Reset	<p>Click this button to clear the form of all criteria.</p>
Add Fields	<p>Displays a drop down menu from which you can select different options to add to your search form. This can include Client IP Address, ID Store, Impersonating and other options.</p> <ol style="list-style-type: none"> 1. Click the Add Fields button. 2. Select items in the list to add them to the form and click Save. <p>After adding an item, notice that a list is available to assist with the search.</p>
Reorder	<p>Displays a pop-up dialog allowing you to reorder the search fields.</p>
View	<p>Choose commands from the View menu above the results table to configure the table. Commands include:</p> <ul style="list-style-type: none"> • Columns: Displays a menu with the following options you can use to hide or display specific details in the table: • Detach: Expands the results table to a full-screen view • Attach: Restores the Session Management page view. • Reorder Columns: Specifies a new order for columns containing session data in the results table.

Table 16-12 (Cont.) Session Management Controls and the Results Table

Name	Description
Delete	<p>Choose this command button (red X) after selecting items in the results table to delete.</p> <p>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).</p> <p>Also: A Confirmation window appears where you can confirm or decline the operation.</p>
Detach	<p>Click Detach to expand the results table to a full-page view.</p> <p>Note: If the table is already a detached full-page, click Detach to restore the Session Management page.</p>
Results table (not named)	<p>After searching for the active sessions of a specific user, results are displayed in the table. Details include:</p> <ul style="list-style-type: none"> • 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

 **Note:**

- The option *isunset* against a field selected in the search criteria, returns sessions, for which the corresponding field value has not been set. For example, if your search criteria is *Client IP Address* with option *unset*, the search returns the sessions, for which the Client IP address is not set.
- The option *isnull* against a field selected in the search criteria, returns sessions, for which the corresponding field value is set as null.

For example, if your search criteria is *User Id* with option *isnull*, the search returns the sessions with UserID value equal to null.

16.5.2 Locating and Managing Active Sessions

Users with valid Administrator credentials can 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:**

["Session Management Controls"](#)

Skip any steps that do not apply to your requirements. The OAM Server must be running.

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Session Management**.
The Session Management Search page appears with the Username field and a results table.
3. **Add Fields:** From the Add Fields list, choose the desired field name ([Table 16-12](#)).
4. **Choose Operators:** Open the list of operators for the chosen search field, and choose the desired function.
5. **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.
6. **Configure the Results Table:** Use functions on the View menu to create the desired results table.
7. **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.
8. **Delete sessions for all users:**
 - a. Click the **Delete All Sessions** button in the upper-right corner.
 - b. Click **Yes** when you are asked to confirm.
9. Close the Session Management page when you finish.
10. Proceed to ["Validating Server-Side Session Operations"](#).

16.6 Validating Server-Side Session Operations

You can verify your configured session lifecycle operations.

1. Authenticate:
 - a. Access a resource from a browser using a credential other than your Administrative credential.

- b. Verify that the session exists, as described in "[Locating and Managing Active Sessions](#)".
2. Multiple Sessions:
 - a. From a second browser (with cookies removed), access the same resource.
 - b. Verify that two sessions exist.
3. Delete all sessions, (Step 7 of "[Locating and Managing Active Sessions](#)") and confirm that the Active sessions are removed.
4. Re-authentication Verification:
 - a. From the second browser (Step 2), access a different resource to confirm that you must re-authenticate.
 - b. Enter credentials for the resource.
 - c. Verify that a session was created.
5. Database Verification:
 - a. Delete all sessions.
 - b. Connect to the database and run the following query:


```
SQL> select * from am_session
```
 - c. Confirm that you see the following results:


```
no row selected
```
 - d. From the second browser, access a different resource.
 - e. Connect to the database and run the following query


```
SQL> select * from am_session
```
 - f. Confirm that you see one row of data:


```
1 rows selected
```
 - g. Select rows from OAM_SESSION_ATTRIBUTES and confirm that data exists for the user.

16.7 Using REST APIs for CRUD Operations on a Session

Session REST APIs manage sessions within a data center. Generally, CRUD operations on a session are performed in the session layer. Without adding new APIs or capabilities to the session layer, the Session REST APIs help Create, Read, Update, and Delete sessions. Administrators with *Session REST API* User role can use Session REST API for CRUD operations on a session.

Following are the tasks that the Administrator can perform using the Session REST APIs:

- [Searching Sessions Using Session REST API](#)
- [Updating a Session Using Session REST API](#)
- [Deleting a Session Using Session REST API](#)
- [Assigning a Delegated Administrator](#)

16.7.1 Searching Sessions Using Session REST API

Administrators with *Session REST API User* role can search for user sessions using `searchSessions` API.

Use the `userId` to search for sessions pertaining to a particular user. To search for a particular session, use the `sessionId`. If you have the `clientIp`, you can search for all the sessions coming from that client address.

Note:

Searching a session lists out only 28 sessions per query and the displayed search results are not paginated.

1. Provide `userId` to search for the sessions of a particular user.

Sample request

```
curl -H "Content-Type: application/json" -H "Authorization: Basic
<Base64 encoded auth header>" -X POST -d '{"userId":"user2"}'
http://<HOST>:<PORT>/oam/services/rest/access/api/v1/sessions
```

Sample response

```
<?xml version="1.0" encoding="UTF-8"?>
<SessionResults>
<totalRecords>2</totalRecords>
<sessions>
<sessionData>
<sessionId>53f96ca1-e65a-47ee-a758-a1cebb63f4b3|
L+h1SktCgMtjnOnmIWL+gnpIstrT9hGuPD0beqGK5Cc=</sessionId>
<createTime>2017-05-31T13:56:19.000-07:00</createTime>
<updateTime>2017-05-31T13:56:19.000-07:00</updateTime>
<lastAccessTime>2017-05-31T13:56:19.000-07:00</lastAccessTime>
<userId>user2</userId>
<clientIp>5.6.7.8</clientIp>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
<sessionData>
<sessionId>a3d62e11-3f22-4336-b76c-e8f8cbd46306|
L+h1SktCgMtjnOnmIWL+gnpIstrT9hGuPD0beqGK5Cc=</sessionId>
<createTime>2017-05-31T13:55:43.000-07:00</createTime>
<updateTime>2017-05-31T13:55:43.000-07:00</updateTime>
<lastAccessTime>2017-05-31T13:55:43.000-07:00</lastAccessTime>
<userId>user2</userId>
<clientIp>1.2.3.4</clientIp>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
</sessions>
```



```
</SessionResults>
```

2. Provide `clientId` to search for the sessions of a particular client address.

Sample request

```
curl -H "Content-Type: application/json" -H "Authorization: Basic <Base64
encoded auth header>" -X POST -d '{"clientId":"1.2.3.4"}' http://
<HOST>:<PORT>/oam/services/rest/access/api/v1/sessions
```

Sample response

```
<?xml version="1.0" encoding="UTF-8"?>
<SessionResults>
<totalRecords>4</totalRecords>
<sessions>
<sessionData>
<sessionId>a9cacf25-18cc-4d72-9237-e32394fa294e|
U90idWYSK4hXcdo6LlVD2+JuHBLvbGtCbbhlfmoDvMA=</sessionId>
<createTime>2017-05-31T13:57:32.000-07:00</createTime>
<updateTime>2017-05-31T13:57:32.000-07:00</updateTime>
<lastAccessTime>2017-05-31T13:57:32.000-07:00</lastAccessTime>
<userId>user5</userId>
<clientId>1.2.3.4</clientId>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
<sessionData>
<sessionId>32de23f1-9f53-47e7-b4d5-9d0376187241|
c+4quN74tM7P6qvbYVqa6BQOg3RYHdsFT3PzPajvEzM=</sessionId>
<createTime>2017-05-31T13:56:35.000-07:00</createTime>
<updateTime>2017-05-31T13:56:35.000-07:00</updateTime>
<lastAccessTime>2017-05-31T13:56:35.000-07:00</lastAccessTime>
<userId>user3</userId>
<clientId>1.2.3.4</clientId>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
<sessionData>
<sessionId>a3d62e11-3f22-4336-b76c-e8f8cbd46306|
L+h1SktCgMtjnOnmIWL+gnpIstrT9hGuPD0beqGK5Cc=</sessionId>
<createTime>2017-05-31T13:55:43.000-07:00</createTime>
<updateTime>2017-05-31T13:55:43.000-07:00</updateTime>
<lastAccessTime>2017-05-31T13:55:43.000-07:00</lastAccessTime>
<userId>user2</userId>
<clientId>1.2.3.4</clientId>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
<sessionData>
<sessionId>c639fb4d-f467-4da3-88c8-0e6754d809a7|
hUNJ4YlF30b5ycVlEq7iZ8LwFa8i9G5QhmKdkr3TizU=</sessionId>
<createTime>2017-05-31T13:57:17.000-07:00</createTime>
<updateTime>2017-05-31T13:57:17.000-07:00</updateTime>
<lastAccessTime>2017-05-31T13:57:17.000-07:00</lastAccessTime>
```

```

<userId>user4</userId>
<clientId>1.2.3.4</clientId>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
</sessions>
</SessionResults>

```

16.7.2 Updating a Session Using Session REST API

Administrators with *Session REST API User role* can update a session using `updateSessions` API.

To extend or shorten the session validity (`expiryTime`) for a particular user session that comes from a specific IP, the Administrator can use `updateSessions` API.

1. Search for the user's session using `userId` and `clientId`. See [Searching Sessions Using Session REST API](#).
2. Enter the `sessionId` and `expiryTime` in the session expiry instant.

Sample request

```

curl -H "Content-Type: application/json" -H "Authorization: Basic
<Base64 encoded auth header>" -X PUT -d '{ "sessionId":
"da9c1056-9216-4bb2-8946-d15da0574c6f|
U90idWYSK4hXcdo6LlVD2+JuHBLvbGtCbbhlfmoDvMA=", "expiryTime":
"2017-06-01T22:26:40.425-07:00", "userId": "user5" }' http://
<HOST>:<PORT>/oam/services/rest/access/api/v1/session

```

Sample response

```

<?xml version="1.0" encoding="UTF-8"?>
<SessionData>
<sessionId>da9c1056-9216-4bb2-8946-d15da0574c6f|
U90idWYSK4hXcdo6LlVD2+JuHBLvbGtCbbhlfmoDvMA=</sessionId>
<createTime>2017-05-31T14:26:40.425-07:00</createTime>
<updateTime>2017-05-31T14:42:14.502-07:00</updateTime>
<lastAccessTime>2017-05-31T14:42:14.502-07:00</lastAccessTime>
<expiryTime>2017-05-31T22:26:40.425-07:00</expiryTime>
<userId>user5</userId>
<clientId>1.2.3.4</clientId>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</SessionData>

```

The session is updated using `updateSessions` API.

16.7.3 Deleting a Session Using Session REST API

The Administrator with *Session REST API User role* can delete one or more user sessions using `deleteSession` REST API.

1. To delete a single session, use `sessionId` in the `deleteSession` REST API as follows:

Sample request

```
curl -H "Content-Type: application/json" -H "Authorization: Basic <Base64 encoded auth header>" -X "DELETE" "http://<HOST>:<PORT>/oam/services/rest/access/api/v1/session?sessionId=3b844e4d-9019-4928-9dca-3ba2ebbf475d%7CU90idWYSK4hXcdo6LlVD2%2BJuHBLvbGtCbhlfmoDvMA%3D"
```

Sample response

```
<?xml version="1.0" encoding="UTF-8"?>
<SessionResults>
<totalRecords>1</totalRecords>
<sessions>
<sessionData>
<sessionId>3b844e4d-9019-4928-9dca-3ba2ebbf475d|
U90idWYSK4hXcdo6LlVD2+JuHBLvbGtCbhlfmoDvMA=</sessionId>
<createTime>2017-05-31T13:57:59.545-07:00</createTime>
<updateTime>2017-05-31T13:57:59.545-07:00</updateTime>
<lastAccessTime>2017-05-31T13:57:59.545-07:00</lastAccessTime>
<expiryTime>2017-05-31T21:57:59.545-07:00</expiryTime>
<userId>user5</userId>
<clientIp>5.6.7.8</clientIp>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
</sessions>
</SessionResults>
```

 **Note:**

The `sessionId` is encoded as it contains special characters.

2. When a user is terminated, the Administrator can search for all the user's sessions using the `userId` (or optionally using `id-store`) and delete all the sessions as follows:

Sample request

```
curl -H "Content-Type: application/json" -H "Authorization: Basic <Base64 encoded auth header>" -X "DELETE" "http://<HOST>:<PORT>/oam/services/rest/access/api/v1/session?userId=user3"
```

Sample response

```
<?xml version="1.0" encoding="UTF-8"?>
<SessionResults>
<totalRecords>2</totalRecords>
<sessions>
<sessionData>
<sessionId>32de23f1-9f53-47e7-b4d5-9d0376187241|
c+4quN74tM7P6qvbYVqa6BQOg3RYHdsFT3PzPajvEzM=</sessionId>
<createTime>2017-05-31T13:56:35.426-07:00</createTime>
```

```
<updateTime>2017-05-31T13:56:35.426-07:00</updateTime>
<lastAccessTime>2017-05-31T13:56:35.426-07:00</lastAccessTime>
<expiryTime>2017-05-31T21:56:35.426-07:00</expiryTime>
<userId>user3</userId>
<clientIp>1.2.3.4</clientIp>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
<sessionData>
<sessionId>3e8fd79e-03b8-4c90-bf4d-964119460a0a|
c+4quN74tM7P6qvbYVqa6BQ0g3RYHdsFT3PzPajvEzM=</sessionId>
<createTime>2017-05-31T13:56:52.918-07:00</createTime>
<updateTime>2017-05-31T13:56:52.918-07:00</updateTime>
<lastAccessTime>2017-05-31T13:56:52.918-07:00</lastAccessTime>
<expiryTime>2017-05-31T21:56:52.918-07:00</expiryTime>
<userId>user3</userId>
<clientIp>5.6.7.8</clientIp>
<idStoreName>UserIdentityStore1</idStoreName>
<isImpersonating>>false</isImpersonating>
</sessionData>
</sessions>
</SessionResults>
```

 **Note:**

The Administrator can use `searchSessions` API along with the `clientIP` parameter to list all the sessions and delete sessions from a specific client address.

 **See Also:**

[Searching Sessions Using Session REST API](#)

16.7.4 Assigning a Delegated Administrator

To the specified users who are defined in LDAP, grant 'Session-REST API User' role before adding them to the group of administrators managing user sessions.

Only a weblog administrator or a user with *Session-REST API User* role can perform Session REST API functions on OAM runtime Sever. Through the delegated Admin page, from the `/oamconsole`, add users and assign user roles to the specific users.

Part V

Implementing Multi-Data Centers

Oracle Access Management allows for distribution of identical copies of directory service data across more than one data center. Administrators implement and manage Multi-Data Center environments.

These multiple data centers (referred to as multi-data centers) provide a scalable deployment model to support access management requirements for millions of users. This section contains the following chapters:

- [Understanding Multi-Data Centers](#)
- [Configuring Multi-Data Centers](#)
- [Synchronizing Data In A Multi-Data Center](#)
- [Setting Up the Multi-Data Center: A Sequence](#)

Understanding Multi-Data Centers

Oracle Access Manager allows for distribution of identical copies of directory service data across more than one data center. These multiple data centers (referred to as *multi-data centers*) provide a scalable deployment model to support access management requirements for millions of users.

The Access Manager Multi-Data Center (MDC) topology scales horizontally - within a single data center by clustering multiple nodes, or across multiple data centers. This model provides for load balancing as well as failover capabilities in the case that one of the nodes or data centers goes down. This chapter contains introductory details.

- [Introducing the Multi-Data Center](#)
- [Multi-Data Center Deployments](#)
- [Active-Active Multi-Data Center Topology Deployment](#)
- [Load Balancing Between Access Management Components](#)
- [Understanding Time Outs and Session Syncs](#)
- [Replicating a Multi-Data Center Environment](#)
- [Multi-Data Center Recommendations](#)

17.1 Introducing the Multi-Data Center

Large organizations using Access Manager typically deploy their applications across Multi-Data Centers (MDC) to distribute load as well as address disaster recovery. Deploying Access Manager in Multi-Data Centers allows for the transfer of user session details transparently after configuration of Single Sign-On (SSO) between them.

The scope of a data center comprises protected applications, WebGate agents, Access Manager servers, and other infrastructure entities including identity stores and databases.

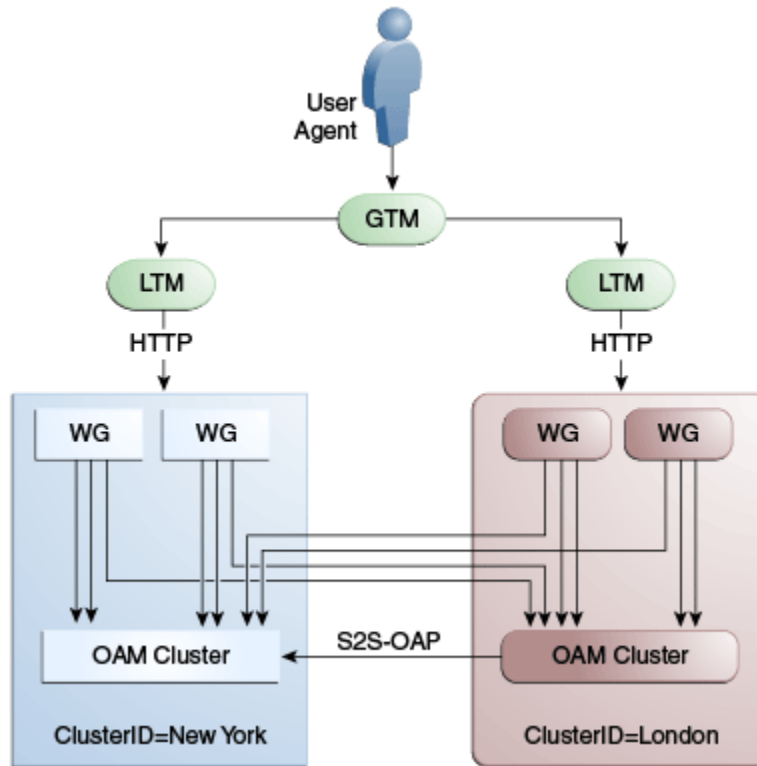
**Note:**

Access Manager supports scenarios where applications are distributed across two or more data centers.

The Multi-Data Center approach supported by Access Manager is a Master-Clone deployment in which the first data center is specified as the Master and one or more Clone data centers mirror it. (Master and Clone data centers can also be referred to as Supplier and Consumer data centers). During setup of the Multi-Data Center, session adoption policies are configured to determine where a request would be sent if the Master Data Center is down. Following the setup, a manner of replicating data from the Master to the Clone(s) will be designated. This can be done using the Automated Policy Sync (APS) Replication Service or it can be done manually.

A data center may include applications, data stores, load balancers, and the like. Each data center includes a full Access Manager installation. The WebLogic Server domain in which the instance of Access Manager is installed will not span data centers. Additionally, data centers maintain user to data center affinity. Figure 17-1 illustrates the Multi-Data Center system architecture.

Figure 17-1 Multi-Data Center System Architecture



Note:

Global load balancers are configured to route HTTP traffic to the geographically closest data center. No load balancers are used to manage Oracle Access Protocol traffic.

All applications are protected by WebGate agents configured against Access Manager clusters in the respective local data centers. Every WebGate has a primary server and one or more secondary servers; WebGate agents in each data center have Access Manager server nodes from the same data center in the primary list and nodes from other data centers in the secondary list.

Thus, it is possible for a user request to be routed to a different data center when:

- a local data center goes down
- there is a load spike causing redistribution of traffic

- certain applications are deployed in only one data center
- WebGates are configured to load balance within one data center but failover across data centers

This section describes the following topics:

- [Understanding Cookies for Multi-Data Center](#)
- [Session Adoption During Authorization](#)
- [Session Indexing](#)
- [Supported Multi-Data Center Topologies](#)

 **See Also:**

- [Configuring Multi-Data Centers](#)
- [Synchronizing Data In A Multi-Data Center](#)

17.1.1 Understanding Cookies for Multi-Data Center

SSO cookies (`OAM_ID`, `OAMAuthn`, and `OAM_GITO`) are enhanced and used by the Multi-Data Centers.

This section describes the following topics:

- [OAM_ID Cookie](#)
- [OAMAuthn WebGate Cookie](#)
- [OAM_GITO \(Global Inactivity Time Out\) Cookie](#)

17.1.1.1 OAM_ID Cookie

The `OAM_ID` cookie is the SSO cookie for Access Manager and holds the attributes required to enable the MDC behavior across all data centers.

If a subsequent request from a user in the same SSO session is routed to a different data center in the MDC topology, session adoption is triggered per the configured session adoption policies. ‘Session adoption’ refers to the action of a data center creating a local user session based on the submission of a valid authentication cookie (`OAM_ID`) that indicates a session for the user exists in another other data center in the topology. (It may or may not involve re-authentication of the user.)

When a user session is created in a data center, the `OAM_ID` cookie will be augmented or updated with the following:

- `clusterid` of the data center
- `sessionid`
- `latest_visited_clusterid`

In MDC deployments, `OAM_ID` is a host-scoped cookie. Its domain parameter is set to a virtual host name which is a singleton across data centers and is mapped by the global load balancer to the Access Manager servers in the Access Manager data center based on the

load balancer level user traffic routing rules (for example, based on geographical affinity). The `OAM_ID` cookie is not accessible to applications other than the Access Manager servers.

17.1.1.2 OAMAuthn WebGate Cookie

OAMAuthn is the WebGate cookie for 11g and 12c. On successful authentication and authorization, a user will be granted access to a protected resource. At that point, the browser will have a valid WebGate cookie with the `clusterid:sessionid` of the authenticating data center.

If authentication followed by authorization spans across multiple data centers, the data center authorizing the user request will trigger session adoption by retrieving the session's originating `clusterid` from the WebGate cookie. (WebGates need to have the same host name in each data center due to host scoping of the WebGate cookies.) After adopting the session, a new session will be created in the current data center with the synced session details.



Note:

The WebGate cookie cannot be updated during authorization hence the newly created `sessionid` cannot be persisted for future authorization references. In this case, the remote `sessionid` and the local `sessionids` are linked through session indexing. During a subsequent authorization call to a data center, a new session will be created when:

- MDC is enabled
- a session matching the `sessionid` in the WebGate cookie is not present in the local data center
- there is no session with the `Session Index` that matches the `sessionid` in the WebGate cookie
- a valid session exists in the remote data center (based on the MDC SessionSync Policy)

In these instances, a new session is created in the local data center with a `Session Index` that refers to the `sessionid` in the WebGate cookie.

17.1.1.3 OAM_GITO (Global Inactivity Time Out) Cookie

OAM_GITO is a domain cookie set as an authorization response.

The session details of the authentication process will be recorded in the `OAM_ID` cookie. If the authorization hops to a different data center, session adoption will occur by creating a new session in the data center servicing the authorization request and setting the session index of the new session as the incoming `sessionid`. Since subsequent authentication requests will only be aware of the `clusterid:sessionid` mapping available in the `OAM_ID` cookie, a session hop to a different data center for authorization will go unnoticed during the authentication request. To address this gap, an `OAM_GITO` cookie (which also facilitates time out tracking across WebGate agents) is introduced.

During authorization, the `OAM_GITO` cookie is set as a domain cookie. For subsequent authentication requests, the contents of the `OAM_GITO` cookie will be read to determine the latest session information and the inactivity or idle time out values. The `OAM_GITO` cookie contains the following data.

- Data Center Identifier
- Session Identifier
- User Identifier
- Last Access Time
- Token Creation Time

 **Note:**

For the `OAM_GITO` cookie, all WebGates and Access Manager servers should share a common domain hierarchy. For example, if the server domain is `.us.example.com` then all WebGates must have (at least) `.example.com` as a common domain hierarchy; this enables the `OAM_GITO` cookie to be set with the `.example.com` domain.

17.1.2 Session Adoption During Authorization

Multi-Data Center session adoption is supported during the authorization flow. After successful authentication, the `OAMAuthn` cookie will be augmented with the cluster ID details of the data center where authentication has taken place.

During authorization, if the request is routed to a different data center, Access Manager runtime checks for a valid remote session and determines if it is a Multi-Data Center scenario. When a valid remote session is located, the Multi-Data Center session adoption process is triggered per the session adoption policy.

The session adoption policy can be configured so that the clone Access Manager cluster would make a back-end request for session details from the master Access Manager cluster using the Oracle Access Protocol (OAP). The session adoption policy can also be configured to invalidate the previous session so the user has a session only in one data center at a given time. Following the session adoption process, a new session will be created in the data center servicing the authorization request.

 **Note:**

Since `OAMAuthn` cookie updates are not supported during authorization, the newly created session's `Session Index` will be set to that of the incoming `Session ID`.

 **See Also:**

- [Multi-Data Center Deployments](#)
- [Session Indexing](#)

17.1.3 Session Indexing

The Session Index refers to the session identifier in the OAMAuth cookie.

A new session with a Session Index is created in the local data center during an authorization call to a data center. It occurs when the following conditions are met:

- Session matching Session ID in the OAMAuth cookie is not present in the local data center.
- MDC is enabled.
- No session with Session Index matching Session ID in the OAMAuth cookie.
- Valid session exists in the remote data center based on the MDC Session Sync Policy.

17.1.4 Supported Multi-Data Center Topologies

Access Manager supports three Multi-Data Center topologies: Active-Active, Active-Passive, and Active-Standby modes.

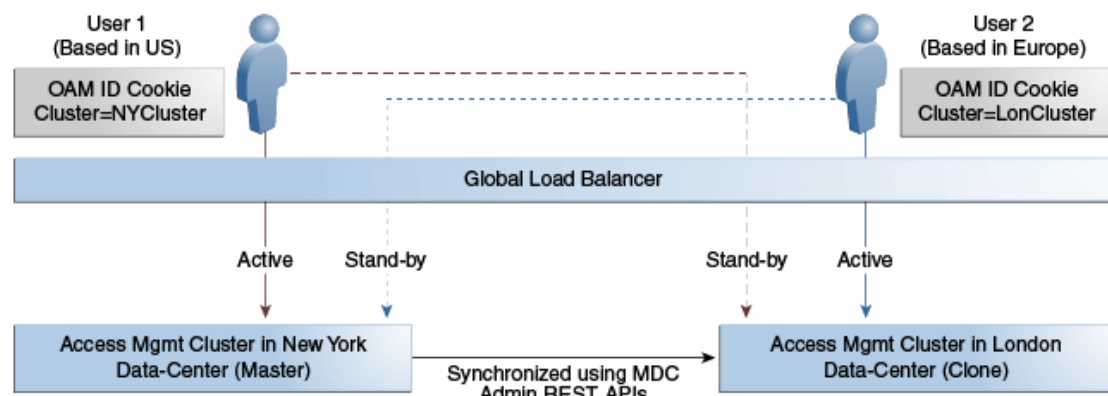
The following sections contain details on these modes.

- [The MDC Active-Active Mode](#)
- [The MDC Active-Passive Mode](#)
- [The MDC Active-Hot Standby Mode](#)

17.1.4.1 The MDC Active-Active Mode

An Active-Active topology is when Master and Clone data centers are exact replicas and active at the same time.

They cater to different sets of users based on defined criteria; geography, for example. A load balancer routes traffic to the appropriate Data Center. [Figure 17-2](#) illustrates a Multi-Data Center set up in Active-Active mode during normal operations.

Figure 17-2 Active-Active Deployment Mode

In [Figure 17-2](#), the New York Data Center is designated as the Master and all policy and configuration changes are restricted to it. The London Data Center is designated as a Clone and uses REST APIs to periodically synchronize data with the New York Data Center. The global load balancer is configured to route users in different geographical locations (US and Europe) to the appropriate data centers (New York or London) based on proximity to the data center (as opposed to proximity of the application being accessed). For example, all requests from US-based User 1 will be routed to the New York Data Center (NYDC) and all requests from Europe-based User 2 will be routed to the London Data Center (LDC).

 **Note:**

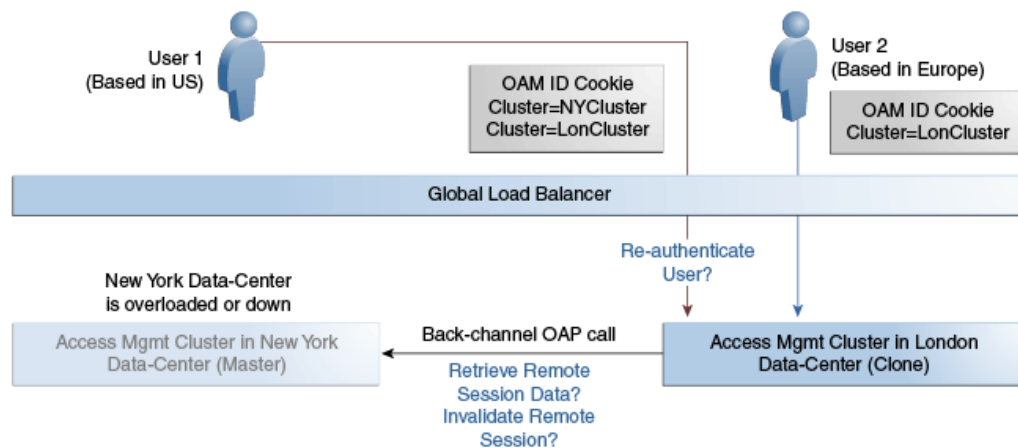
The Access Manager clusters in [Figure 17-2](#) are independent and not part of the same Oracle WebLogic domain. WebLogic domains are not recommended to span across data centers.

In this example, if NYDC was overloaded with requests, the global load balancer would start routing User 1 requests to the Clone Access Manager cluster in LDC. The Clone Access Manager cluster uses the OAM_ID cookie of the user to check for a valid session in the Master cluster. If there is a valid session in the Master cluster, a new session is created without prompting for authentication or re-authentication.

Further, the session adoption policy can be configured such that the Clone Access Manager cluster would make a back-end request for session details from the Master Access Manager cluster using the Oracle Access Protocol (OAP). The session adoption policy can also be configured to invalidate the remote session (the session in NYDC) so the user has a session only in one data center at a given time.

[Figure 17-3](#) illustrates how a user might be rerouted if the Master cluster is overloaded or down. When the Master Access Manager cluster goes completely down, the clone Access Manager cluster tries to obtain the session details of User 1. Since the Master Access Manager cluster is completely inaccessible, User 1 is forced to re-authenticate and establish a new session in the Clone Access Manager cluster. In this case, any information stored in the previous session is lost.

Figure 17-3 Active-Active Mode Failover



Note:

An Active-Active topology with agent failover is when an agent has Access Manager servers in one data center configured as primary and Access Manager servers in the other data centers configured as secondary to aid failover scenarios.

See Also:

[Active-Active Multi-Data Center Topology Deployment](#)

17.1.4.2 The MDC Active-Passive Mode

An Active-Passive topology is when the primary data center is operable but the clone data center is not. In this topology, the clone can be brought up within a reasonable time in cases when the primary data center fails. Thus, in the Active-Passive Mode one of the data centers is passive and services are not started.

In this use case, the data center does not have to be brought up immediately but within a reasonable amount of time in cases when the primary data center fails. There is no need to do an MDC setup although policy data will be kept in sync.

17.1.4.3 The MDC Active-Hot Standby Mode

Active-Hot Standby is when one of the data centers is in *hot standby* mode. In this case, traffic is not be routed to the Hot Standby data center unless the active data center goes down.

In this use case, you do not need additional data centers for traffic on a daily basis but only keep one ready. Follow the Active-Active mode steps to deploy in Active-Hot Standby Mode but do not route traffic to the center defined as Hot Standby. The Hot Standby center will continue to sync data but will only be used when traffic is directed there by the load balancer or by an administrator.

17.2 Multi-Data Center Deployments

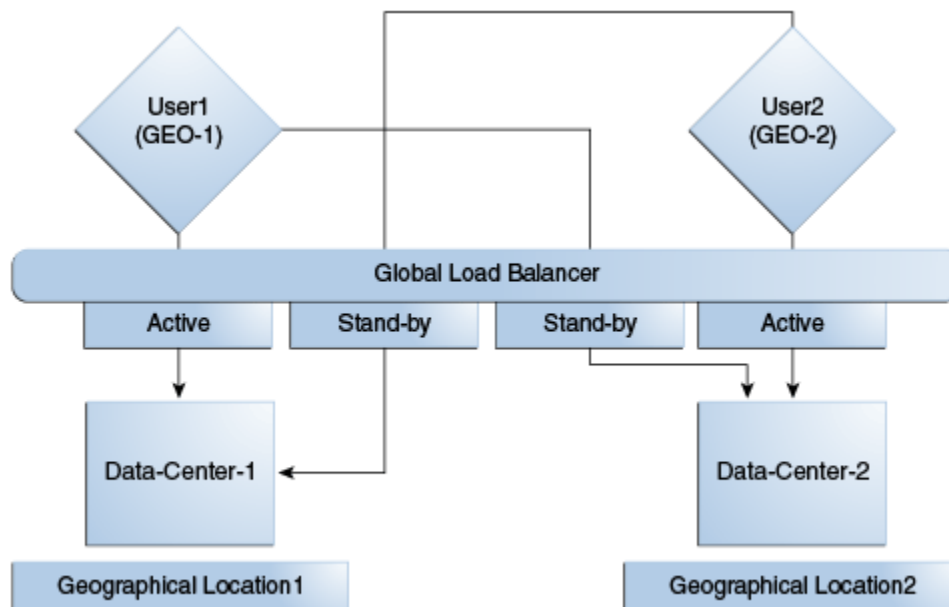
In a Multi-Data Center deployment, each data center will include a full Access Manager installation; WebLogic Server domains will not span the data centers.

Global load balancers maintain users to data center affinity although a user request may be routed to a different data center when:

- the data center goes down.
- a load spike causes redistribution of traffic.
- each data center is not a mirror of the other. (For example, certain applications may only be deployed in a single data center)
- WebGates are configured to load balance within the data center and failover across data centers.

Figure 17-4 illustrates a basic Multi-Data Center deployment.

Figure 17-4 Multi-Data Center Deployment



The following sections describe several deployment scenarios:

- [Session Adoption Without Re-authentication, Session Invalidation or Session Data Retrieval](#)
- [Session Adoption Without Re-authentication But With Session Invalidation and Session Data Retrieval](#)
- [Session Adoption Without Re-authentication and Session Invalidation But With On-demand Session Data Retrieval](#)
- [Authentication and Authorization Requests Served By Different Data Centers](#)

- [Logout and Session Invalidation](#)
- [Stretch Cluster Deployments](#)



Note:

The OAP connection used for back channel communication does not support load balancing or failover so a load balancer needs to be used.

17.2.1 Session Adoption Without Re-authentication, Session Invalidation or Session Data Retrieval

The following scenario illustrates the flow when the Session Adoption Policy is configured without re-authentication, remote session invalidation and remote session data retrieval.

It is assumed the user has affinity with DC1.

1. User is authenticated by DC1.
On successful authentication, the OAM_ID cookie is augmented with a unique data center identifier referencing DC1 and the user can access applications protected by Access Manager in DC1.
2. Upon accessing an application deployed in DC2, the user is routed to DC2 by a global load balancer.
3. Access Manager in DC2 is presented with the augmented OAM_ID cookie issued by DC1.
On successful validation, Access Manager in DC2 knows that this user has been routed from the remote DC1.
4. Access Manager in DC2 looks up the Session Adoption Policy.
The Session Adoption Policy is configured without reauthentication, remote session invalidation or remote session data retrieval.
5. Access Manager in DC2 creates a local user session using the information present in the DC1 OAM_ID cookie (lifetime, user) and re-initializes the static session information (\$user responses).
6. Access Manager in DC2 updates the OAM_ID cookie with its data center identifier.
Data center chaining is also recorded in the OAM_ID cookie.
7. User then accesses an application protected by Access Manager in DC1 and is routed back to DC1 by the global load balancer.
8. Access Manager in DC1 is presented with the OAM_ID cookie issued by itself and updated by DC2.
On successful validation, Access Manager in DC1 knows that this user has sessions in both DC1 and DC2.
9. Access Manager in DC1 attempts to locate the session referenced in the OAM_ID cookie.
 - If found, the session in DC1 is updated.

- If not found, Access Manager in DC1 looks up the Session Adoption Policy (also configured without reauthentication, remote session invalidation and remote session data retrieval).
10. Access Manager in DC1 updates the OAM_ID cookie with its data center identifier and records data center chaining as previously in DC2.

17.2.2 Session Adoption Without Re-authentication But With Session Invalidation and Session Data Retrieval

The following scenario illustrates the flow when the Session Adoption Policy is configured without re-authentication but with remote session invalidation and remote session data retrieval.

It is assumed the user has affinity with DC1.

1. User is authenticated by DC1.

On successful authentication, the OAM_ID cookie is augmented with a unique data center identifier referencing DC1.

2. Upon accessing an application deployed in DC2, the user is routed to DC2 by a global load balancer.

3. Access Manager in DC2 is presented with the augmented OAM_ID cookie issued by DC1.

On successful validation, Access Manager in DC2 knows that this user has been routed from the remote DC1.

4. Access Manager in DC2 looks up the Session Adoption Policy.

The Session Adoption Policy is configured without reauthentication but with remote session invalidation and remote session data retrieval.

5. Access Manager in DC2 makes a back-channel (OAP) call (containing the session identifier) to Access Manager in DC1 to retrieve session data.

The session on DC1 is terminated following data retrieval. If this step fails due to a bad session reference, a local session is created. See [Session Adoption Without Re-authentication, Session Invalidation or Session Data Retrieval](#).

6. Access Manager in DC2 creates a local user session using the information present in the OAM_ID cookie (lifetime, user) and re-initializes the static session information (\$user responses).

7. Access Manager in DC2 rewrites the OAM_ID cookie with its own data center identifier.

8. The user then accesses an application protected by Access Manager in DC1 and is routed to DC1 by the global load balancer.

9. Access Manager in DC1 is presented with the OAM_ID cookie issued by DC2.

On successful validation, Access Manager in DC1 knows that this user has sessions in DC2.

10. Access Manager in DC1 makes a back-channel (OAP) call (containing the session identifier) to Access Manager in DC2 to retrieve session data.

If the session is found, a session is created using the retrieved data. If it is not found, the OAM Server in DC1 creates a new session. The session on DC2 is terminated following data retrieval.

17.2.3 Session Adoption Without Re-authentication and Session Invalidation But With On-demand Session Data Retrieval

Multi-Data Center supports session adoption without re-authentication except that the non-local session are not terminated and the local session is created using session data retrieved from the remote DC.

Note that the OAM_ID cookie is updated to include an attribute that indicates which data center is currently being accessed.

17.2.4 Authentication and Authorization Requests Served By Different Data Centers

Consider a scenario where an authentication request is served by the New York Data Center (NYDC) but the authorization request is presented to the London Data Center (LDC) because of user affinity.

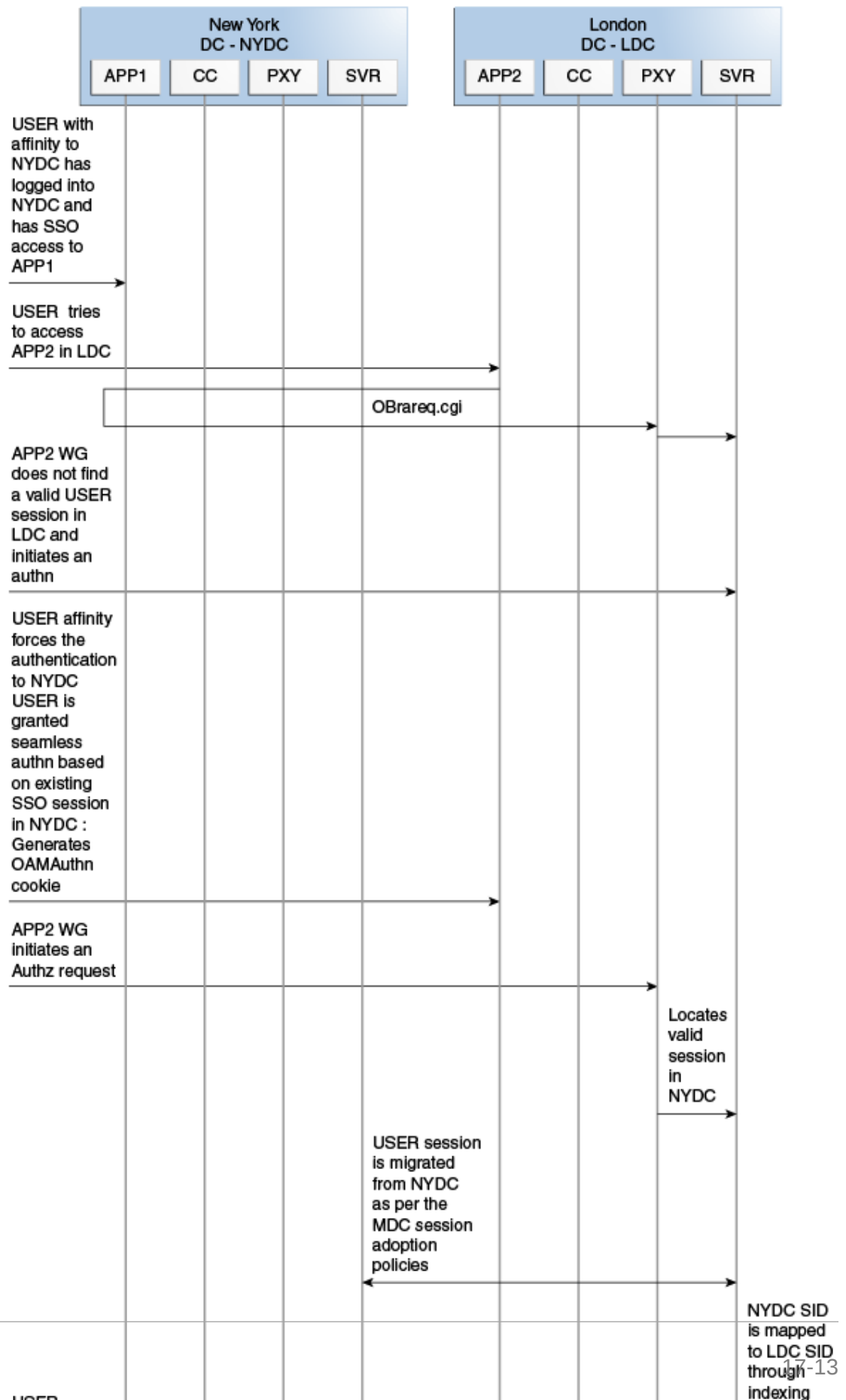
If Remote Session Termination is enabled, the scenario requires a combination of the OAM_ID cookie, the OamAuthn/ObSSO authorization cookie and the GITO cookie to perform the seamless Multi-Data Center operations. This flow (and [Figure 17-5](#) following it) illustrates this. It is assumed that the user has affinity with NYDC.

1. Upon accessing APP1, a user is authenticated by NYDC.
On successful authentication, the OAM_ID cookie is augmented with a unique data center identifier referencing NYDC. The subsequent authorization call will be served by the primary server for the accessed resource, NYDC. Authorization generates the authorization cookie with the NYDC identifier (cluster-id) in it and the user is granted access to the APP1.
2. User attempts to access APP2 in LDC.
3. The WebGate for APP2 finds no valid session in LDC and initiates authentication.
Due to user affinity, the authentication request is routed to NYDC where seamless authentication occurs. The OamAuthn cookie contents are generated and shared with the APP2 WebGate.
4. The APP2 WebGate forwards the subsequent authorization request to APP2's primary server, LDC with the authorization cookie previously generated.
During authorization, LDC will determine that this is a Multi-Data Center scenario and a valid session is present in NYDC. In this case, authorization is accomplished by syncing the remote session as per the configured session adoption policies.
5. A new session is created in LDC during authorization and the incoming session id is set as the new session's index.

Subsequent authorization calls are honored as long as the session search by index returns a valid session in LDC. Each authorization will update the GITO cookie with the cluster-id, session-id and access time. The GITO cookie will be re-written as an authorization response each time.

If a subsequent authentication request from the same user hits NYDC, it will use the information in the OAM_ID and GITO cookies to determine which Data Center has the most current session for the user. The Multi-Data Center flows are triggered seamlessly based on the configured Session Adoption policies.

Figure 17-5 Requests Served By Different Data Centers



17.2.5 Logout and Session Invalidation

In Multi-Data Center scenarios, logout ensures that all server side sessions across data centers and all authentication cookies are cleared out. For session invalidation, termination of a session artifact over the back-channel will not remove the session cookie and state information maintained in the WebGates.

However, the lack of a server session will result in an Authorization failure which will result in re-authentication. In the case of no session invalidation, the logout clears all server side sessions that are part of the current SSO session across Data Centers. This flow (and [Figure 17-6](#) following it) illustrates logout. It is assumed that the user has affinity with NYDC.

1. User with affinity to NYDC gets access to APP1 after successful authentication with NYDC.

2. User attempts to access APP2.

At this point there is a user session in NYDC as well as LDC as part of SSO.

3. User logs out from APP1.

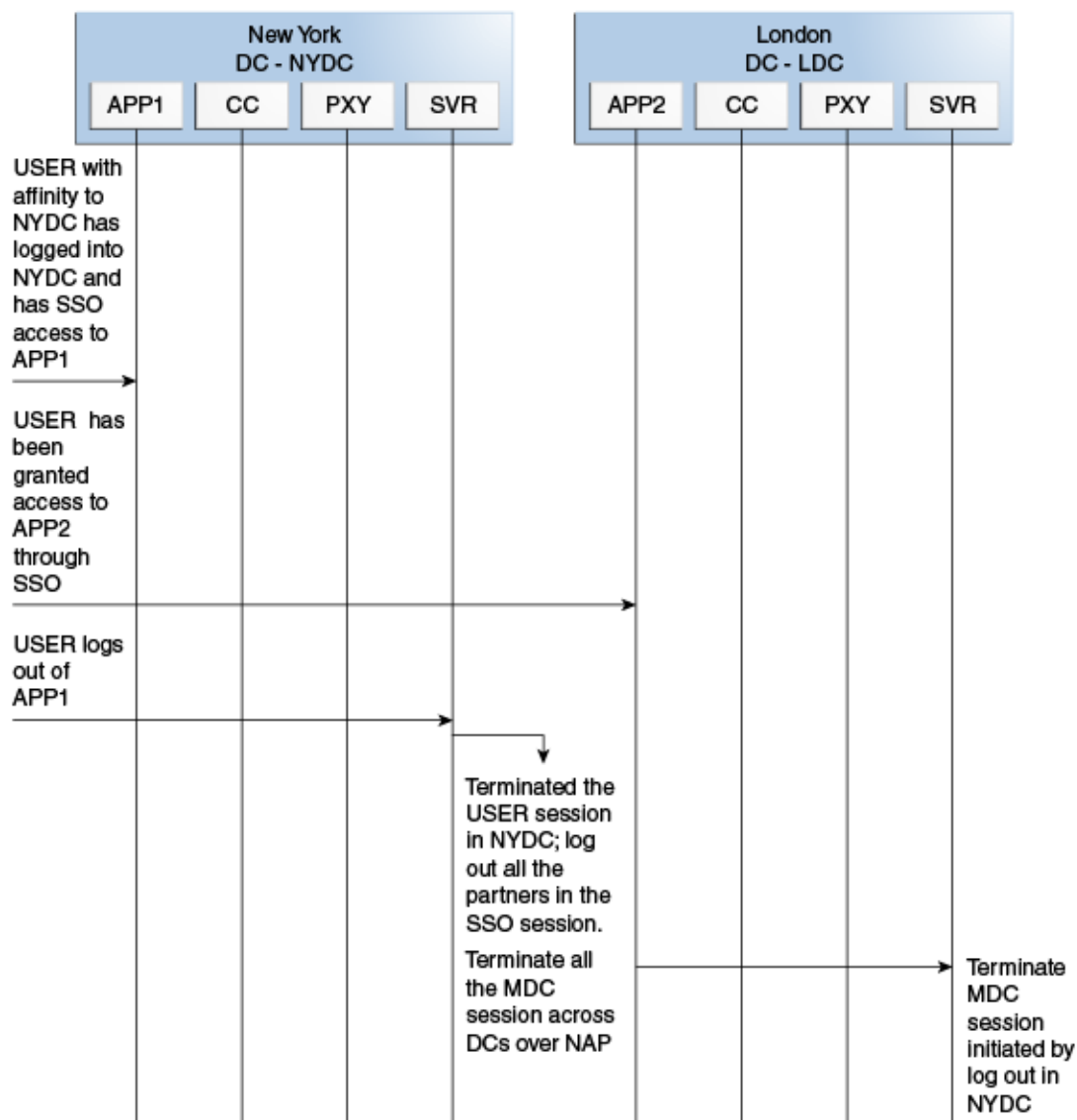
Due to affinity, the logout request will reach NYDC.

4. The NYDC server terminates the user's SSO session and logs out from all the SSO partners.

5. The NYDC server sends an OAP terminate session request to all relevant Data Centers associated with the SSO session - including LDC.

This results in clearing all user sessions associated with the SSO across Data Centers.

Figure 17-6 Logout and Session Invalidation



17.2.6 Stretch Cluster Deployments

For data centers that are geographically very close and have a guaranteed latency of less than 5 milliseconds, customers can choose a Stretch Cluster deployment.

In this case, unlike the traditional multi data center deployment described in the preceding sections, a single OAM cluster is stretched across multiple data centers; there are some OAM nodes in one data center and the remaining nodes in another data center. Though the deployment is spread across two data centers, Access Manager treats this as a single cluster. The policy database would reside in one of the data centers. The following limitations apply to a Stretch Cluster Deployment.

- Access Manager depends on the underlying WebLogic and Coherence layers to keep the nodes in sync. The latency between data centers must be less than 5 milliseconds at all times. Any spike in the latency may cause instability and unpredictable behavior.

- The cross data center chatter at runtime in a Stretch Cluster deployment is relatively more than that in the traditional multi data center deployment. In case of the latter, the runtime communication between data centers is restricted to use-cases where a session has to be adopted across data centers.
- Since it is a single cluster across data centers, it does not offer the same level of reliability/availability as a traditional multi data center deployment. The policy database can become a single point of failure. In a traditional multi data center deployment, each data center is self-sufficient and operates independent of the other data center which provides far better reliability.

Figure 17-7 illustrates a Stretch Cluster deployment while Figure 17-8 below it illustrates a traditional MDC deployment. Oracle does recommend a traditional multi data center deployment over a Stretch Cluster deployment.

Figure 17-7 Stretch Cluster Deployment

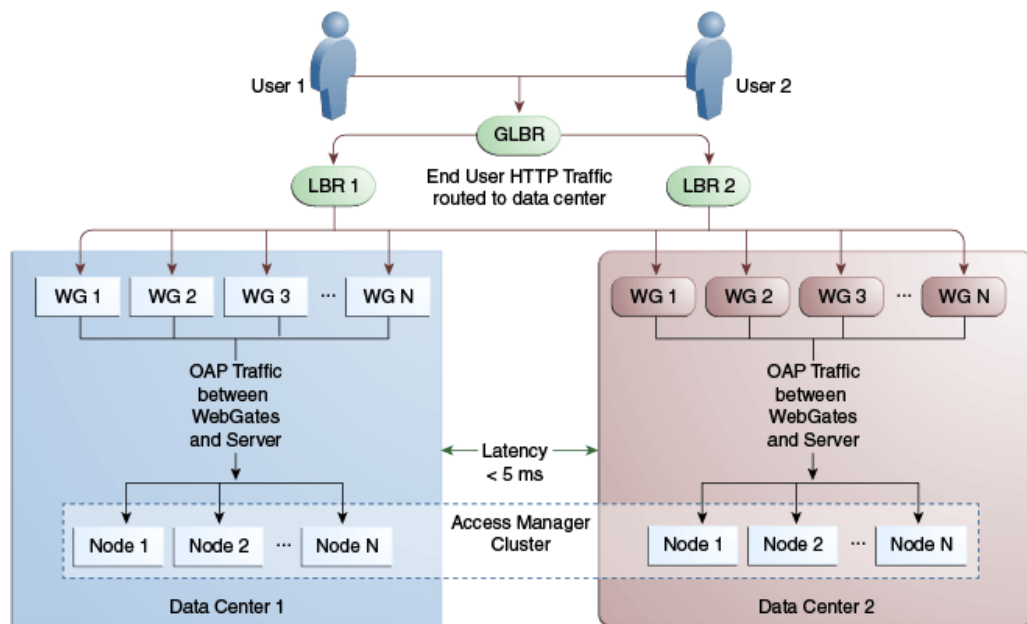
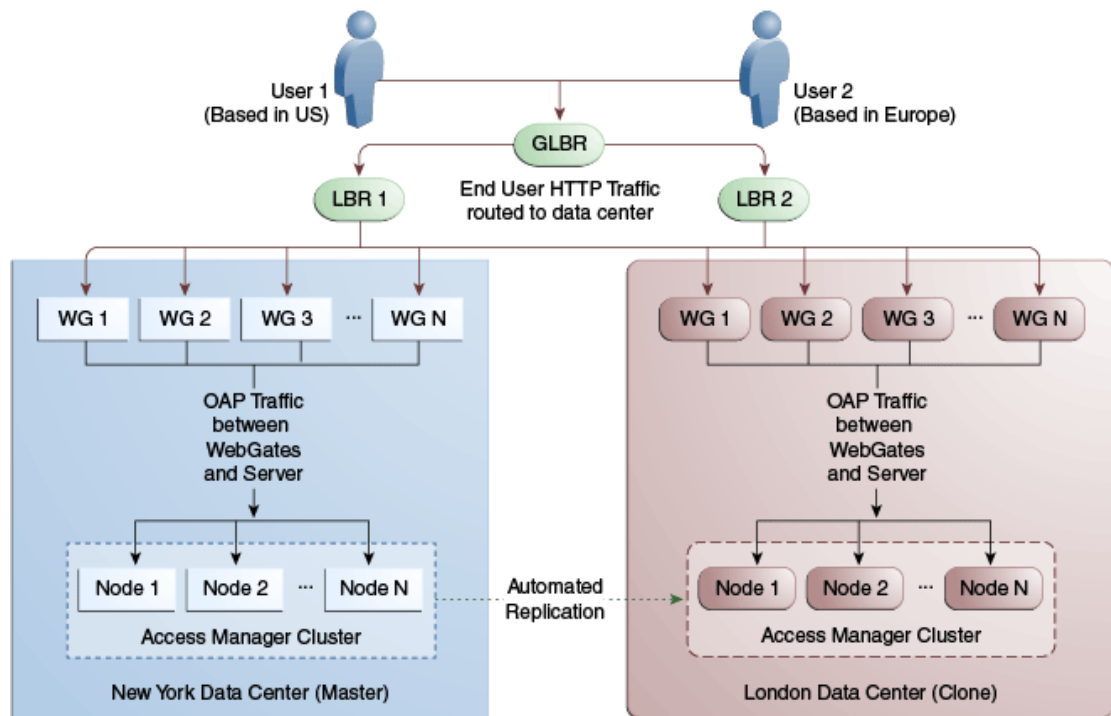


Figure 17-8 Traditional MDC Deployment



See Also:

[Supported Multi-Data Center Topologies](#)

17.3 Active-Active Multi-Data Center Topology Deployment

An Active-Active topology is when Master and Clone data Centers are exact replicas of each other (including applications, data stores and the like).

They are active at the same time and cater to different sets of users based on defined criteria - geography, for example. A load balancer routes traffic to the appropriate Data Center. Identical Access Manager clusters are deployed in both locales with New York designated as the Master and London as the Clone.



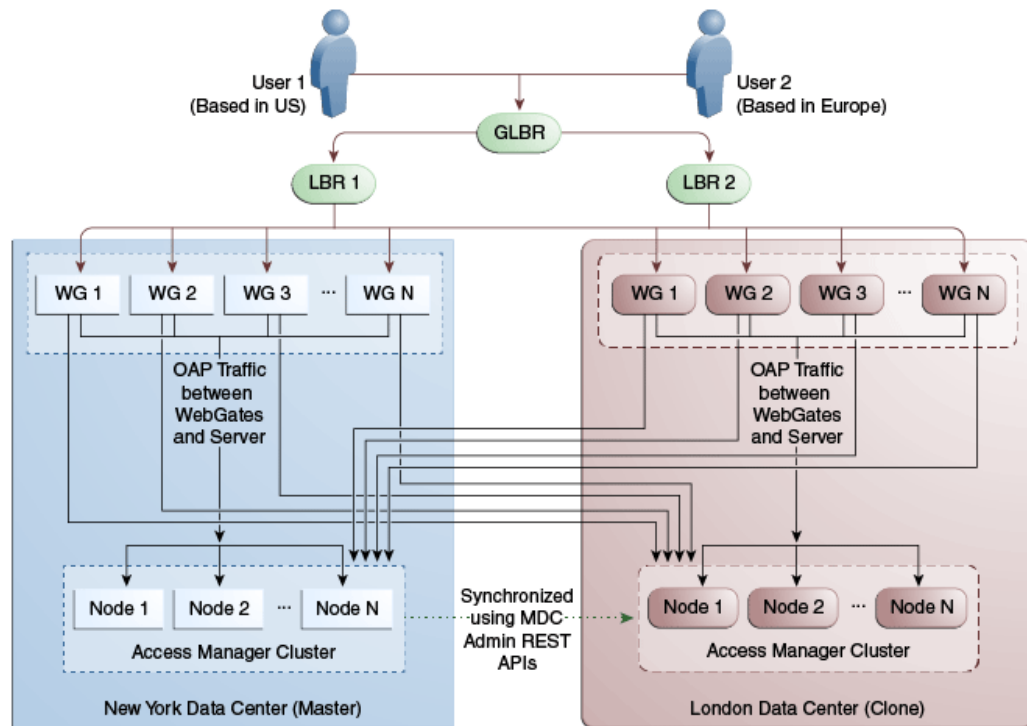
Note:

An Active-Active topology with agent failover is when an agent has Access Manager servers in one data center configured as primary and Access Manager servers in the other data centers configured as secondary to aid failover scenarios.

Figure 17-9 illustrates the topology for a Multi-Data Center deployment in Active-Active mode. The New York Data Center is designated as the Master and all policy and configuration changes are restricted to it. The London Data Center is designated as a Clone and uses

MDC admin REST APIs periodically to synchronize data with the New York Data Center. The global load balancer is configured to route users in different geographical locations (US and Europe) to the appropriate data centers (New York or Europe) based on proximity to the data center (as opposed to proximity of the application being accessed). For example, all requests from US-based User 1 will be routed to the New York Data Center (NYDC) and all requests from Europe-based User 2 will be routed to the London Data Center (LDC).

Figure 17-9 Active-Active Topology



The Global Load Balancer is configured for session stickiness so once a user has been assigned to a particular data center, all subsequent requests from that user would be routed to the same data center. In this example, User 1 will always be routed to the New York Data Center and User 2 to the London Data Center.

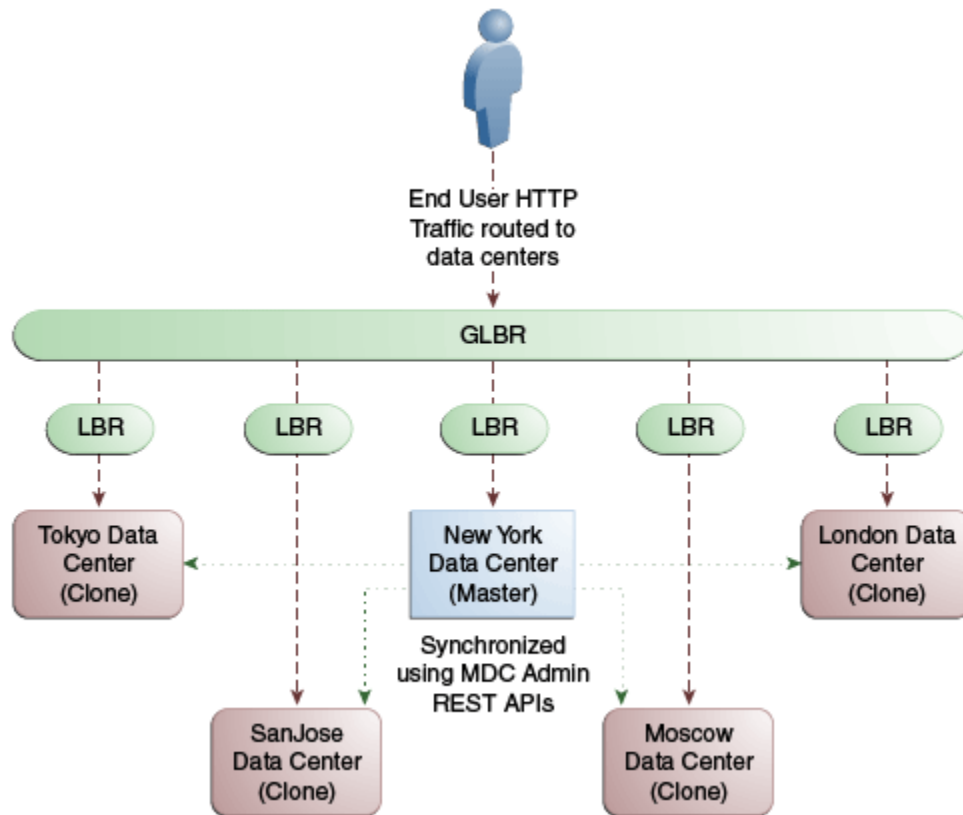
User requests in the respective data centers are intercepted by different WebGates depending on the application being accessed. Each WebGate has the various nodes of the Access Manager cluster within the same data center configured as its primary servers. In this case, the WebGates load balance and failover the local data center.

Note:

Administrators have the flexibility to configure the primary servers for every WebGate in different orders based on load characteristics. Running monitoring scripts in each data center will detect if any of the Access Manager components – the WebGates or the servers – are unresponsive so administrators can reconfigure the load balancers to direct user traffic to a different data center.

Any number of Clone data centers can be configured to distribute the load across the globe. The only condition is that all Clone data centers are synchronized from a single Master using MDC admin REST APIs. Figure 17-10 below depicts an Active-Active Multi-Data Center deployment across five data centers.

Figure 17-10 Active-Active Topology Across Multiple Data Centers



17.4 Load Balancing Between Access Management Components

The topology described earlier shows global and local load balancers for routing the end user HTTP traffic to various data centers. Customers can choose to deploy load balancers between the access manager components to simplify the configuration of the access manager components by using virtual host names.

For example, instead of configuring the primary servers in each WebGate in the NYDC as `ssonode1.ny.acme.com`, `ssonode2.ny.acme.com` and so on, they can all point to a single virtual host name like `sso.ny.acme.com` and the load balancer will resolve the DNS to direct them to various nodes of the cluster. However, while introducing a load balancer between Access Manager components, there are a few constraining requirements to keep in mind.

- OAP connections are persistent and need to be kept open for a configurable duration even while idle.
- The WebGates need to be configured to recycle their connections proactively prior to the Load Balancer terminating the connections, unless the Load Balancer is capable of

sending TCP resets to both the Webgate and the server ensuring clean connection cleanup.

- The Load Balancer should distribute the OAP connection uniformly across the active Access Manager Servers for each WebGate (distributing the OAP connections according the source IP), otherwise a load imbalance may occur.

Figure 17-11 illustrates a variation of the deployment topology with local load balancers (LBR 3 and LBR 4) front ending the clusters in each data center. These local load balancers can be Oracle HTTP Servers (OHS) with `mod_wl_ohs`. The OAP traffic still flows between the WebGates and the Access Manager clusters within the data center but the load balancers perform the DNS routing to facilitate the use of virtual host names.

See Also [Monitoring the Health of an Access Manager Server](#)

Figure 17-11 Load Balancing Access Manager Components

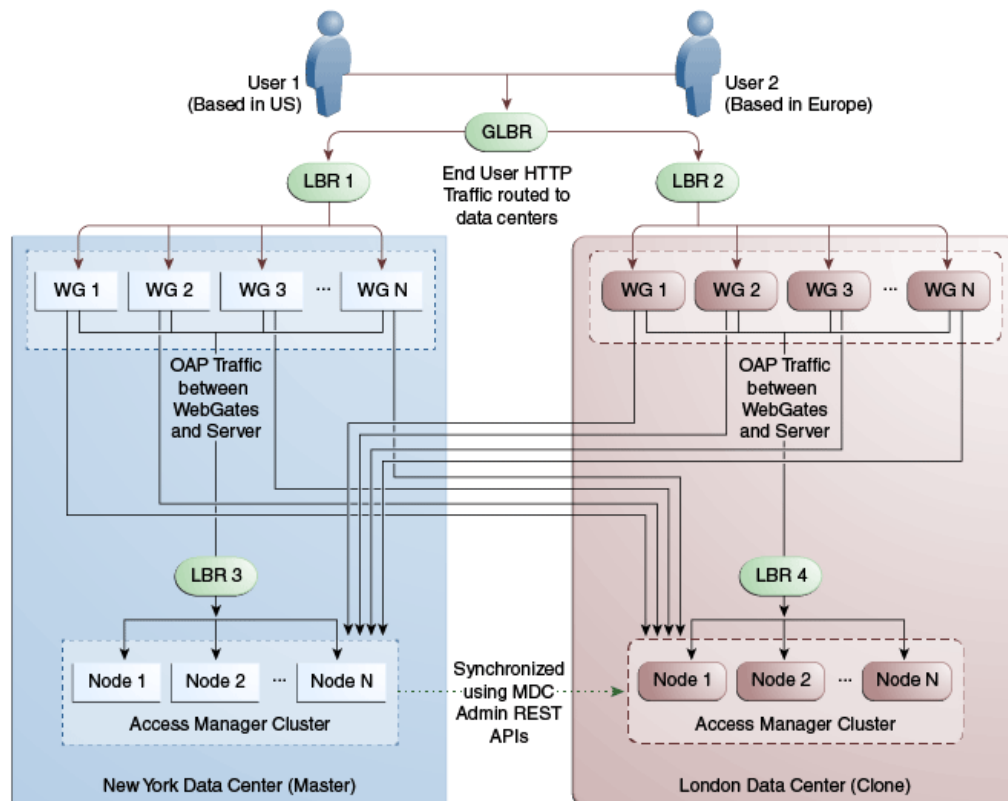
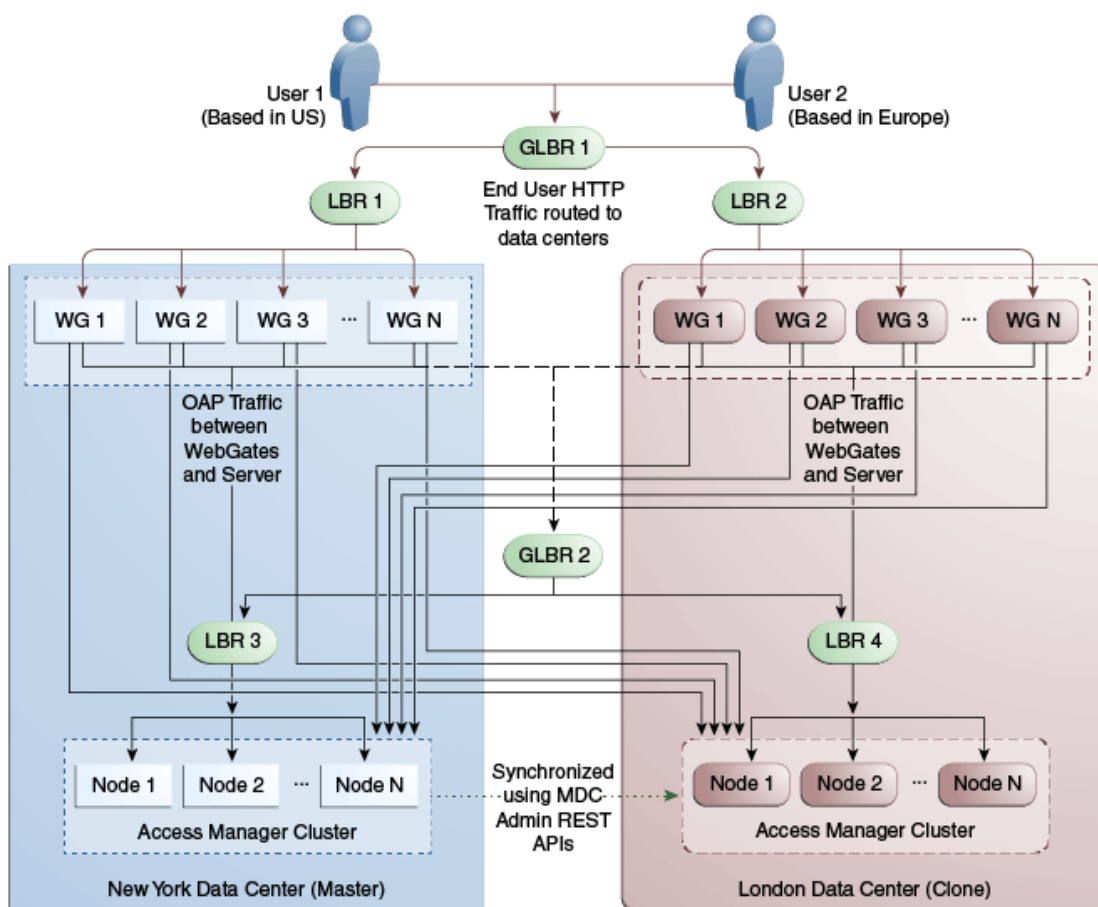


Figure 17-12 illustrates a second variation of the deployment topology with the introduction of a global load balancer (GLBR2) to front end local load balancers (LBR3 and LBR4). In this case, the host names can be virtualized not just within the data center but across the data centers. The WebGates in each data center would be configured to load balance locally but fail over remotely. One key benefit of this topology is that it guarantees high availability at all layers of the stack. Even if the entire Access Manager cluster in a data center were to go down, the WebGates in that data center would fail over to the Access Manager cluster in the other data center.

Figure 17-12 Global Load Balancer Front Ends Local Load Balancer



17.5 Understanding Time Outs and Session Syncs

The following sections contain information on how the Multi-Data Center deals with session time outs and syncs.

- [Maximum Session Constraints](#)
- [Multi-Data Center Policy Configurations for Idle Timeout](#)
- [Expiring Multi-Data Center Sessions](#)
- [Session Synchronization and Multi-Data Center Fail Over](#)

17.5.1 Maximum Session Constraints

Credential Collector user affinity ensures that maximum session constraints per user are honored.

There is no Multi-Data Center session store to validate the maximum sessions allowed per user.

17.5.2 Multi-Data Center Policy Configurations for Idle Timeout

The `OAM_ID` and `OAM_GITO` cookies are used to calculate and enforce idle (inactivity) time outs. The `OAM_GITO` cookie, though, can be set only if there is a common sub-domain across WebGates. Thus, Multi-Data Center policies should be configured based on whether or not the `OAM_GITO` cookie is set.

Table 17-1 documents the policy configurations.

Table 17-1 Multi-Data Center Policy Configurations for Idle Timeout

OAM_GITO Set	Multi-Data Center Policies
Yes Idle time out will be calculated from the latest OAM_GITO cookie	<code>SessionMustBeAnchoredToDataCenterServicingUser=<true/false></code> <code>SessionDataRetrievalOnDemand=true</code> <code>Reauthenticate=false</code> <code>SessionDataRetrievalOnDemandMax_retry_attempts=<number></code> <code>SessionDataRetrievalOnDemandMax_conn_wait_time=<milliseconds></code> <code>SessionContinuationOnSyncFailure=<true/false></code> <code>MDCGitoCookieDomain=<sub domain></code>
No Idle time out will be calculated from the OAM_ID cookie because OAM_GITO is not available	<code>SessionMustBeAnchoredToDataCenterServicingUser=false</code> <code>SessionDataRetrievalOnDemand=true</code> <code>Reauthenticate=false</code> <code>SessionDataRetrievalOnDemandMax_retry_attempts=<number></code> <code>SessionDataRetrievalOnDemandMax_conn_wait_time=<milliseconds></code> <code>SessionContinuationOnSyncFailure=<true/false></code> <code>#MDCGitoCookieDomain= This setting should be commented or removed</code>

17.5.3 Expiring Multi-Data Center Sessions

Session expiration will be managed by the data center with which the user has affinity.

Users have affinity to a particular data center based on the global traffic manager or load balancer.

17.5.4 Session Synchronization and Multi-Data Center Fail Over

When a valid session corresponding to the user's request exists in a remote data center, based on MDC session synchronization policies, the remote session attributes are migrated and synced to the data center servicing the current request. If the synchronization fails, the datacenter can still access the requested resource as the Webgate failover across data centers is supported by the MDC.

Access Manager server side sessions are created and maintained based on Single Sign-On (SSO) credentials. The attributes stored in the session include (but are not limited to) the user identifier, an identity store reference, subject, custom attributes, partner data, client IP address and authentication level. SSO will be granted if the server can locate a valid session corresponding to the user's request.

In a Multi-Data Center scenario, when a user request hops across data centers, the data center servicing the request should validate for a legitimate session locally and across data centers. If a valid session for a given request exists in a remote data center, the remote session needs to be migrated to the current data center based on the MDC session synchronization policies. (See [Multi-Data Center Deployments](#).) During this session synchronization, all session attributes from the remote session are synced to the newly created session in the data center servicing the current request.

The Multi-Data Center also supports WebGate failover across data centers. When a WebGate fails over from one data center to a second, the session data can not be synchronized because the first data center servers are down. Thus, the second data center will decide whether or not to proceed with the session adoption based on the setting configured for `SessionContinuationOnSyncFailure`. When true, even if the OAP communication to the remote data center fails, the data center servicing the current request can proceed to create a new session locally based on the mandatory attributes available in the cookie. This provides seamless access to the requested resource despite the synchronization failure. [Table 17-2](#) summarizes prominent session synchronization and failover scenarios. The parameters in this table are explained in greater detail in [Table C-2](#).

Table 17-2 Session Synchronization and Failover Scenarios

MDC Deployment	MDC Policy	Validate Remote Session	Session Synchronized in DC Servicing User From Remote DC	Terminate Remote Session	User Challenged
Active-Active	<code>SessionMustBeAnchoredToDataCenterServicingUser=true</code> <code>SessionDataRetrievalOnDemand=true</code> <code>Reauthenticate=false</code> <code>SessionDataRetrievalOnDemandMax_retry_attempts=<number></code> <code>SessionDataRetrievalOnDemandMax_conn_wait_time=<milliseconds></code> <code>SessionContinuationOnSyncFailure= false</code> <code>MDCGitoCookieDomain=<subdomain></code>	Yes	Yes	Yes	When a valid session could not be located in a remote data center

Table 17-2 (Cont.) Session Synchronization and Failover Scenarios

MDC Deployment	MDC Policy	Validate Remote Session	Session Synchronized in DC Servicing User From Remote DC	Terminate Remote Session	User Challenged
Active-Active	SessionMustBeAnchoredToDataCenterServicingUser=false SessionDataRetrievalOnDemand=true Reauthenticate=false SessionDataRetrievalOnDemandMax_retry_attempts=<number> SessionDataRetrievalOnDemandMax_conn_wait_time=<milliseconds> SessionContinuationOnSyncFailure= false MDCGitoCookieDomain=<subdomain>	Yes	Yes	No	When a valid session could not be located in a remote data center
Active-Standby	SessionMustBeAnchoredToDataCenterServicingUser=true SessionDataRetrievalOnDemand=true Reauthenticate=false SessionDataRetrievalOnDemandMax_retry_attempts=<number> SessionDataRetrievalOnDemandMax_conn_wait_time=<milliseconds> SessionContinuationOnSyncFailure= false MDCGitoCookieDomain=<subdomain>	Could not validate as the remote DC is down	No, since the remote DC is down	Could not terminate as the remote data center is down	Yes
Active-Standby	SessionMustBeAnchoredToDataCenterServicingUser=false SessionDataRetrievalOnDemand=true Reauthenticate=false SessionDataRetrievalOnDemandMax_retry_attempts=<number> SessionDataRetrievalOnDemandMax_conn_wait_time=<milliseconds> SessionContinuationOnSyncFailure= true MDCGitoCookieDomain=<subdomain>	Could not validate as the remote data center is down	No, since the remote data center is down	Could not terminate as the remote data center is down	No Provides seamless access by creating a local session from the details available in the valid cookie

17.6 Replicating a Multi-Data Center Environment

Data in the Multi-Data Center environment must be replicated from the Master (supplier) to the Clones (consumers) as part of the initial setup procedure.

Following this initial replication, the following artifacts must be synced across data centers on a regular basis:

- WebGate Profiles: While the WebGate profile is replicated to the Clone, the primary server list and logout URL details are updated with information about the Clone data center.
- Authentication Modules
- OAM Proxy Configurations
- Session Manager configurations
- Policy and partner data



See Also:

- [Replicating Data Using the WLST](#)
- [Syncing Data Using Automated Policy Synchronization](#)
- [Synchronizing Data In A Multi-Data Center](#)

17.6.1 Replicating Data Using the WLST

Initial replication of data (when setting up the Multi-Data Center) must be done manually using the WLST.

Following this initial replication, WLST commands or the Automated Policy Sync Replication Service can be used to sync the already replicated data. When using the WLST, partner profiles and policies are exported from the Master data center and then imported to the Clone data center. Replication of data in a Multi-Data Center environment is a requirement and using WLST for this purpose is the minimum method for accomplishing this.



See Also:

- [About Synchronizing Data Manually in a Multi-Data Center](#)
- [Syncing Data Using Automated Policy Synchronization](#)

17.6.2 Syncing Data Using Automated Policy Synchronization

Automated Policy Synchronization (APS, also referred to as the Replication Service) is a set of REST API used to automatically replicate data from the Master data center to Clone data centers.

It can be configured to keep Access Manager data synchronized across multiple data centers. A valid replication agreement between the data centers must be present before APS can run. See [Understanding the Multi-Data Center Synchronization](#).



Note:

APS is only designed to keep data centers in sync and it is not used to do a complete replication from scratch. You will first need to replicate data manually using the WLST to establish a base line.

17.7 Multi-Data Center Recommendations

This section contains recommendations regarding the Multi-Data Center functionality.

- [Using a Common Domain](#)
- [Concerning the DCC and the OAM_GITO](#)
- [Using an External Load Balancer](#)
- [Honoring Maximum Sessions](#)
- [WebGate Cookie Cannot be Refreshed During Authorization](#)

17.7.1 Using a Common Domain

It is recommended that WebGates be domain-scoped in a manner that a common domain can be inferred across all WebGates and the OAM Server Credential Collectors. This allows for WebGates to set an encrypted GITO cookie to be shared with the OAM Server.

For example, if WebGates are configured on `applications.abc.com` and the OAM Server Credential Collectors on `server.abc.com`, `abc.com` is the common domain used to set the GITO cookie. In scenarios where a common domain cannot be inferred, setting the GITO cookie is not practical as a given data center may not be aware of the latest user sessions in another data center. This would result in the data center computing session idle-timeout based on old session data and could result in re-authenticating the user even though a more active session lives elsewhere.

 **Note:**

A similar issue occurs during server fail-over when the `SessionContinuationOnSyncFailure` property is set. The expectation is to retrieve the session from contents of the `OAM_ID` cookie. Since it's not possible to retrieve the actual inactivity time out value from the GITO cookie, a re-authentication could result.

When there is no common cookie domain across WebGates and OAM servers, make the following configuration changes to address idle time out issues.

- Run the `enableMultiDataCentreMode` WLST command after removing the `MDCGitoCookieDomain` property from the input properties file.
- Because a WebGate cookie cannot be refreshed during authorization, set the value of the WebGate cookie validity lower than the value of the session idle time out property. Consider a session idle time out value of 30 minutes and a WebGate cookie validity value of 15 minutes; in this case, every 15 minutes the session will be refreshed in the authenticating data center.

17.7.2 Concerning the DCC and the OAM_GITO

The `OAM_GITO` cookie is not applicable when DCC is used.

The reasons are:

- The `#MDCGitoCookieDomain=setting` should be commented out.
- The `SessionMustBeAnchoredToDataCenterServicingUser` parameter must be set to `false`.
- The WebGate cookie expiration interval should be set as documented in [Using a Common Domain](#)

17.7.3 Using an External Load Balancer

Access Manager uses the 11g SDK API to retrieve session data but this API does not support SDK based load-balancing across the configured set of primary servers. Use an external TCP based load balancer to front-end the OAP endpoints of the data center nodes where high performance is expected.

 **Note:**

Failover between primary and secondary OAM servers is supported in the current release of 11g SDK APIs.

17.7.4 Honoring Maximum Sessions

A typical Multi-Data Center scenario authenticates users against the data center with which the user geography has an affinity. In the rare scenarios where user authentication and session creation for a given user spans across member data centers (bypassing geographic affinity and load spike), the maximum sessions the user has in the whole Multi-Data Center topology would not be honored.

17.7.5 WebGate Cookie Cannot be Refreshed During Authorization

Set the value of the WebGate cookie validity lower than the value of the session idle time out property as a WebGate cookie cannot be refreshed during authorization.

Consider a session idle time out value of 30 minutes and a WebGate cookie validity value of 15 minutes; in this case, every 15 minutes the session will be refreshed in the authenticating data center. Its recommended to set the WebGate cookie expiration to less than 2 minutes.

18

Configuring Multi-Data Centers

The Multi-Data Center feature is disabled by default. You have to enable and configure the Multi-Data Center functionality.

This section describes the following topics:

- [Before Setting Up a Multi-Data Center](#)
- [Primary Multi-Data Center Use Cases](#)
- [Setting Up a Master and a Clone in Multi-Data Center](#)
- [Adding an Additional Clone Data Center to the Existing Multi-Data Center Setup](#)
- [Multi-Data Center Security Modes](#)

18.1 Before Setting Up a Multi-Data Center

Before you proceed with the Multi-Data Center (MDC) configuration process ensure the system level requirements are met.

- Ensure you have a fully functioning Oracle Access Management environment with all applicable WebGates configured.
- Partners (WebGates or agents) are anchored to a single data center thus, partner registration is done at the individual data centers.
- Clocks on the machines in which Access Manager and agents are deployed must be in sync. Non-MDC Access Manager clusters require the clocks of WebGate agents be in sync with Access Manager servers. This requirement applies to the MDC as well. If the clocks are out of sync, token validations will not be consistent resulting in deviations from the expected behaviors regarding the token expiry interval, validity interval, timeouts and the like.
- The identity stores in a Multi-Data Center topology must have the same name.
- WebLogic Server domains do not span data centers.
- Ensure that the OAM Managed servers in the Master and Clone data centers are front ended by the single (SSL-terminated) load balancer. The load balancer should send all requests in a user session consistently to the same back end server (persistence, stickiness) and it should be route traffic geographically (geo-affinity). Check if this load balancer is configured in the OAM Admin Console of the Master data center before restarting the servers.
- Any firewall between data centers must allow communication over the Oracle Access Protocol (OAP) channel. This entails opening the necessary ports and taking into account the lifetime of the connection. In regards to the latter, the `MaxSessionTime` parameter in the WebGate profile should be set to less than the firewall timeout value.
- OAM Admin server in the Master and Clone data center should be SSL-enabled.
- All the managed servers in the Master and Clone data centers should be configured with the same security mode.

- Use SIMPLE mode to secure communication between OAM Servers and WebGates using out-of-box certificates.
- Use CERT mode, if you have access to a trusted third-party Certificate Authority (CA).
- The ID Stores are configured for Master and Clone data centers and they have the same name.

18.2 Primary Multi-Data Center Use Cases

The primary MDC deployments include active-active and active-standby use cases.

Table 18-1 lists the primary MDC use cases.

Table 18-1 MDC Use Cases

MDC Deployment	MDC Policy	Validate Remote Session	Session Synchronized in data center Servicing User From Remote DC	Terminate Remote Session	User Challenge
Active-Active	SessionMustBeAnchoredToDataCenterServicingUser=false SessionDataRetrievalOnDemand=true Reauthenticate=false SessionDataRetrievalOnDemandMax_retry_attempts=<number> SessionDataRetrievalOnDemandMax_conn_wait_time=<milliseconds> SessionContinuationOnSyncFailure = false MDCGitoCookieDomain=<sub domain>	Yes	Yes	No	When a valid session could not be located in a remote data center
Active-Standby	SessionMustBeAnchoredToDataCenterServicingUser=false SessionDataRetrievalOnDemand=true Reauthenticate=false SessionDataRetrievalOnDemandMax_retry_attempts=<number> SessionDataRetrievalOnDemandMax_conn_wait_time=<milliseconds> SessionContinuationOnSyncFailure = true MDCGitoCookieDomain=<sub domain>	Could not validate as the remote data center is down	No, since the remote data center is down	Could not terminate as the remote data center is down	No Provides seamless access by creating a local session from the details available in the valid cookie

18.3 Setting Up a Master and a Clone in Multi-Data Center

The MDC feature is disabled by default. To set up an Access Manager MDC, start with an Access Manager cluster, set all MDC global configurations and designate the cluster as the Master data center.

Ensure that the Data Center 1 cluster, Data Center 2 cluster and its four nodes are configured and ready for Multi-Data Center configurations. See [Before Setting Up a Multi-Data Center](#).

Note:

To access protected resources for SSO agents created in the data center, both the `PrimaryServerList` and `OAMRestEndPointHostName` must point to the same Data Center, for consistent behavior.

- When an SSO agent for clone data center is created on master data center, the `OAMRestEndPointHostName`, `OAMRestEndPointPort`, and `OAMServerCommunicationMode` parameters must have the clone data center values. This ensures the session gets created on the Clone data center.
- When an SSO agent for master data center is created on master data center, it will automatically have the master values, and therefore no changes are required.

Also see, [About OAP over REST Communication](#) for details.

Setting up Master Data Center

Configure a Master data center for MDC environment using MDC ADMIN REST APIs as follows:

1. Run the following command with appropriate values to configure the Master data center.

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master'
-d
'{"mdcTopologyType":"value", "masterMDCAgentID":"value", "cloneMDCAgentID":
"value",
"accessClientPassword":"value", "artifactPassword":"value",
"cloneServerURL":"value", "cloneServerPolicyManagerURL":"value", "masterServ
erPolicyManagerURL":"value",
"agentKeyPassword":"value", "certModeKeystorePassword":"value", "masterServe
rURL":"value",
"cloneAdminUserNamePassword":"value", "trustStorePath":"value",
"keyStorePath":"value", "artifactsZipLocation":"value"}'
```

- `mdcTopologyType`: Choose one of the two topology types available for MDC configuration, `ACTIVE_ACTIVE` or `DISASTER_RECOVERY`.
- `masterMDCAgentID`: Enter the MDC NAP Agent Name for the Master data center.
- `cloneMDCAgentID`: Enter the MDC NAP Agent Name for the Clone data center.

- `accessClientPassword`: Provide the password required to be used by the MDC NAP agents in Master and Clone data centers.
- `artifactPassword`: Provide the password that is used to protect cloning artifacts.
- `cloneServerURL`: Enter the URL of the Clone Admin server or the URL of the reverse proxy front ending the Clone Admin server.
- `cloneServerPolicyManagerURL`: Enter the URL of the Clone Policy Manager or the URL of the reverse proxy front ending the Clone Policy Manager.
- `masterServerPolicyManagerURL`: Enter the URL of the Master Policy Manager or the URL of the reverse proxy front ending the Master Policy Manager.
- (Only for CERT mode) `agentKeyPassword`: Enter the agent key password used to register partners in the CERT mode.
- (Only for CERT mode) `certModeKeystorePassword`: Enter the keystore password used to protect `clientTrustStore.jks` and `clientKeyStore.jks`.
- (Optional) `masterServerURL`: Enter the URL of the Master Admin server or the URL of the reverse proxy front ending the Master Admin Server.
- (Optional) `cloneAdminUserNamePassword`: Enter the user credentials of the Clone data center's Administrator if the username and password of the Administrator for Master and Clone data centers are different.
- (Optional) `trustStorePath`: Enter the following depending on SIMPLE or CERT mode:
 - For SIMPLE mode : Provide the path to `oamclient-truststore.jks` file if this file is available in folders other than `$MW_HOME/user_projects/domains/OAMDomain/output/webgate-ssl-SHA-256/`
 - For CERT mode : Provide the path to `clientTrustStore.jks` file if this file is available in folders other than `$MW_HOME/user_projects/domains/OAMDomain/config/fmwconfig/oam-mdc-cert-artifacts/`
- (Optional) `keyStorePath`: Enter the following depending on SIMPLE or CERT mode:
 - For SIMPLE mode : Provide the path to `oamclient-keystore.jks` file if this file is available in folders other than `$MW_HOME/user_projects/domains/OAMDomain/output/webgate-ssl-SHA-256/`
 - For CERT mode : Provide the path to `clientKeyStore.jks` file if this file is available in folder other than `$MW_HOME/user_projects/domains/OAMDomain/config/fmwconfig/oam-mdc-cert-artifacts/`
- (Optional) `artifactsZipLocation`: Provide the location where cloning artifacts has to be stored; specify only if cloning artifacts need to be stored in any location other than `/tmp`

Here are the sample Curl commands for configuring a Master data center in SIMPLE and CERT modes using Active-Active MDC topology:

- Using CERT mode:

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/
master'
-d '{"mdcTopologyType":"ACTIVE_ACTIVE",
"masterMDCAgentID":"MDCmasterNAPagent", "cloneMDCAgentID":"MDCcloneNAPa
gent",
"accessClientPassword":"password", "artifactPassword":"password",
"cloneServerURL":"https://oamadmin1-
dc2.poc.com:7002", "cloneServerPolicyManagerURL":"https://oamadmin1-
dc2.poc.com:14151", "masterServerPolicyManagerURL":"https://oamadmin1-
dc1.poc.com:14151",
"cloneAdminUserNamePassword":"weblogic:password", "agentKeyPassword":"p
assword", "certModeKeystorePassword":"password"}
```

- Using SIMPLE mode:

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/
master'
-d '{"mdcTopologyType":"ACTIVE_ACTIVE",
"masterMDCAgentID":"MDCmasterNAPagent", "cloneMDCAgentID":"MDCcloneNAPa
gent",
"accessClientPassword":"password", "artifactPassword":"password",
"cloneServerURL":"https://oamadmin1-
dc2.poc.com:7002", "cloneServerPolicyManagerURL":"https://oamadmin1-
dc2.poc.com:14151", "masterServerPolicyManagerURL":"https://oamadmin1-
dc1.poc.com:14151",
"cloneAdminUserNamePassword":"weblogic:password"}
```

See MDC Master REST API in *REST API for Multi Data Center in Oracle Access Manager*.

Setting up Clone Data Center

Configure a Clone data center for MDC environment using MDC ADMIN REST APIs as follows:

1. Run the following command with appropriate values to configure the Clone data center.

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone'
-d
'{"masterServerURL":"value", "artifactPassword":"value", "masterAdminUserNam
ePassword":"value",
"masterServerPolicyManagerURL":"value", "artifactsZipLocation":"value",
"masterArtifactsZipLocation":"value"}
```

- `masterServerURL`: Enter the URL of the Master Admin server or the URL of the reverse proxy front ending the Master Admin Server.
- `masterServerPolicyManagerURL`: Enter the URL of the Master Policy Manager or the URL of the reverse proxy front ending the Master Policy Manager.

- `artifactPassword`: Provide the same password that protects cloning artifacts and used while setting up the Master data center
- (Optional) `masterAdminUserNamePassword`: Enter the user credentials of the Master data center's Administrator if the username and password of the Administrator for Master and Clone data centers are different.
- (Optional) `artifactsZipLocation`: Provide the location where backup artifacts should be stored in Clone data center (artifacts present in Clone data center are backed up before replacing it with Master artifacts); specify only when the backup artifacts need to be stored in any location other than `/tmp`.
- (Optional) `masterArtifactsZipLocation`: Provide the location where cloning artifacts are present in Master data center; specify only when `artifactsZipLocation` was used in input while configuring the Master data center.

Here is the sample Curl command for configuring a Clone data center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/
clone'
-d '{"masterServerURL":"https://oamadmin1-dc1.poc.com:7002/",
"masterServerPolicyManagerURL":"https://oamadmin1-
dc1.poc.com:14151",
"artifactPassword":"password","masterAdminUserNamePassword":"password"}'
```

See MDC Clone REST API in *REST API for Multi Data Center in Oracle Access Manager*.

2. Run the following command to reconfigure the Clone Data Center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X
POST 'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/
clone/configuration'
```

 **Note:**

This command does not require any input parameters. It updates the flag, `DataCenterType` to `Clone`. To make the clone write-protected, execute the WLST command `setMultiDataCenterWrite(WriteEnabledFlag="false")`. It ignores any update to clone configuration.

See MDC Reconfigure Clone REST API in *REST API for Multi Data Center in Oracle Access Manager*.

You have successfully setup, one master and one clone data center.

**See Also:**[Adding an Additional Clone Data Center to the Existing Multi-Data Center Setup](#)

18.4 Adding an Additional Clone Data Center to the Existing Multi-Data Center Setup

You can add an additional clone data center to the existing MDC environment if the Master and Clone data centers are using 12.2.1.3.0 binaries.

**Note:**

Upgrading Oracle Access Management Multi-Data Center Environments

1. Optionally, You can run the diagnostic REST APIs on the Master and the Clone Data Centers to view the MDC configuration settings:

```
curl -k -u weblogic:password 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/configuration'
curl -k -u weblogic:password 'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/configuration'
```

Verify the following from the output of the command:

- When the diagnostic REST API is executed on the Master,
In `dcConfigMap` entry, `MultiDataCenterEnabled` should be `true`,
`MultiDataCenterPartners` should contain the existing MDC Partners and `agentMap` entry should contain the information about agents associated with the MDCPartners.
- When the diagnostic REST API is executed on the Clone,
In `dcConfigMap` entry, `MultiDataCenterEnabled` should be `false`,
`MultiDataCenterPartners` list should be empty and `agentMap` entry should be empty.

See MDC Diagnostic REST API in *REST API for Multi Data Center in Oracle Access Manager*.

2. Run the following command with appropriate values to add a new clone to the Master data center.

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master/clone' -d '{ "cloneMDCAgentID": "value", "accessClientPassword": "value", "artifactPassword": "value", "cloneServerURL": "value", "agentKeyPassword": "value", "certModeKeystorePassword": "value", "cloneAdminUserNamePassword": "value", "trustStorePath": "value", "keyStorePath": "value", "artifactsZipLocation": "value" }'
```


- `cloneMDCAgentID`: Enter the MDC NAP Agent Name for the new Clone data center.
- `accessClientPassword`: Provide the password required to use the MDC NAP agents in the new Clone data centers.
- `artifactPassword`: Provide the password that is used to protect cloning artifacts.
- `cloneServerURL`: Enter the URL of the new Clone Admin server or the URL of the reverse proxy front ending the new Clone Admin server.
- (Only for CERT mode) `agentKeyPassword`: Enter the agent key password used to register the new Clone partners in the CERT mode.
- (Only for CERT mode) `certModeKeystorePassword`: Enter the keystore password used to protect `clientTrustStore.jks` and `clientKeyStore.jks`.
- (Optional) `cloneAdminUserNamePassword`: Enter the user credentials of the new Clone data center's Administrator if the username and password of the Administrator for Master and new Clone data centers are different.
- (Optional) `trustStorePath`: Enter the following depending on SIMPLE or CERT mode:
 - For SIMPLE mode, Provide the path to `oamclient-truststore.jks` file if this file is available in folder other than `$MW_HOME/user_projects/domains/OAMDomain/output/webgate-ssl-SHA-256/`.
 - For CERT mode, Provide the path to `clientTrustStore.jks` file if this file is available in folder other than `$MW_HOME/user_projects/domains/OAMDomain/config/fmwconfig/oam-mdc-cert-artifacts/`.
- (Optional) `keyStorePath`: Enter the following depending on SIMPLE or CERT mode:
 - For SIMPLE mode, Provide the path to `oamclient-keystore.jks` file if this file is available in folder other than `DOMAIN_HOME/output/webgate-ssl-SHA-256/`.
 - For CERT mode, Provide the path to `clientKeyStore.jks` file if this file is available in folder other than `$MW_HOME/user_projects/domains/OAMDomain/config/fmwconfig/oam-mdc-cert-artifacts/`.
- (Optional) `artifactsZipLocation`: Provide the location where cloning artifacts has to be stored; specify only if cloning artifacts need to be stored in any location other than `/tmp`

Here are the sample Curl commands for configuring Managed Servers in SIMPLE and CERT modes:

- Using CERT mode:

```
curl -u weblogic:password -H 'Content-Type: application/json' -X
POST 'https://oamadmin1-dcl.poc.com:7002/oam/services/rest/mdc/
master/clone' -d
'{"cloneMDCAgentID":"CloneNAPAgent2","accessClientPassword":"pass
word","artifactPassword":"password","cloneServerURL":"https://
oamadmin1-
```

```
dc2.poc.com:7002", "agentKeyPassword": "password", "certModeKeystorePassw
ord": "password"}'
```

- Using SIMPLE mode:

```
curl -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master/
clone' -d
'{"cloneMDCAgentID": "CloneNAPAgent2", "accessClientPassword": "password"
, "artifactPassword": "password", "cloneServerURL": "https://oamadmin1-
dc2.poc.com:7002"}'
```

See MDC Clone REST API in *REST API for Multi Data Center in Oracle Access Manager*.

3. Run the following command with appropriate values to configure the Clone data center.

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone' -d
'{"masterServerURL": "value", "artifactPassword": "value", "masterAdminUserNam
ePassword": "value", "artifactsZipLocation": "value",
"masterArtifactsZipLocation": "value"}'
```

- `masterServerURL`: Enter the URL of the Master Admin server or the URL of the reverse proxy front ending the Master Admin Server.
- `artifactPassword`: Provide the same password that protects cloning artifacts and used while setting up the Master data center
- (Optional) `masterAdminUserNamePassword`: Enter the user credentials of the Master data center's Administrator if the username and password of the Administrator for Master and Clone data centers are different.
- (Optional) `artifactsZipLocation`: Provide the location where backup artifacts should be stored in Clone data center (artifacts present in Clone data center will be backed up before replacing it with Master artifacts); Specify only when the backup artifacts need to be stored in any location other than `/tmp`.
- (Optional) `masterArtifactsZipLocation`: Provide the location where cloning artifacts are present in Master data center; specify only when `artifactsZipLocation` was used in input while configuring the Master data center.

Here is the sample Curl command for configuring a Clone data center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone' -d
'{"masterServerURL": "https://oamadmin1-
dc1.poc.com:7002/", "artifactPassword": "password", "masterAdminUserNamePassw
ord": "password"}'
```

4. Run the following command to reconfigure the Clone Data Center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
' https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone/
configuration'
```

 **Note:**

This command does not require any input parameters. It updates the flag, `DataCenterType` to `Clone`. To make the clone write-protected, execute the WLST command `setMultiDataCenterWrite(WriteEnabledFlag="false")`. It ignores any update to clone configuration.

See MDC Reconfigure Clone REST API in *REST API for Multi Data Center in Oracle Access Manager*.

5. Restart Clone Admin and managed servers.
6. Run the following diagnostic REST API on the Master and the Clone Data Centers to verify MDC configurations:

```
curl -k -u weblogic:password 'https://oamadmin1-
dc1.poc.com:7002/oam/services/rest/mdc/configuration'
curl -k -u weblogic:password 'https://oamadmin1-
dc2.poc.com:7002/oam/services/rest/mdc/configuration'
```

7. Export the partner and policy information from Data Center 1, Node 1 and then import it to Data Center 2, Node 1.
 - a. To export, change to the `$MW_HOME/oracle_common/common/bin` directory and run WLST to export from Data Center 1, Node 1.

```
./wlst.sh
connect()
exportAccessStore(toFile="<name and location of the master
metadata ZIP file>", namePath="/")
exit()
```

- b. Copy the exported file (that is, *<name and location of the master metadata ZIP file>*) from Data Center 1, Node 1 to Data Center 2, Node 1. To import, change to the `$MW_HOME/oracle_common/common/bin` directory and run WLST to import on Data Center 2, Node 1.

```
./wlst.sh
connect()
importAccessStore(fromFile="<name and location of master
metadata ZIP file>", namePath="/")
exit()
```

After exporting the partner and policy information from Master data center to Clone data center continue with enabling APS steps as specified in [Enabling Automated Policy Synchronization](#).

18.5 Multi-Data Center Security Modes

A Multi-Data Center relies on the Oracle Access Protocol (OAP) channel for inter data center session management operations and back channel communication. The

security mode of the MDC partner profile should match the security mode defined for the Access Manager server: OPEN, SIMPLE or CERT.

The following sections have details about the security modes.

- [OPEN Security Mode](#)
- [SIMPLE Security Mode](#)
- [CERT Security Mode](#)

18.5.1 OPEN Security Mode

OPEN Security Mode is the default mode of the Access Manager deployment. No configuration is needed. Oracle recommends SIMPLE or CERT modes to avoid security issues. In a multi-data center environment, all the Admin and managed servers need to be configured with the same security mode.

Use MDC Admin REST commands to setup the master data center in OPEN mode and provide the following mandatory and optional MDC parameters as shown in the example:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d
'{"mdcTopologyType":"value", "masterMDCAgentID":"value", "cloneMDCAgentID":"va
lue", "accessClientPassword":"value", "artifactPassword":"value", "cloneServerU
RL":"value", "agentKeyPassword":"value", "certModeKeystorePassword":"value", "ma
sterServerURL":"value", "cloneAdminUserNamePassword":"value",
"artifactsZipLocation":"value"}'
```

- `mdcTopologyType`: Choose one of the two topology types available for MDC configuration, `ACTIVE_ACTIVE` or `DISASTER_RECOVERY`.
- `masterMDCAgentID`: Enter the MDC NAP Agent Name for the Master data center.
- `cloneMDCAgentID`: Enter the MDC NAP Agent Name for the Clone data center.
- `accessClientPassword`: Provide the password required to use the MDC NAP agents in Master and Clone data centers.
- `artifactPassword`: Provide the password that is used to protect cloning artifacts.
- `cloneServerURL`: Enter the URL of the Clone Admin server or the URL of the reverse proxy front ending the Clone Admin server.
- (Optional) `masterServerURL`: Enter the URL of the Master Admin server or the URL of the reverse proxy front ending the Master Admin Server.
- (Optional) `cloneAdminUserNamePassword`: Enter the user credentials of the Clone data center's Administrator if the username and password of the Administrator for Master and Clone data centers are different.
- (Optional) `artifactsZipLocation`: Provide the location where cloning artifacts has to be stored; specify only if cloning artifacts need to be stored in any location other than `/tmp`

Here are the sample Curl commands for configuring a Master data center in OPEN mode using Active-Active and Disaster_Recovery MDC topologies:

- Using Active-Active MDC topology:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d '{"mdcTopologyType":"ACTIVE_ACTIVE", "masterMDCAgentID":"MDCmasterNAPagent", "cloneMDCAgentID":"MDCcloneNAPagent", "accessClientPassword":"password", "artifactPassword":"password", "cloneServerURL":"https://oamadmin1-dc2.poc.com:7002", "cloneAdminUserNamePassword":"weblogic:password"}'
```

- Using Disaster Recovery MDC topology:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d '{"mdcTopologyType":"DISASTER_RECOVERY", "masterMDCAgentID":"MDCmasterNAPagent", "cloneMDCAgentID":"MDCcloneNAPagent", "accessClientPassword":"password", "artifactPassword":"password", "cloneServerURL":"https://oamadmin1-dc2.poc.com:7002", "cloneAdminUserNamePassword":"weblogic:password"}'
```



See Also:

[Setting Up a Multi-Data Center](#)

[Enabling Automated Policy Synchronization](#)

18.5.2 SIMPLE Security Mode

Follow the instructions in [Configuring Simple Mode Communication with Access Manager](#) to set up the Access Manager servers in SIMPLE mode. In a multi-data center environment, all the Admin and managed servers need to be configured with the same security mode. Use SIMPLE mode to secure communications between OAM Servers and WebGates using out-of-box certificates.

Use MDC Admin REST commands to setup the master data center in SIMPLE mode and provide the following mandatory and optional MDC parameters as shown in the example:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d '{"mdcTopologyType":"value", "masterMDCAgentID":"value", "cloneMDCAgentID":"value", "accessClientPassword":"value", "artifactPassword":"value", "cloneServerURL":"value", "agentKeyPassword":"value", "certModeKeystorePassword":"value", "masterServerURL":"value", "cloneAdminUserNamePassword":"value", "trustStorePath":"value", "keyStorePath":"value", "artifactsZipLocation":"value"}'
```

- `mdcTopologyType`: Choose one of the two topology types available for MDC configuration, `ACTIVE_ACTIVE` or `DISASTER_RECOVERY`.

- `masterMDCAgentID`: Enter the MDC NAP Agent Name for the Master data center.
- `cloneMDCAgentID`: Enter the MDC NAP Agent Name for the Clone data center.
- `accessClientPassword`: Provide the password required to use the MDC NAP agents in Master and Clone data centers.
- `artifactPassword`: Provide the password that is used to protect cloning artifacts.
- `cloneServerURL`: Enter the URL of the Clone Admin server or the URL of the reverse proxy front ending the Clone Admin server.
- (Optional) `masterServerURL`: Enter the URL of the Master Admin server or the URL of the reverse proxy front ending the Master Admin Server.
- (Optional) `cloneAdminUserNamePassword`: Enter the user credentials of the Clone data center's Administrator if the username and password of the Administrator for Master and Clone data centers are different.
- (Optional) `trustStorePath`: Provide the path to `oamclient-truststore.jks` file if this file is available in folders other than `%DOMAIN_HOME%/output/webgate-ssl-SHA-256`
- (Optional) `keyStorePath`: Provide the path to `oamclient-keystore.jks` file if this file is available in folders other than `%DOMAIN_HOME%/output/webgate-ssl-SHA-256`
- (Optional) `artifactsZipLocation`: Provide the location where cloning artifacts has to be stored; specify only if cloning artifacts need to be stored in any location other than `/tmp`

Here are the sample `curl` commands for configuring a Master data center in SIMPLE mode using Active-Active and Disaster_Recovery MDC topologies:

- Using Active-Active MDC topology:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d
'{"mdcTopologyType":"ACTIVE_ACTIVE",
"masterMDCAgentID":"MDCmasterNAPagent", "cloneMDCAgentID":"MDCcloneNAPagent",
"accessClientPassword":"password", "artifactPassword":"password", "cloneServerURL":"https://oamadmin1-dc2.poc.com:7002", "cloneAdminUserNamePassword":"weblogic:password"}
```

- Using Disaster Recovery MDC topology:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d
'{"mdcTopologyType":"DISASTER_RECOVERY",
"masterMDCAgentID":"MDCmasterNAPagent", "cloneMDCAgentID":"MDCcloneNAPagent",
"accessClientPassword":"password", "artifactPassword":"password", "cloneServerURL":"https://oamadmin1-dc2.poc.com:7002", "cloneAdminUserNamePassword":"weblogic:password"}
```

**See Also:**[Setting Up a Multi-Data Center](#)[Enabling Automated Policy Synchronization](#)

18.5.3 CERT Security Mode

Follow the instructions in [Configuring Cert Mode Communication for Access Manager](#) to set up the Access Manager servers in CERT mode. In a multi-data center environment, all the Admin and managed servers need to be configured with the same security mode. Use CERT mode if you have access to a trusted third-party Certificate Authority (CA).

Create an MDC partner in each of the member data centers in CERT mode. Generate the `clientTrustStore.jks` and `clientKeyStore.jks` KeyStores for the partners to communicate in CERT mode.

In an MDC setup, each Clone data center is a replica of the Master data center. For the newly cloned data centers to communicate with the existing data centers in CERT mode, the KeyStores generated may be reused across data centers. However, while configuring the domain across multiple nodes (such as adding a new OAM server to a new host), ensure that the new host's file system has the required artifacts stored in the same directory structure as that of the AdminServer node.

1. Run the following `openssl` command from a Linux command prompt to generate `aaa_key.pem` and `aaa_req.pem`.

```
openssl req -new -keyout aaa_key.pem -out aaa_req.pem -utf8 -sha256
```

Use the `certreq` command to generate the certificate.

2. Create `aaa_cert.pem` using the following procedure.
 - a. Open `aaa_req.pem` in a text editor and copy the contents.
Exclude the trailing spaces from your selection.
 - b. Paste the copied text into `Signcsr`.
Include `[-----BEGIN CERTIFICATE REQUEST-----` and `-----END CERTIFICATE REQUEST-----]`.
 - c. Copy the output into a text editor and save it as `aaa_cert.pem`.
3. Create `aaa_chain` using the following procedure.
 - a. Open `certreq`.
 - b. Click on `chain.pem` and copy/paste the contents into a text editor and save it as `aaa_chain.pem`.
Excluding trailing and leading spaces from your selection.
4. Encrypt the private key (`aaa_key.pem`) using the following command.

```
openssl rsa -in aaa_key.pem -passin pass: -out aaa_key.pem -passout  
pass:Welcome1 -des
```

The password used in this command must be defined as the access client password or agent key password while registering the MDC partner.

5. Copy `aaa_key.pem`, `aaa_cert.pem`, and `aaa_chain.pem` to a temporary location.

For example, `/tmp/clientCertArtifacts/`

6. Convert `aaa_cert.pem` and `aaa_key.pem` into DER format using one of the following commands.

```
-openssl x509 -in /tmp/clientCertArtifacts/aaa_cert.pem -inform PEM -out
/tmp/clientCertArtifacts/aaa_cert.der -outform DER;
```

```
-openssl pkcs8 -topk8 -nocrypt -in /tmp/clientCertArtifacts/aaa_key.pem
-inform PEM -out /tmp/clientCertArtifacts/aaa_key.der -outform DER;
```

7. Import the `aaa_key.der` and `aaa_cert.der` into `clientKeyStore.jks`; and the `aaa_chain.pem` into `clientTrustStore.jks` with the below steps

```
-cd $MW_HOME/idm/oam/server/tools/importcert/;
```

```
-unzip importcert.zip;
```

```
-java -cp importcert.jar
oracle.security.am.common.tools.importcerts.CertificateImport -keystore
/tmp/clientCertArtifacts/clientKeyStore.jks -privatekeyfile
/tmp/clientCertArtifacts/aaa_key.der -signedcertfile
/tmp/clientCertArtifacts/aaa_cert.der -storetype jks -genkeystore yes
```

```
-keytool -importcert -file /tmp/clientCertArtifacts/aaa_chain.pem -trustcacerts
-keystore /tmp/clientCertArtifacts/clientTrustStore.jks -storetype JKS
```

Enter the keystore passwords when prompted. The password needs to be set in the input parameter, `certModeKeystorePassword` while setting up Master data center.

If not done when creating the certificates for the WebGate, import the `aaa_key.der` and `aaa_cert.der` formatted certificates into the `.oamkeystore` using the same Oracle provided `importcert.jar` used in the previous step.

```
-java -cp importcert.jar
oracle.security.am.common.tools.importcerts.CertificateImport
-keystore /scratch/Oracle/Middleware/domains/
base_domain/config/fmwconfig/.oamkeystore -privatekeyfile
/tmp/clientCertArtifacts/aaa_key.der -signedcertfile
/tmp/clientCertArtifacts/aaa_cert.der -alias mycertmodel -storetype JCEKS
```

alias is the alias name defined when setting CERT mode in Access Manager

Use MDC Admin REST commands to setup the master data center in CERT mode and provide the following mandatory and optional MDC parameters as shown in the example:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d
'{"mdcTopologyType":"value", "masterMDCAgentID":"value", "cloneMDCAgentID":"va
lue", "accessClientPassword":"value", "artifactPassword":"value", "cloneServerU
RL":"value", "agentKeyPassword":"value", "certModeKeystorePassword":"value", "ma
sterServerURL":"value",
"cloneAdminUserNamePassword":"value", "trustStorePath":"value",
"keyStorePath":"value", "artifactsZipLocation":"value"}'
```


- `mdcTopologyType`: Choose one of the two topology types available for MDC configuration, `ACTIVE_ACTIVE` or `DISASTER_RECOVERY`.
- `masterMDCAgentID`: Enter the MDC NAP Agent Name for the Master data center.
- `cloneMDCAgentID`: Enter the MDC NAP Agent Name for the Clone data center.
- `accessClientPassword`: Provide the password required to use the MDC NAP agents in Master and Clone data centers.
- `artifactPassword`: Provide the password that is used to protect cloning artifacts.
- `cloneServerURL`: Enter the URL of the Clone Admin server or the URL of the reverse proxy front ending the Clone Admin server.
- (Only for CERT mode) `agentKeyPassword`: Enter the agent key password used to register partners in the CERT mode.
- (Optional) `masterServerURL`: Enter the URL of the Master Admin server or the URL of the reverse proxy front ending the Master Admin Server.
- (Optional) `cloneAdminUserNamePassword`: Enter the user credentials of the Clone data center's Administrator if the username and password of the Administrator for Master and Clone data centers are different.
- (Optional) `trustStorePath`: Provide the path to `clientTrustStore.jks` file if this file is available in folders other than `$MW_HOME/user_projects/domains/OAMDomain/config/fmwconfig/oam-mdc-cert-artifacts/`
- (Optional) `keyStorePath`: Provide the path to `clientKeyStore.jks` file if this file is available in folders other than `$MW_HOME/user_projects/domains/OAMDomain/config/fmwconfig/oam-mdc-cert-artifacts/`
- (Optional) `artifactsZipLocation`: Provide the location where cloning artifacts has to be stored; specify only if cloning artifacts need to be stored in any location other than `/tmp`

Here are the sample `curl` commands for configuring a Master data center in CERT mode using Active-Active and Disaster_Recovery MDC topologies:

- Using Active-Active MDC topology:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d '{"mdcTopologyType":"ACTIVE_ACTIVE", "masterMDCAgentID":"MDCmasterNAPagent", "cloneMDCAgentID":"MDCcloneNAPagent", "accessClientPassword":"password", "artifactPassword":"password", "cloneServerURL":"https://oamadmin1-dc2.poc.com:7002", "cloneAdminUserNamePassword":"weblogic:password", "agentKeyPassword":"password", "certModeKeystorePassword":"password"}'
```

- Using Disaster Recovery MDC topology:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -d '{"mdcTopologyType":"DISASTER_RECOVERY",
```

```
"masterMDCAgentID":"MDCmasterNAPagent","cloneMDCAgentID":"MDCcloneNAPagent",  
",  
"accessClientPassword":"password","artifactPassword":"password","cloneServerURL":"https://oamadmin1-  
dc2.poc.com:7002","cloneAdminUserNamePassword":"weblogic:password","agentKeyPassword":"password", "certModeKeystorePassword":"password"}'
```



See Also:

[Setting Up a Multi-Data Center](#)

[Enabling Automated Policy Synchronization](#)

19

Synchronizing Data In A Multi-Data Center

The Multi-Data Center infrastructure can be configured to keep Access Manager data synchronized across multiple data centers. This can be done using the Automated Policy Synchronization Replication Service or data can be replicated manually.

The following topics describe how to synchronize across data centers and replicate data:

- [Understanding the Multi-Data Center Synchronization](#)
- [Enabling Data Replication](#)
- [Synchronizing Master and Clone Metadata](#)
- [Using REST API for Replication Agreements](#)
- [Customizing Transformation Rules](#)
- [Disabling Automated Policy Synchronization](#)
- [Best Practices for Replication](#)

19.1 Understanding the Multi-Data Center Synchronization

The Multi-Data Center (MDC) infrastructure can be configured to keep Access Manager data synchronized across multiple data centers.

Policy, system configuration, and partner metadata are all synchronized with APS.

Note:

For creating a Clone (also referred to as a Consumer) from a Master (also referred to as a Supplier), MDC Admin REST APIs are used. Once the MDC infrastructure is deployed, APS can be enabled to automatically sync any changes from the Master to the Clones.

See [Before Setting Up a Multi-Data Center](#).

APS (also referred to as the Replication Service) is a set of REST API. The binaries are installed as part of the Access Manager application and deployed in the AdminServer. It is enabled by default. After enabling the service, create a pull model Replication Agreement between the Clone data center and the Master. The Clone polls for changes from the Master as long as the Replication Agreement is valid for it. Conversely, the Master will respond to the Clone's request as long as it finds a valid replication agreement. The Clone applies the changes locally.

 **Note:**

APS is optional; administrator-initiated import and export based replication is still available using WLST command procedure used previous to this release.

When setting up the Replication Service, the following may occur:

- Replication Agreement is established. One data center is registered as a Replication Clone and another one is registered in a separate geographical location as its Master. From the Master data center, the changes are pulled and applied to the Clone data center.
- Data center-specific configurations that may not be replicated across data centers are defined.
- Access Manager configuration changes are tracked in each data center; the current replication state can be queried in any of the data centers.
- Change Log is generated; it can be applied in the context of a similar setup running in another data center.
- A pull from the Master data center can be triggered, if required; for example, if there is a failure of automated replication.
- Access Manager configuration artifacts in the Master-Clone model are replicated.

 **Note:**

APS does not sync IDS Profiles, OAM Keystores, and Oracle Platform Security Services artifacts (jps-config.xml changes, credential store configuration and the like).

This section describes the following topics:

- [How Replication Works](#)
- [Understanding the Replication Agreement](#)
- [About Synchronizing Data Manually in a Multi-Data Center](#)

19.1.1 How Replication Works

Replication works in a Master-Clone topology. In this topology, multiple Clones pull changes from a single Master. One data center is defined by the administrator as the Master and one or more other data centers are Clones.

The administrator makes changes to the Master that are replicated to the Clones. Only Master to Clone replication is supported; changes to Clones are not replicated back to the Master.



Note:

Multi-master replication is not supported.

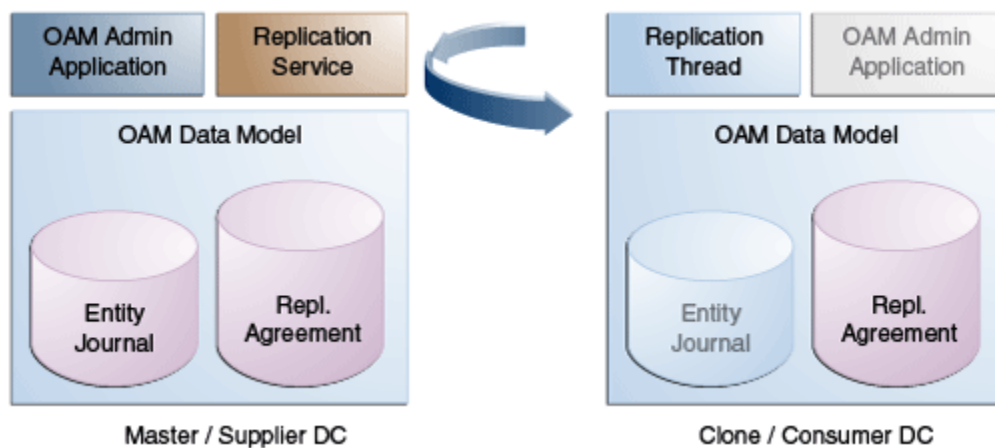
To partake in replication, the Master data center (initiator of the replication) and the Clone data center (receiver of the changes) must have a Replication Agreement stored in the Access Manager data store. [Table 19-1](#) documents the states in which replication can be deployed.

Table 19-1 Replication States

State	Definition
Active	An Access Manager domain (including Admin and managed servers) is setup to serve access requests. In an active state, the Access Manager server provides the web access management functionality without additional MDC features.
Bootstrapping	This state is optional for some Data Centers; for example, the first one in an MDC topology. A Data Center goes through this intermediate state when added to an existing MDC topology. The new DC contacts the master and bootstraps itself to the same state. The bootstrap includes synchronizing the server keys, policy artifacts, partners, and system configuration. After completion of bootstrapping, the DC will be Replication Ready.
Replication Ready	In this state, MDC is enabled, the DC is made part of the topology, and the replication service is enabled. Once enabled, a clone can be registered with the master via a Replication Agreement. Once established, clone DCs can query and start pulling changelogs from the master.

[Figure 19-1](#) illustrates the replication flow.

Figure 19-1 Replication Flow



Each Clone pulls changes from the Master. A replication thread runs on the Clone after the pre-configured interval of time and fetches changes from the Replication Service (REST endpoint) running on the Master environment.

For every cloned environment, the Master keeps track of a change sequence number indicating when it was last synced. The Master also keeps track of the list of changes (Create/Update/Delete) that have been pulled by the Clones in a changelog using specific change sequence numbers. When updating configuration metadata, the Clone can also change environment specific parameter values depending on transformation rules.

See [Customizing Transformation Rules](#).

19.1.2 Understanding the Replication Agreement

Configuration changes (defined as *journals*) are replicated from a Master node to Clone nodes. On receiving the journals, each node updates its configuration to match the journal and remain in a synchronized state. The nodes, though, need to enter a Replication Agreement to receive change journals.

When a new data center is added to an existing MDC topology, it has to bootstrap itself to be in sync with the existing data centers. This bootstrap operation will get the current Access Manager policies, system configuration, partner metadata, and server keys for the existing MDC topology. After the bootstrap operation, the new data center captures the last change sequence number from the topology's Master so that during replication it can be used to determine the current state.



Note:

Automated bootstrap is the ideal scenario but you can execute diagnostic MDC REST APIs first to ensure the Master and Clone data centers are in the same state. If the data centers are not in same state, then you have to execute MDC ADMIN REST APIs.

To establish a Replication Agreement, the Clone data center must know the Master's changelog sequence number. If the data center is added to the topology on 'day-0' and the Replication Agreement was created on 'day-1', there is a need to bootstrap again. To avoid this and to keep the flow simple, creating a Replication Agreement should take care of the bootstrap and actual replication agreement creation.

See [Setting Up a Master and a Clone in Multi-Data Center](#).

19.1.3 About Synchronizing Data Manually in a Multi-Data Center

Data in an MDC topology can also be synced manually. The manual option uses WLST export and import calls to migrate the data from one data center to another. Partner profiles and policies should both be exported and imported using WLST.

See Access Manager WLST Commands in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference for Identity and Access Management*.

19.2 Enabling Data Replication

Data Replication on the Master and Clone data centers is enabled by default. Setting the flag `-Doracle.oam.EnableMDCReplication` to `true` is not mandatory.

Validate the Replication REST endpoints in Master and Clone data centers as follows:

1. Validate that the REST endpoints are enabled in the Master data center.

```
curl -u weblogic:password 'https://supplier.example.com:7002/oam/services/  
rest/_replication/hello
```

2. Validate that the REST endpoints are enabled in the Clone data center.

```
curl -u weblogic:password 'https://supplier.example.com:7002/oam/services/  
rest/_replication/hello
```

3. Repeat the process of validation on all Clone data centers.

19.3 Synchronizing Master and Clone Metadata

The process for syncing metadata across an MDC involves first syncing Access Manager UDM metadata and then creating a replication agreement.

See [Understanding the Replication Agreement](#).

The following topics describe how to synchronize master and clone metadata:

- [Synchronizing the UDM Metadata](#)
- [Creating a Replication Agreement](#)
- [Modifying a Replication Agreement](#)

19.3.1 Synchronizing the UDM Metadata

You must synchronize the UDM Metadata before you can create the replication agreement.

To sync Access Manager UDM metadata stored in the Master to all Clones:

1. Execute the `exportAccessStore WLST` command on the Master Data Center to create a ZIP file containing the UDM metadata.

```
exportAccessStore(toFile="/master/location/dclmetadata.zip",  
namePath="/")
```

2. Copy `dclmetadata.zip` to the Clone DC location.

3. Execute the `importAccessStore WLST` command on the Clone Data Center to import the UDM metadata.

```
importAccessStore(fromFile="/clone/location/dclmetadata.zip",  
namePath="/")
```

4. Repeat on all Clone DCs.

19.4 Using REST API for Replication Agreements

Use REST APIs provided by Access Manager to manage replication agreements.

This section describes the following topics:

- [Querying for Replication Agreement Details](#)
- [Creating a Replication Agreement](#)
- [Modifying a Replication Agreement](#)

- [Deleting a Replication Agreement](#)

19.4.1 Querying for Replication Agreement Details

Execute a REST request at the Master's endpoint to query the details of the Replication Agreement (including the polling interval) between a Master and a Clone data center.

1. If the replication agreement identifier is unknown, run the following command to list all the replication agreement identifiers, else skip this step.

```
curl -k -u weblogic:password 'https://supplier.example.com:7002/oam/services/rest/_replication/agreements'
```

Sample output 1:

```
{"featureEnabled":"true","identifiers":"201411211137273612","ok":"true"}
```

Sample output 2:

```
{"featureEnabled":"true","identifiers":["201411211137273612","201411211137273900"],"ok":"true"}
```

2. Query the details of a Clone data center's replication agreement (including the polling interval) as follows:

```
curl -k -u weblogic:password 'https://supplier.example.com:7002/oam/services/rest/_replication/201409231329353668?type=consumer'
```

Sample output:

```
{"enabled":"true","identifier":"201409231329353668","ok":"true",  
"pollInterval":"60","startingSequenceNumber":"110","state":"READY"}
```

3. Query the details of a Master data center's replication agreement (including the polling interval) as follows:

```
curl -u weblogic:password 'https://supplier.example.com:7002/oam/services/rest/_replication/201409231329353668'
```

Sample output:

```
{"enabled":"true","identifier":"201409231329353668","lastSequenceNumber":"110","ok":"true","pollInterval":"3600","startingSequenceNumber":"110","state":"ACTIVE"}
```

 **Note:**

The poll interval of the Master data center's replication agreement does not affect the actual replication.

19.4.2 Creating a Replication Agreement

Creating a Replication Agreement is a one time operation which will enable the Clone data center(s) to pull changes from the Master data center.

The replication agreement can be created using any REST client. In this procedure, we use the standard Curl utility.

After you execute this command, the following results occur:

- Insert an entry in the Master's Replication Agreement store containing details regarding the Clone that wants to pull changes.
- Insert an entry in the Clone's Replication Agreement store containing details regarding the Master from which it will pull changes. Replication configuration values like the poll interval will also be set.

To create a replication agreement:

1. Ensure the Master and Clone DC REST endpoints are up and running.
2. Execute the following command on the Master DC.

This command will use the `repluser` specified for replication queries from the Master to the Clone. `repluser` is expected to be available in the default identity stores for all involved DCs.

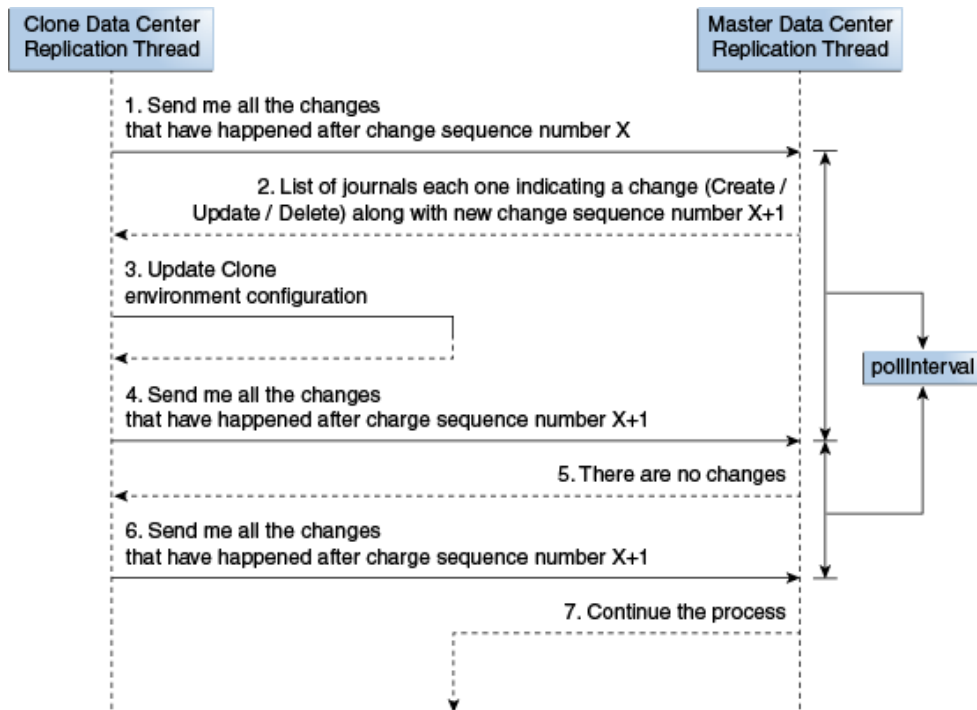
```
curl -u <repluser> -H 'Content-Type: application/json' -X POST
'https://supplier.example.com:7002/oam/services/rest/
_replication/setup' -d '{"name":"DC12DC2",
"source":"DC1","target":"DC2","documentType":"ENTITY"}'
```

The following is an example of output for the command.

```
{"enabled":"true","identifier":"201409231329353668","ok":"true",
"pollInterval":"900","startingSequenceNumber":"110","state" : "READY"}
```

Be sure to note the values of the replication identifier, `pollInterval` and `startingSequenceNumber`. The identifier is a reference specific to this Replication Agreement and is used for replication related queries. The `pollInterval` is a value (in seconds) after which the Clone will poll for changes against the Master. (Typically policy and configuration are not changed often so this number can be as high as the default value of 900 seconds.) The `startingSequenceNumber` is the value before which all records will be unavailable. In the example, all records before the value of 110 are unavailable. It is implicit that bootstrapping happened before creating the Replication Agreement thus the Clone can start pulling changes from sequence number 110. The Clone also has an entry created in its local replication table which keeps track of the last sequence number. The starting sequence process is illustrated in [Figure 19-2](#).

Figure 19-2 Starting Sequence Illustrated



The create replication agreement command will return details of an already existing replication agreement if applicable. In this case, the value of ok will be false.

```

{"enabled":"true","identifier":"201409231329353668","ok":"false",
 "pollInterval":"900","startingSequenceNumber":"110",
 "state":"READY"}

```

Note:

If a specific user needs to be used for replication, the user's credentials can be provided in the command in the format "BASIC base64(user:password)". For example, "BASIC base64(weblogic:welcome1)" is specified as "BASIC d2VibG9naWM6d2VsY29tZTE=" in the following command.

```

curl -u <repluser> -H 'Content-Type: application/json' -X POST
  'https://supplier.example.com:7002/oam/services/rest/
  _replication/setup' -d
  '{"source":"DC1","target":"DC2","documentType":"ENTITY","config":
  {"entry":{"key":"authorization","value":"BASIC
  d2VibG9naWM6d2VsY29tZTE="}}}'

```

Basic Authorization is supported for replication REST API.

3. Restart the Master and Clone AdminServers.

Once the replication agreement is created and the AdminServers restarted, the Clone will start polling for changes. See [Modifying Polling Interval](#).

19.4.3 Modifying a Replication Agreement

Using the Replication Agreement identifier, changes can be made to the Replication Agreement configuration. Replication Agreement properties (enabled status, poll interval and the like) can be updated by executing a REST request at the Master's endpoint. Either the Master or Clone Replication Agreement will be updated as specified by the value of the `replicaType` parameter. The clone will poll for changes, apply them and wait the duration specified as the `pollInterval`.

In this example, the value of `pollInterval` will be changed to 60 seconds.

Service responds back with JSON object that is the status of replication agreement before making the change. You need to fetch replication agreement status again to see updated configuration.

1. Query the existing replication agreements using the following command and obtain the replication Identifier, `replId` that needs to be used in the following steps.

```
curl -k -u weblogic:password 'https://oamadmin1-dcl.poc.com:7002/oam/services/rest/_replication/agreements'
```

 **Note:**

If there are multiple replication agreements, select the identifier for which replication agreement needs to be modified by querying the corresponding Clone data center.

2. Execute the following command to get the current status of the Replication Agreement in Clone machine:

```
curl -k -u weblogic:password 'https://oamadmin1-dcl.poc.com:7002/oam/services/rest/_replication/201409040157218184?type=consumer'
```

The JSON response would be:

```
{"enabled":"true","identifier":"201409040157218184","ok":"true","pollInterval":"900","startingSequenceNumber":"101","state":"READY"}
```

3. Execute the following command to modify the value of `pollInterval` in Clone machine:

```
curl -k -u weblogic:password 'Content-Type: application/json' -X PUT 'https://oamadmin1-dcl.poc.com:7002/oam/services/rest/_replication/201409040157218184' -d '{"pollInterval":"60","replicaType":"CONSUMER"}
```

The JSON response would be:

```
{"enabled":"true","identifier":"201409040157218184","ok":"true","pollInterval":"60","startingSequenceNumber":"101","state":"READY"}
```

4. Restart the AdminServer on Clone machine.
5. Execute the following command to get the current status of the Replication Agreement.

This will validate that the change has been made. Note the value of `pollInterval` in the JSON Response is different from the value returned in the first step of this procedure.

```
curl -k -u weblogic:password 'https://oamadmin1-dcl.poc.com:7002/oam/services/rest/_replication/201409040157218184?type=consumer'
```

The JSON response would be:

```
{ "enabled": "true", "identifier": "201409040157218184?", "ok": "true", "pollInterval": "60", "startingSequenceNumber": "101", "state": "READY" }
```

Table 19-2 Modifying Replication Agreement Properties

Property	Modification Command
BatchSize	<p>Number of change records (journals) returned by the master as a result of a getChanges query by clone. Ideally the default batch size of 32 is sufficient as all changes are pulled in multiple batches as part of fetching. However if the setup needs a large batch size, execute the following command:</p> <pre>curl -k -u weblogic:password -H 'Content-Type: application/json' -X PUT 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/<replid>' -d '{"batchSize": "100", "replicaType": "SUPPLIER"}'</pre>
User Context	<p>In rare instances, the user context for replication poll may need to be modified.</p> <pre>curl -k -u weblogic:password -H 'Content-Type: application/json' -X PUT 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/201409231329353668' -d '{"replicaType": "CONSUMER", "config": {"entry": {"key": "authorization", "value": " BASIC cG9sbHVzZXI6c2VjcmV0"}}}'</pre> <p>'cG9sbHVzZXI6c2VjcmV0' is a base 64 encoded value for polluser credentials. Any user credentials can be used here instead of the repluser which is used to execute the command.</p>

19.4.4 Deleting a Replication Agreement

A Replication Agreement can be deleted by executing the REST API at the Master data center's endpoint. Replication Agreements that are currently active and in use cannot be deleted until the Master and Clone have been disabled. Disable the replication agreement before you remove it from on the Clone and Master data centers.

1. Disable the replication agreement on the Clone as follows:

```
curl -u <repluser> -H 'Content-Type: application/json' -X PUT
  'https://supplier.example.com:7002/oam/services/rest/_replication/
  201409231
  329353668' -d '{"enabled": "false", "replicaType": "CONSUMER"}'
```

2. Disable the replication agreement on the Master as follows:

```
curl -u <repluser> -H 'Content-Type: application/json' -X PUT
  'https://supplier.example.com:7002/oam/services/rest/_replication/
  201409231
  329353668' -d '{"enabled": "false", "replicaType": "SUPPLIER"}'
```

3. Delete the replication agreement as follows:

```
curl -u weblogic:welcome1 -H 'Content-Type: application/json' -X DELETE
  'https://supplier.example.com:7002/oam/services/rest/_replication/
201409231
329353668'
```

19.5 Customizing Transformation Rules

Transformation rules are used by APS and you can modify rules, as required.

The transformation rules illustrated in the following example are the default rules provided by Access Manager. A Clone can be configured to override these OOTB rules. This section documents how some of these rules can be modified and how to configure Access Manager to recognize these custom rules.

1. Specify the custom transformation rules in a file (say, `transformationRules.txt`). It is recommended to copy the default OOTB transformation rules specified as follows and modify rules as required.

```
<?xml version="1.0" encoding="UTF-8"?>
<mdc-transform-rule>
<changes-to-include entity-path="/policy"/>
<changes-to-include entity-path="/oauth"/>
<changes-to-include entity-path="/IDM"/>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Agent/WebGate/
Instance" />
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/AuthenticationModules"/>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/oamproxy"/>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/Sme/SessionConfigurations"/>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/OAMServerProfile">
<ignore attribute-match="/OAMSERVER/serverprotocol"/>
<ignore attribute-match="/OAMSERVER/serverhost"/>
<ignore attribute-match="/OAMSERVER/serverport"/>
<ignore attribute-match="/OAMSERVER/serversslterminated"/>
<ignore attribute-match="/HostAlias/oamserverHttps/serverprotocol"/>
<ignore attribute-match="/HostAlias/oamserverHttps/serverhost"/>
<ignore attribute-match="/HostAlias/oamserverHttps/serverport"/>
</changes-to-include>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/OAMServerProfile/OAMSERVER">
<ignore attribute-match="/serverprotocol"/>
<ignore attribute-match="/serverhost"/>
<ignore attribute-match="/serverport"/>
```

```

<ignore attribute-match="/serversslterminated"/>
</changes-to-include>
<changes-to-include
entity-path="/config/NGAMConfiguration/DataCenterConfiguration/
Cluster">
<ignore attribute-match="/DataCenterType"/>
<ignore attribute-match="/ClusterId"/>
<ignore attribute-match="/WriteEnabledFlag"/>
</changes-to-include>
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Descriptors/OAMSEntityDescriptor" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Federation/IdentityProvider" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Federation/ServiceProvider" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Server/NGAMServer/Profile/STS/
fedattributeprofiles" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Server/NGAMServer/Profile/STS/
fedpartnerprofiles" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Server/NGAMServer/Profile/STS/fedserverconfig" />
</mdc-transform-rule>

```

The rule document will mention the XPath of system configuration artifacts to be replicated for a clone. If there is any transformation to be done during replication for the entry in XPath, it can be provided as replacement rule for that clone. To add a new XPath for replication to a clone, create a new transformation XML file, using the above XML document as a template. Add and remove XPaths as required. For example, adding the following XPath as the child of an <mdc-transformation-rule> node and saving the file to the clone's file system will modify Available Services.

2. Add the following environmental variable to the *EXTRA_JAVA_PROPERTIES* of *setDomainEnv.sh* (On linux or Unix) or *setDomainEnv.cmd* (On Windows) file available at *\$MW_HOME/user_projects/domains/OAMDomain/bin*.

```
-Doracle.oam.MDCRuleFile=/scratch/transformationRules.txt
```

Each Clone can use different transformation rules. [Figure 19-3](#) illustrates how to apply these custom rules.

Figure 19-3 Applying Custom Transformation Rules

```

#Adding -Doracle.oam.MDCRuleFile=/scratch/transformationRules.txt to EXTRA_JAVA_PROPERTIES for enabling custom transformation rules.
EXTRA_JAVA_PROPERTIES=" -Djavax.management.builder.initial=weblogic.management.jmx.mbeanserver.MLSMBeanServerBuilder -Doracle.oam.MDCRuleFile=/scratch/transformationRules.txt
${EXTRA_JAVA_PROPERTIES}"
export EXTRA_JAVA_PROPERTIES

```

3. Restart Clone Admin and Managed Servers.

This transformation rule will not change the WebGate agent definitions. The following information details how you can modify these changes for the *PrimaryServerList* and *logoutRedirectUrl* attributes.

- You can use the transformation rule in the following example to update the *PrimaryServerList* and *logoutRedirectUrl* attributes with the Access Manager

server host from the Clone environment. This change can be viewed in the oam-config.xml file; it replaces the value of the PrimaryServerList attribute with the value equal to \${DeployedComponent/Server/NGAMServer/Profile/OAMServerProfile/OAMSERVER/serverhost}; for example, oam1-lon.example.com. The limitation of this rule is that it updates all servers in the primary list.

```
<changes-to-include
  entity-path="/config/NGAMConfiguration/DeployedComponent/Agent/WebGate/
Instance">
  <replace attribute-match="*/PrimaryServerList*/host" value-
match="(.*)">
    <replace-with
n="1">${/config/NGAMConfiguration/DeployedComponent/Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverhost}</replace-with>
    </replace>
    <replace attribute-match="*/UserDefinedParameters/logoutRedirectUrl"
value-match="(.*)<a target="_blank" href="://">://</a>(.*):(.*)/oam/
server/logout">
    <replace-with
n="1">${/config/NGAMConfiguration/DeployedComponent/Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverprotocol}</replace-with>
    <replace-with
n="2">${/config/NGAMConfiguration/DeployedComponent/Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverhost}</replace-with>
    <replace-with
n="3">${/config/NGAMConfiguration/DeployedComponent/Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverport}</replace-with>
    </replace>
    <replace attribute-match="*/logoutRedirectUrl"
value-match="(.*)<a target="_blank" href="://">://</a>(.*):(.*)/oam/
server/logout">
    <replace-with
n="1">${/config/NGAMConfiguration/DeployedComponent/Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverprotocol}</replace-with>
    <replace-with
n="2">${/config/NGAMConfiguration/DeployedComponent/Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverhost}</replace-with>
    <replace-with
n="3">${/config/NGAMConfiguration/DeployedComponent/Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverport}</replace-with>
    </replace>
  </changes-to-include>
```

You can use the transformation rule in the following example to update servers in PrimaryServerList with the different Clone servers.

```
<changes-to-include entity-path=
"/config/NGAMConfiguration/DeployedComponent/Agent/WebGate/Instance">
  <replace attribute-match="*/PrimaryServerList/0/host" value-match="(.*)">
    <replace-with n="1">${/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/
Instance/oam_server1/host"}
    </replace-with>
  </replace>
  <replace attribute-match="*/PrimaryServerList/1/host" value-match="(.*)">
```

```

        <replace-with n="1">${"/config/NGAMConfiguration/DeployedComponent/
Server/NGAMServer/
Instance/oam_server2/host"}
    </replace-with>
</replace>
</changes-to-include>

```

 **Note:**

The OAM Managed Servers such as `oam_server1` and `oam_server2` must be updated with the names specified during deployment.

A load balancer is recommended between the WebGate and Access Manager server. In this case, you do not have to update the `PrimaryServerList` across data centers and can remove this transformation rule from the XML.

The following example illustrates how to change the transformation rule to update the `PrimaryServerList` only for `WebgateAgent1` and `WebgateAgent2` agents and not `WebGate` agents.

```

<changes-to-include entity-path="/config/NGAMConfiguration/DeployedComponent/
Agent/WebGate/Instance">
    <replace attribute-match="/WebgateAgent1/PrimaryServerList/*/host" value-
match="(.)">
        <replace-with n="1">${"/config/NGAMConfiguration/DeployedComponent/
Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverhost"}
    </replace-with>
</replace>
    <replace attribute-match="/WebgateAgent2/PrimaryServerList/*/host" value-
match="(.)">
        <replace-with n="1">${"/config/NGAMConfiguration/DeployedComponent/
Server/NGAMServer/
Profile/OAMServerProfile/OAMSERVER/serverhost"}
    </replace-with>
</replace>
</changes-to-include>

```

- The `logoutRedirectUrl` attribute updates the logout URL protocol, host and port for all `WebGate` agents with respective values from the Clone. If a load balancer is used globally to define the logout URL for all `WebGate` agents in the Master environment, you don't need to replace the logout URL in the Clone environment and can remove the transformation rule. If you are using a DCC authentication scheme and a global load balancer host name to define the DCC login and logout URL, then again you don't need to replace the login and logout URL in the Clone environment and can remove the transformation rule.

19.6 Disabling Automated Policy Synchronization

Disable and then delete the replication agreements from the Master and Clone data centers.

1. Query the existing replication agreements using the following command and obtain the replication Identifier, `replId`; use it in Step 2 through Step 4.

```
curl -k -u weblogic:password 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/agreements'
```

 **Note:**

If there are multiple replication agreements, select the identifier for which APS needs to be disabled by querying the corresponding Clone Data Center.

2. Disable the replication agreement on Clone data center.

```
curl -u weblogic:password -H 'Content-Type: application/json' -X PUT 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/replId' -d '{"enabled":"false","replicaType":"CONSUMER"}'
```

3. Disable the replication agreement on Master data center.

```
curl -u weblogic:password -H 'Content-Type: application/json' -X PUT 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/replId' -d '{"enabled":"false","replicaType":"SUPPLIER"}'
```

4. Delete the replication agreement from Master and Clone data centers.

```
curl -u weblogic:password -H 'Content-Type: application/json' -X DELETE 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/replId'
```

19.7 Best Practices for Replication

Here are some best practices to help you set up data replication.

The following points and the information in the sections should be taken into account when setting up data replication.

- It is recommended that as many Policy Domain artifacts are created as possible before cloning. This will help the replication manager work efficiently during incremental updates.
- The OAM server instance list will be used as the Well Known Addresses (WKA) to create a Coherence cluster so do not add other data center servers to the server instance list.
- To allow for a WebGate profile to point to a remote data center in the secondary server list, use the Other option to provide OAP with the host and port details of the remote data center.
- See [Enabling Replication Logs](#)
- See [Changing the User Identifier](#)

19.7.1 Enabling Replication Logs

To get detailed logs on replication agreement and replication poll-related issues, enable the logger 'oracle.oam.replication' by executing the WLST command.

```
setLogLevel(logger="oracle.oam.replication", level="TRACE:32", persist="0",  
target="AdminServer")
```

This enables logger only till next shutdown of AdminServer. To keep the logger state across restart, set the persist attribute to "1".

19.7.2 Changing the User Identifier

While creating replication agreement if you have not specified any authorization header of the user to be used for replication if the user's password got changed at later point, you can edit the replication agreement with the latest user identity and password.

Run the following command:

```
curl -u <repluser> -H 'Content-Type: application/json' -X PUT  
'https://supplier.example.com:7002/oam/services/rest/_replication/201409231  
329353668' -d '{"replicaType":"CONSUMER","config":{"entry":  
{"key":"authorization","value":"BASIC d2VibG9naWM6d2VsY29tZTE="}}}'
```

20

Setting Up the Multi-Data Center: A Sequence

The sequence of steps in this chapter will help you to setup a Multi-Data Center with four nodes using Oracle Access Manager. The configuration spans two Data Centers with two nodes per Data Center. The nodes are configured in Active/Active Mode.

This chapter contains the following section.

- [Before You Begin](#)
- [Setting Up a Multi-Data Center](#)
- [Enabling Automated Policy Synchronization](#)
- [Troubleshooting the Multi-Data Center Setup](#)

20.1 Before You Begin

Before you proceed with MDC configuration ensure the system level requirements are met.

Read the following chapters before beginning the steps documented in this sequence for an understanding of Multi-Data Center and its features.

- [Understanding Multi-Data Centers](#)
- [Configuring Multi-Data Centers](#)
- [Synchronizing Data In A Multi-Data Center](#)

Confirm the following before you begin the Multi-Data Center set-up sequence.

- Check that your operating system is up-to-date with all necessary patches applied.
- Verify that each machine has more than 30 GB space available and more than 8GB of memory available.
- Mount the binaries you will be using. The applicable Oracle software includes:
 - Oracle Fusion Middleware Identity and Access Management 12c (12.2.1.4.0)
 - Oracle WebLogic Server 12c (12.2.1.4.0)
 - Oracle Database 12c (12.2.x.x)
 - Oracle Fusion Middleware Repository Creation Utility 12c (12.2.1.4.0)
- Add `/etc/hosts` entries on all four nodes being configured.
- Verify that the Oracle Database is connected and accessible.
- OAM Admin server in the Master and Clone data center should be SSL-enabled.
- The OAM managed servers in the Master and Clone data centers should be SSL-enabled or the load balancer front-ending the OAM managed servers should be SSL-terminated or both. Before restarting the servers, ensure that the load balancer is configured in the OAM Admin Console of the Master data center.
- All the managed servers in the Master and Clone data centers should be configured with the same security mode.

- Use SIMPLE mode to secure communication between OAM Servers and WebGates using out-of-box certificates.
- Use CERT mode, if you have access to a trusted third-party Certificate Authority (CA).
- The ID Stores are configured for Master and Clone data centers and they have the same name.

20.2 Setting Up a Multi-Data Center

For a successful set-up of a Multi-Data Center with data replication, the configuration spans two data centers with two nodes per data center. The nodes are configured in Active-Active Mode. MDC Admin REST APIs are used for diagnostics and configuration of Master and Clone data centers.

1. Install the Java Development Kit (JDK) 1.8.0 on Data Center 1 Node 1 and set the appropriate environment variables.
2. Install WebLogic Server 12c (12.2.1.4.0) on Data Center 1, Node 1.
This process creates the Middleware Home (<MW_HOME>).
3. Install the Oracle Identity and Access Management 12c (12.2.1.4.0) software on Data Center 1, Node 1.

Oracle Identity and Access Management contains the Oracle Access Management suite which includes Oracle Access Manager.

4. Run the Repository Creation Utility (RCU) 12c (12.2.1.4.0) on Data Center 1, Node 1.

It creates and loads the appropriate database schemas for Oracle Identity and Access Management products. And you can run it from `$MW_HOME/oracle_common/bin`.

5. Configure Oracle Access Management on Data Center 1, Node 1 using the Oracle Fusion Middleware Configuration Wizard script, `$MW_HOME/oracle_common/common/bin/config.sh` script (on Linux or UNIX), or `$MW_HOME\oracle_common\common\bin\config.cmd` (on Windows). Minimally, you will be configuring:

- a new WebLogic domain
- an Oracle Access Management Administration Server
- an Oracle Access Management Managed Server
- Oracle Access Manager

6. Modify the following WebLogic scripts on Data Center 1, Node 1:

On Linux or UNIX:

- a. Open `startWeblogic.sh` and `startManagedWeblogic.sh` using `vi` and enter the appropriate value for `WLS_USER`.

Enter the password, if prompted, while starting the servers; do not hard code it.

- b. Save `startWeblogic.sh` and `startManagedWeblogic.sh`.

On Windows:

- a. Open `startWeblogic.cmd` and `startManagedWeblogic.cmd` using notepad and enter the appropriate value for `WLS_USER`.
Enter the password, if prompted, while starting the servers; do not hard code it.
 - b. Save `startWeblogic.cmd` and `startManagedWeblogic.cmd`
7. Start the Administration and Managed Servers.
 8. Repeat Step 1 through Step 7 to prepare other data centers until you have added all the required nodes of your Master and Clone data centers (Data Center 1, Node 2; Data Center 2, Node 1; Data Center 2, Node 2) to the MDC setup.
 9. Run pack and unpack scripts within the same data center to create a Managed Server JAR and to copy the JAR between nodes.

- a. Run `pack.sh` located in the `<MW_HOME>/oracle_common/common/bin` directory to create the JAR file on Data Center 1, Node 1.

```
./pack.sh -domain=$MW_HOME/user_projects/domains/OAMDomain  
-template=OAMManagedServer.jar -template_name="OAM Domain" -  
managed=true
```

- b. Copy `OAMManagedServer.jar` to the `MW_HOME/oracle_common/common/bin` directory on Data Center 1, Node 2.
- c. Run `unpack.sh` to unpack the Managed server JAR on Data Center 1, Node 2. The JAR is used as a template to create the OAMDomain Domain Structure on Data Center 1, Node 2.

```
mkdir -p $MW_HOME/user_projects/domains/OAMDomain  
cd <MW_HOME>/oracle_common/common/bin  
./unpack.sh -domain=$MW_HOME/user_projects/domains/OAMDomain -  
template=OAMManagedServer.jar
```

- d. Repeat the same procedure (Step a through Step c) to create `OAMManagedServer.jar` on Data Center 2, node 1 and to copy it to `MW_HOME/oracle_common/common/bin` directory on Data Center 2, node 2.
10. At this point in the sequence, the Data Center 1 cluster, Data Center 2 cluster and its four nodes are configured and ready for Multi-Data Center configurations. Perform a validation check at this point:
 - OAM Admin server in the Master and Clone data center should be SSL-enabled.
 - The OAM managed servers in the Master and Clone data centers should be SSL-enabled or the load balancer front-ending the OAM managed servers should be SSL-terminated or both. Before restarting the servers, ensure that the load balancer is configured in the OAM Admin Console of the Master data center.
 - All the managed servers in the Master and Clone data centers should be configured with the same security mode.
 - Use SIMPLE mode to secure communication between OAM Servers and WebGates using out-of-box certificates.
 - Use CERT mode, if you have access to a trusted third-party Certificate Authority (CA).
 - The ID Stores are configured for Master and Clone data centers and they have the same name.

Optionally, You can run the diagnostic REST APIs on the Master and the Clone Data Centers to view the MDC configuration settings:

```
curl -k -u weblogic:password 'https://oamadmin1-
dc1.poc.com:7002/oam/services/rest/mdc/configuration'
curl -k -u weblogic:password 'https://oamadmin1-
dc2.poc.com:7002/oam/services/rest/mdc/configuration'
```

Verify the following from the output of the command:

- In `dcConfigMap` entry, `MultiDataCenterEnabled` should be false and `MultiDataCenterPartners` list should be empty.
- `agentMap` entry should be empty.

Note: If `MultiDataCenterEnabled` is true (MDC is already enabled) and the data center has to be setup again for some reasons, see [Overwriting the Existing MDC Configuration or Recovering from an Inconsistent State](#)

See MDC Diagnostic REST API in *REST API for Multi Data Center in Oracle Access Manager*.

11. Run the following command with appropriate values to configure the Master data center.

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/
master'
-d
'{"mdcTopologyType":"value", "masterMDCAgentID":"value", "cloneMDCAge
ntID":"value",
"accessClientPassword":"value", "artifactPassword":"value",
"cloneServerURL":"value", "cloneServerPolicyManagerURL":"value", "mast
erServerPolicyManagerURL":"value",
"agentKeyPassword":"value", "certModeKeystorePassword":"value", "maste
rServerURL":"value",
"cloneAdminUserNamePassword":"value", "trustStorePath":"value",
"keyStorePath":"value", "artifactsZipLocation":"value"}'
```

- `mdcTopologyType`: Choose one of the two topology types available for MDC configuration, `ACTIVE_ACTIVE` or `DISASTER_RECOVERY`.
- `masterMDCAgentID`: Enter the MDC NAP Agent Name for the Master data center.
- `cloneMDCAgentID`: Enter the MDC NAP Agent Name for the Clone data center.
- `accessClientPassword`: Provide the password required to be used by the MDC NAP agents in Master and Clone data centers.
- `artifactPassword`: Provide the password that is used to protect cloning artifacts.
- `cloneServerURL`: Enter the URL of the Clone Admin server or the URL of the reverse proxy front ending the Clone Admin server.

- `cloneServerPolicyManagerURL`: Enter the URL of the Clone Policy Manager or the URL of the reverse proxy front ending the Clone Policy Manager.
- `masterServerPolicyManagerURL`: Enter the URL of the Master Policy Manager or the URL of the reverse proxy front ending the Master Policy Manager.
- (Only for CERT mode) `agentKeyPassword`: Enter the agent key password used to register partners in the CERT mode.
- (Only for CERT mode) `certModeKeystorePassword`: Enter the keystore password used to protect `clientTrustStore.jks` and `clientKeyStore.jks`.
- (Optional) `masterServerURL`: Enter the URL of the Master Admin server or the URL of the reverse proxy front ending the Master Admin Server.
- (Optional) `cloneAdminUserNamePassword`: Enter the user credentials of the Clone data center's Administrator if the username and password of the Administrator for Master and Clone data centers are different.
- (Optional) `trustStorePath`: Enter the following depending on SIMPLE or CERT mode:
 - For SIMPLE mode : Provide the path to `oamclient-truststore.jks` file if this file is available in folders other than `$MW_HOME/user_projects/domains/OAMDomain/output/webgate-ssl-SHA-256/`
 - For CERT mode : Provide the path to `clientTrustStore.jks` file if this file is available in folders other than `$MW_HOME/user_projects/domains/OAMDomain/config/fmwconfig/oam-mdc-cert-artifacts/`
- (Optional) `keyStorePath`: Enter the following depending on SIMPLE or CERT mode:
 - For SIMPLE mode : Provide the path to `oamclient-keystore.jks` file if this file is available in folders other than `$MW_HOME/user_projects/domains/OAMDomain/output/webgate-ssl-SHA-256/`
 - For CERT mode : Provide the path to `clientKeyStore.jks` file if this file is available in folder other than `$MW_HOME/user_projects/domains/OAMDomain/config/fmwconfig/oam-mdc-cert-artifacts/`
- (Optional) `artifactsZipLocation`: Provide the location where cloning artifacts has to be stored; specify only if cloning artifacts need to be stored in any location other than `/tmp`

Here are the sample Curl commands for configuring a Master data center in SIMPLE and CERT modes using Active-Active MDC topology:

- Using CERT mode:

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/
master'
-d '{"mdcTopologyType":"ACTIVE_ACTIVE",
"masterMDCAgentID":"MDCmasterNAPagent","cloneMDCAgentID":"MDCcloneNAPa
gent",
"accessClientPassword":"password","artifactPassword":"password",
"cloneServerURL":"https://oamadmin1-
dc2.poc.com:7002","cloneServerPolicyManagerURL":"https://oamadmin1-
dc2.poc.com:14151","masterServerPolicyManagerURL":"https://oamadmin1-
dc1.poc.com:14151",
```

```
"cloneAdminUserNamePassword":"weblogic:password","agentKeyPassword":"password", "certModeKeystorePassword":"password"}'
```

- Using SIMPLE mode:

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master'
-d '{"mdcTopologyType":"ACTIVE_ACTIVE",
"masterMDCAgentID":"MDCmasterNAPagent", "cloneMDCAgentID":"MDCcloneNAPagent",
"accessClientPassword":"password", "artifactPassword":"password",
"cloneServerURL":"https://oamadmin1-dc2.poc.com:7002", "cloneServerPolicyManagerURL":"https://oamadmin1-dc2.poc.com:14151", "masterServerPolicyManagerURL":"https://oamadmin1-dc1.poc.com:14151",
"cloneAdminUserNamePassword":"weblogic:password"}'
```

See MDC Master REST API in *REST API for Multi Data Center in Oracle Access Manager*.

12. Run the following command with appropriate values to configure the Clone data center.

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone'
-d
 '{"masterServerURL":"value", "artifactPassword":"value", "masterAdminUserNamePassword":"value",
"masterServerPolicyManagerURL":"Value", "artifactsZipLocation":"value", "masterArtifactsZipLocation":"value"}'
```

- `masterServerURL`: Enter the URL of the Master Admin server or the URL of the reverse proxy front ending the Master Admin Server.
- `masterServerPolicyManagerURL`: Enter the URL of the Master Policy Manager or the URL of the reverse proxy front ending the Master Policy Manager.
- `artifactPassword`: Provide the same password that protects cloning artifacts and used while setting up the Master data center
- (Optional) `masterAdminUserNamePassword`: Enter the user credentials of the Master data center's Administrator if the username and password of the Administrator for Master and Clone data centers are different.
- (Optional) `artifactsZipLocation`: Provide the location where backup artifacts should be stored in Clone data center (artifacts present in Clone data center are backed up before replacing it with Master artifacts); specify only when the backup artifacts need to be stored in any location other than `/tmp`.
- (Optional) `masterArtifactsZipLocation`: Provide the location where cloning artifacts are present in Master data center; specify only

when `artifactsZipLocation` was used in input while configuring the Master data center.

Here is the sample Curl command for configuring a Clone data center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone'
-d '{"masterServerURL":"https://oamadmin1-dc1.poc.com:7002/",
"masterServerPolicyManagerURL":"https://oamadmin1-dc1.poc.com:14151",
"artifactPassword":"password","masterAdminUserNamePassword":"password"}'
```

See MDC Clone REST API in *REST API for Multi Data Center in Oracle Access Manager*.

13. Run the following command to reconfigure the Clone Data Center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
' https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone/
configuration'
```

 **Note:**

This command does not require any input parameters. It updates the flag, `DataCenterType` to `Clone`. To make the clone write-protected, execute the WLST command `setMultiDataCenterWrite(WriteEnabledFlag="false")`. It ignores any update to clone configuration.

See MDC Reconfigure Clone REST API in *REST API for Multi Data Center in Oracle Access Manager*.

14. Restart Clone Administration and managed servers.
15. Run the following diagnostic REST API on the Master and the Clone Data Centers to verify MDC configurations:

```
curl -k -u weblogic:password 'https://oamadmin1-dc1.poc.com:7002/oam/
services/rest/mdc/configuration'
curl -k -u weblogic:password 'https://oamadmin1-dc2.poc.com:7002/oam/
services/rest/mdc/configuration'
```

16. Export the partner and policy information from Data Center 1, Node 1 and then import it to Data Center 2, Node 1.
 - a. To export, change to the `$MW_HOME/oracle_common/common/bin` directory and run WLST to export from Data Center 1, Node 1.

```
./wlst.sh
connect()
exportAccessStore(toFile="<name and location of the master metadata
ZIP file>", namePath="/")
exit()
```

- b. Copy the exported file (that is, *<name and location of the master metadata ZIP file>*) from Data Center 1, Node 1 to Data Center 2, Node 1. To import, change to the `$MW_HOME/oracle_common/common/bin` directory and run WLST to import on Data Center 2, Node 1.

```
./wlst.sh
connect()
importAccessStore(fromFile="<name and location of master
metadata ZIP file>", namePath="/")
exit()
```

20.3 Enabling Automated Policy Synchronization

Enabling the Automated Policy Synchronization (APS) feature for automated data synchronization among the servers includes commands for testing the REST services as well as details on adding custom transformation rules to the synchronization.

See [Synchronizing Data In A Multi-Data Center](#) for details on APS and transformation rules.

After exporting the partner and policy information from Master data center to Clone data center, perform the following steps to enable APS:

1. Validate the REST services using the following commands:

```
curl -u weblogic:password 'https://oamadmin1-dc1.poc.com:7002/oam/
services/rest/_replication/hello'
curl -u weblogic:password 'https://oamadmin1-dc2.poc.com:7002/oam/
services/rest/_replication/hello'
```

2. Run the following command in the Master and Clone Data Centers to get the `clusterName`:

```
curl -k -u weblogic:password 'https://oamadmin1-
dc1.poc.com:7002/oam/services/rest/mdc/dc/configuration'
curl -k -u weblogic:password 'https://oamadmin1-
dc2.poc.com:7002/oam/services/rest/mdc/dc/configuration'
```

3. Run the following command with appropriate values to setup replication agreement:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X
POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/
_replication/setup' -d
'{"name":"value","source":"value","target":"value","documentType":"E
NTITY","config":{"entry":
{"key":"authorization","value":"authzValue"}}}'
```

- `name`: Enter a name for the replication agreement.
- `source`: Enter the cluster name of the Master data center (`clusterName` obtained as a result of first command in step 2).

- `target`: Enter the cluster name of the Clone data center (`clusterName` obtained as a result of second command in step 2).
- `documentType`: Default value for this parameter is ENTITY.
- `Config`: The map that contains key value pairs.
 - a. (Optional) `authzValue`: If the username and password of the Administrator for Master and Clone data centres are different, then enter the value of Authorization Header (`authzValue`) to be used for contacting the Clone data center. Authorization Header will be Base 64 encoded value of `CloneAdminUser:CloneAdminPassword`.

For example,

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/setup'
-d '{"name":"DC12DC2","source":"e60ef-oamadmin1-dc1.u","target":"70d7b-
oamadmin1-dc2.u","documentType":"ENTITY","config": {"entry":
{"key":"authorization","value":"Basic b2FtQWRtaW5Vc2VyOldlbGNvbWUy"}}}'
```

 **Tip:**

The configuration changes made to the Master data center will take at least 900 Seconds to get propagated to the Clone data center as the default POLLINTERVAL is 900 Seconds. However, you can modify this polling interval in the Clone data center using SQLDeveloper in the database after setting up the Replication Agreement. See [Modifying Polling Interval in Clone Data Centers](#).

 **Note:**

To enable Replication, it is not mandatory in 12c to set the flag `-Doracle.oam.EnableMDCReplication to true`.

4. Run the following command with appropriate values to setup replication agreement for OAuth Consent management for MDC:

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc1.poc.com:14151/oam/services/rest/
_replication/setup'
-d
'{"name":"value","source":"value","target":"value","documentType":"RUNTIME
_ENTITY",
"config": {"entry":{"key":"authorization","value":"authzValue"}}}'
```

- `name`: Enter a name for the replication agreement.
- `source`: Enter the cluster name of the Master data center (`clusterName` obtained as a result of first command in step 2).
- `target`: Enter the cluster name of the Clone data center (`clusterName` obtained as a result of second command in step 2).

- `documentType`: Specify the value as `RUNTIME_ENTITY`.
- `Config`: The map that contains key value pairs.
 - a. (Optional) `authzValue`: If the username and password of the Administrator for Master and Clone data centres are different, then enter the value of Authorization Header (`authzValue`) to be used for contacting the Clone data center. Authorization Header will be Base 64 encoded value of `CloneAdminUser:CloneAdminPassword`.

For example,

```
curl -k -u weblogic:password -H 'Content-Type: application/json'
-X POST 'https://oamadmin1-dc1.poc.com:14151/oam/services/rest/
_replication/setup'
-d '{"name":"DC1Consent2DC2Consent","source":"e60ef-oamadmin1-
dc1.u","target":"70d7b-oamadmin1-dc2.u",
"documentType":"RUNTIME_ENTITY","config":{"entry":
{"key":"authorization","value":"Basic
b2FtQWRtaW5Vc2VyO1dlbGNvbWUy"}}}'
```

Tip:

The configuration changes made to the Master data center will take at least 900 Seconds to get propagated to the Clone data center as the default `POLLINTERVAL` is 900 Seconds. However, you can modify this polling interval in the Clone data center using `SQLDeveloper` in the database after setting up the Replication Agreement. See [Modifying Polling Interval in Clone Data Centers](#).

Note:

To enable Replication, it is not mandatory in 12c to set the flag `-Doracle.oam.EnableMDCReplication` to `true`.

5. Optionally, create transformation rules, `/tmp/transformationrules.xml` file. Use the following content and modify rules, as required:

```
<?xml version="1.0" encoding="UTF-8"?>
<mdc-transform-rule>
<changes-to-include entity-path="/policy"/>
<changes-to-include entity-path="/oauth"/>
<changes-to-include entity-path="/IDM"/>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Agent/
WebGate/Instance" />
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/AuthenticationModules"/>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/oamproxy"/>
```

```

<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/Sme/SessionConfigurations"/>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/OAMServerProfile">
<ignore attribute-match="/OAMSERVER/serverprotocol"/>
<ignore attribute-match="/OAMSERVER/serverhost"/>
<ignore attribute-match="/OAMSERVER/serverport"/>
<ignore attribute-match="/OAMSERVER/serversslterminated"/>
<ignore attribute-match="/HostAlias/oamserverHttps/serverprotocol"/>
<ignore attribute-match="/HostAlias/oamserverHttps/serverhost"/>
<ignore attribute-match="/HostAlias/oamserverHttps/serverport"/>
</changes-to-include>
<changes-to-include
entity-path="/config/NGAMConfiguration/DeployedComponent/Server/
NGAMServer/Profile/OAMServerProfile/OAMSERVER">
<ignore attribute-match="/serverprotocol"/>
<ignore attribute-match="/serverhost"/>
<ignore attribute-match="/serverport"/>
<ignore attribute-match="/serversslterminated"/>
</changes-to-include>
<changes-to-include
entity-path="/config/NGAMConfiguration/DataCenterConfiguration/Cluster">
<ignore attribute-match="/DataCenterType"/>
<ignore attribute-match="/ClusterId"/>
<ignore attribute-match="/WriteEnabledFlag"/>
</changes-to-include>
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Descriptors/OAMSEntityDescriptor" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Federation/IdentityProvider" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Federation/ServiceProvider" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Server/NGAMServer/Profile/STS/fedattributeprofiles" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Server/NGAMServer/Profile/STS/fedpartnerprofiles" />
<changes-to-include entity-path="/config/NGAMConfiguration/
DeployedComponent/Server/NGAMServer/Profile/STS/fedserverconfig" />
</mdc-transform-rule>

```

Add -Doracle.oam.MDCRuleFile=/tmp/transformationRules.txt to \$MW_HOME/user_projects/domains/OAMDomain/bin/setDomainEnv.sh (on Linux on Unix) or \$MW_HOME\user_projects\domains\OAMDomain\bin\setDomainEnv.cmd (on Windows) on Data Center 2, Node 1 only and save the file.

6. Restart the Administration and Managed Servers.

This completes the Multi-Data Center configuration and APS configuration! You can test the APS function by creating an agent and a policy on data center 1 and verifying that it auto migrates to data center 2.

20.4 Troubleshooting the Multi-Data Center Setup

These troubleshooting tips may help you diagnose and fix some common problems encountered during MDC configuration.

This section analyzes the following issues:

- [When Authorization Header provided is correct, 'Error 401–Unauthorized' is displayed while executing the REST command](#)
- [Curl command returns curl: \(35\) SSL connect error](#)
- [APS Synchronization Failed With 401-UnAuthorized Error](#)
- [Fail to Decrypt oamkeystore Data with Cipher Key from OAM Config](#)
- [Modifying the Polling Interval in Clone Data Centers](#)
- [Overwriting the Existing MDC Configuration or Recovering from an Inconsistent State](#)
- [Changing Security Mode of Managed Servers in Working MDC Env](#)
- [MDC Session Adoption Issues in 11g/12c OAM Setup with SIMPLE mode Servers](#)
- [Request Failed When the Input Parameters Passed are Valid](#)
- [Modifying Session Control Parameters](#)
- [Modifying Backward Compatibility Flag](#)
- [Disabling MDC](#)
- [Backup Existing Artifacts in a Data Center](#)

20.4.1 Unauthorized Error Displayed When the Authorization Header is Correct

When Authorization Header provided is correct, 'Error 401–Unauthorized' is displayed while executing the REST command.

You may see this error when the WebLogic user or password do not match with the OAM Admin users or passwords and/or when OAM Admin users or passwords are different for the Master and Clone data centers. You can fix this by disabling WebLogic authentication and then verify the status of the `EnforceValidBasicAuthCredentials` parameter.

1. Run the following REST commands in both, the Master and Clone data centers to disable WebLogic authentication as shown in the following example:

```
connect('weblogicUser','weblogicPassword','t3://localhost:7001')
edit()
startEdit()
cd('SecurityConfiguration/Your_Domain')
set('EnforceValidBasicAuthCredentials','false')
save()
activate()
```

2. Restart the Master and Clone servers.

3. Verify the status of `EnforceValidBasicAuthCredentials` parameter.

```
connect('weblogicUser','weblogicPassword','t3://localhost:7001')
cd('SecurityConfiguration/Your_Domain')
ls()
```

4. From the list, confirm that the `EnforceValidBasicAuthCredentials` parameter is set to `false`.

20.4.2 Curl Command Returns Curl: (35) SSL Connect Error

You can fix the SSL connect error in two ways.

1. Verify your Curl version and update it to the latest version available.
2. Force specify the TLS version to 1.2 as follows:

```
curl --tlsv1.2 -k -u weblogic:password 'https://oamadmin1-
dc1.poc.com:7002/oam/services/rest/mdc/configuration'
```

3. Use any REST Client (available as extensions to web browsers).

20.4.3 APS Synchronization Failed With 401-UnAuthorized Error

When Admin Users and Password are different for Master and Clone data centers, the APS Synchronization fails. Update the Authorization Header in the replication agreement present in Clone data center to fix this issue.

Query the existing replication agreements using the following command and obtain the replication Identifier, `replId`

```
curl -k -u weblogic:welcome1 'https://oamadmin1-dc1.poc.com:7002/oam/
services/rest/_replication/agreements'
```

Note: If there are multiple identifiers, select the identifier for which the replication agreement needs to be updated by querying the corresponding Clone data center.

Run the following command with appropriate values to update the Authorization Header in the replication agreement of Clone data center.

```
curl -u weblogic:password -H 'Content-Type: application/json' -X PUT
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/"replId"'-
d '{"replicaType":"CONSUMER","config":{"entry":
{"key":"authorization","value":"authzvalue"}}}'
```

- `replId` : Identifier obtained from the above command.
- `authzvalue` : Enter the value of Authorization Header to be used for contacting the Master data center. Authorization Header will be Base 64 encoded value of `MasterAdminUser:MasterAdminPassword`.

For Example,

```
curl -u weblogic:password -H 'Content-Type: application/json' -X PUT
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/
201706200405204694' -d '{"replicaType":"CONSUMER","config":{"entry":
{"key":"authorization","value":Basic d2VibG9naWM6d2VsY29tZTE="}}}'
```

20.4.4 Fail to Decrypt oamkeystore Data with Cipher Key from OAM Config

After executing the REST API for setting up the clone, OAM server logs show exceptions due to internal synchronization of keys. Restarting the Clone data center sets all the required internal keys.

The following exception is shown in OAM server logs after setting up the Clone data center:

```
<Error> <oracle.oam.config>
<OAMSSA-08032> <Configuration event dispatch failed.
oracle.security.am.common.utilities.exception.AmRuntimeException:
Fail to decrypt oamkeystore data with cipher key from OAM config(/
DeployedComponent/Server/NGAMServer/Profile/ssoengine/CipherKey)
at
oracle.security.am.engines.sso.adapter.OAMSessionConfiguration$ConfigLi
stener.configurationChanged(OAMSessionConfiguration.java:295)
at
oracle.security.am.admin.config.BasicFileConfigurationStore$ListenerDis
patcher.run(BasicFileConfigurationStore.java:961)
Caused By: javax.crypto.BadPaddingException: Given final block not
properly padded
```

Setup clone REST API internally synchronizes some keys from the Master data center. After executing the REST API for reconfiguring Clone data center, restart the Clone and this exception is not shown in OAM server logs. You can safely ignore this exception in this scenario.

20.4.5 Modifying the Polling Interval in Clone Data Centers

The configuration changes made to the Master data center will take at least 900 Seconds to get propagated to the Clone data center as the default POLLINTERVAL is 900 Seconds. However, the polling interval can be modified, if required.

Run the following command to modify `pollInterval` parameter:

Query the existing replication agreements using the following command and obtain the replication Identifier, `replId`

```
curl -k -u weblogic:welcome1 'https://oamadmin1-dc1.poc.com:7002/oam/
services/rest/_replication/agreements'
```


Note: If there are multiple identifiers, select the identifier for which the replication agreement needs to be updated by querying the corresponding Clone data center.

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X PUT
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/replId' -
d '{"pollInterval":"value","replicaType":"CONSUMER"}
```

Where

replId is the Identifier obtained from the above command

PollInterval is the time in seconds for the Clone data center to send the query to the Master data center for the latest updates

For example,

```
curl -u weblogic:password -H 'Content-Type: application/json' -X PUT
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/_replication/replId' -
d '{"pollInterval":"60","replicaType":"CONSUMER"}
```

20.4.6 Overwriting the Existing MDC Configuration or Recovering from an Inconsistent State

Set the `forceOverWrite` parameter to overwrite the existing MDC configuration.

1. Disable APS, if configured.
See [Disabling APS](#).
2. Specify the `forceOverWrite` parameter as follows while configuring the Master data center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -
d '{"mdcTopologyType":"ACTIVE_ACTIVE","masterMDCAgentID":"MasterNAPAgent",
cloneMDCAgentID":"CloneNAPAgent1","accessClientPassword":"Welcome123","art
ifactPassword":"password","cloneServerURL":"https://oamadmin1-
dc2.poc.com:7002/","cloneAdminUserNamePassword":"weblogic:password","agent
KeyPassword":"password",
"certModeKeystorePassword":"password","forceOverWrite":"true"}
```

3. Specify the `forceOverWrite` parameter as follows while configuring the Clone data center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone' -d
'{"masterServerURL":"https://oamadmin1-dc1.poc.com:7002/",
"artifactPassword":"password","masterAdminUserNamePassword":"oamAdminUser:
password","forceOverWrite":"true"}
```

 **Note:**

After overwrite the existing MDC configuration, proceed with reconfiguring Clone data center and setting up replication agreement. See, [Setting Up a Multi-Data Center](#) and [Enabling Automated Policy Synchronization](#).

20.4.7 Changing the Security Mode of Managed Servers in Working MDC Environment

The secure communication mode of the servers can be changed to CERT or SIMPLE.

1. Disable APS. See [Disabling Automated Policy Synchronization](#)
2. Change the `WriteEnabled` flag to `true` in Clone data center using WLST commands.

```
connect('weblogic','password','t3://localhost:7001')
```

```
domainRuntime()
setMultiDataCenterWrite(WriteEnabledFlag="true")
```

3. Modify all the Managed Server instances to the required security mode in Master and Clone data centers.

In CERT Mode, follow the additional steps in Master:

- a. Setup OAM Servers in CERT mode. See
- b. Copy the MDC Partner Certificates generated to `%DOMAIN_HOME%/config/fmwconfig/oam-mdc-cert-artifacts/`. See

 **Note:**

Its not necessary to import the CERT mode certificates into `.oamkeystore` and to configure PEM KeyStore Alias and PEM KeyStore Alias Password in Clone data center.

- c. Run the REST commands to configure Master and Clone data centers with an additional parameter `forceOverWrite:true` in the request.

Example: While configuring the Master data center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X
POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/master' -
d
'{"mdcTopologyType":"ACTIVE_ACTIVE","masterMDCAgentID":"MasterNAPAgent",
cloneMDCAgentID":"CloneNAPAgent1","accessClientPassword":"Welcome123","ar
tifactPassword":"password","cloneServerURL":"https://oamadmin1-
dc2.poc.com:7002/","cloneAdminUserNamePassword":"weblogic:password","agen
tKeyPassword":"password",
"certModeKeystorePassword":"password","forceOverWrite":"true"}'
```

Example: While configuring Clone data center:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone' -d
'{"masterServerURL":"https://oamadmin1-dc1.poc.com:7002/",
```

```
"artifactPassword":"password","masterAdminUserNamePassword":"oamAdminUser:password","forceOverWrite":"true"}
```

- d. Re-configure the Clone data center command and Setup Replication Agreement command. See
- e. Restart Master and Clone Admin and Managed Servers.
- f. Verify SSO between Master and Clone data centers.

20.4.8 MDC Session Adoption Issues in 11g–12c OAM Setup with SIMPLE Mode Servers

When the OAM servers are configured in SIMPLE mode, MDC session adoption issues are encountered in an MDC environment where 11g Master and 12c Clone (or vice versa) data centers are deployed.

Relax the java security posture for compatibility if SSL communication fails between a 11g Master data center and a 12c Clone data center.

Perform the following on the machine that has the 12c OAM Server:

1. Locate and edit the file `$JAVA_HOME/jre/lib/security/java.security`.

```
key - jdk.tls.disabledAlgorithms - remove TLSv1, TLSv1.1, MD5withRSA
key - jdk.certpath.disabledAlgorithms - remove MD5
```

2. Restart Admin and Managed Servers.
3. Verify MDC Session adoption use-case.

20.4.9 Request Failed When the Input Parameters Passed are Valid

The Curl command fails and displays 'Request failed' error where the input parameters are valid. Verify the syntax and remove any white spaces in your input.

Following is the command that has the correct format with no white spaces before or after the parameters:

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc2.poc.com:7002/oam/services/rest/mdc/clone' -d
'{"masterServerURL":"https://oamadmin1-
dc1.poc.com:7002/", "artifactPassword":"password", "masterAdminUserNamePassword
":"oamAdminUser:password", "forceOverWrite":"true"}
```

20.4.10 Modifying Session Control Parameters

Modify and specify custom values for Session control parameters.

Run the following command separately in the Master and Clone data centers. When APS is enabled, changes made to the MDC Configuration using this commands will not be propagated to Clone data center(s).

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/dc/mode' -d
'{"config":{"entry":
[{"key":"SessionMustBeAnchoredToDataCenterServicingUser", "value":<<true (for
```

```

Invalidate) or false (for No Invalidation)>>},
{"key":"SessionDataRetrievalOnDemand","value":<<true (for Cross DC
Retrieval) or false (for No Cross DC Retrieval)>>},
{"key":"SessionContinuationOnSyncFailure","value":<<true (for
Invalidation/Retrieval should succeed) or false (for Ignore
failure)>>}, {"key":"Reauthenticate","value":<<true (for Force
Reauthentication) or false (for No Reauthentication)>>},
{"key":"SessionDataRetrievalOnDemandMax_retry_attempts","value":<<the
value equal to the binary that represents the number of attempts for
data retrieval when it fails. DEFAULT: 2>>},
{"key":"SessionDataRetrievalOnDemandMax_conn_wait_time","value":<<the
value equal to the binary that represents the total amount of time in
seconds to wait for a connection. DEFAULT: 1000>>},
{"key":"MDCGitoCookieDomain","value":<<the domain in which OAM_GITO
cookie should be set. OPTIONAL: Set it in MDC Deployments where a
common domain hierarchy can be derived>>}}}]}'

```

For example,

```

curl -k -u weblogic:password -H 'Content-Type: application/json' -X
POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/dc/
mode' -d '{"config":{"entry":
[{"key":"SessionMustBeAnchoredToDataCenterServicingUser","value":"true"
}, {"key":"SessionDataRetrievalOnDemand","value":"true"},
{"key":"SessionContinuationOnSyncFailure","value":"true"},
{"key":"Reauthenticate","value":"true"},
{"key":"SessionDataRetrievalOnDemandMax_retry_attempts","value":"3"},
{"key":"SessionDataRetrievalOnDemandMax_conn_wait_time","value":"80"}]}
}'

```

20.4.11 Modifying Backward Compatibility Flag

Set `isBackwardCompatible` parameter to `true` to enable or `false` to disable backward compatibility in a data center. Use this parameter only when the Master and Clone data centers are running in different versions of OAM.

Note:

If all the data center are using 12.2.1.4.0 binary, this flag should not be enabled.

Run the following command separately in the Master and Clone data centers. When APS is enabled, changes made to the MDC Configuration using this commands will not be propagated to Clone data center(s).

```

curl -k -u weblogic:password -H 'Content-Type: application/json' -X
POST 'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/dc/
compatibility' -d '{"isBackwardCompatible":<<"true" to enable or
"false" to disable backward compatibility if Master and Clone DCs are
running different versions of OAM such as 11g and 12c respectively>>}'

```

For example,

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/dc/compatibility' -
d '{"isBackwardCompatible":"true|false"}
```

20.4.12 Disabling MDC

Set the `isMultiDataCenterEnabled` to `false` to disable MDC.

When APS is enabled, changes made to the MDC Configuration using this commands will not be propagated to Clone data center(s). Run the command separately in the Master and Clone data centers.

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/dc/mode' -d
'{"isMultiDataCenterEnabled":"false"}
```



Note:

Do not set `isMultiDataCenterEnabled` to `true` as its not supported. See [Modifying Session Control Parameters](#).

20.4.13 Backup Existing Artifacts in a Data Center

Specify appropriate values for `artifactPassword` and `artifactsZipLocation` to take a backup of existing artifacts in the data center.

When APS is enabled, changes made to the MDC Configuration using this commands will not be propagated to Clone data center(s). Run the command separately in the Master and Clone data centers.

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/dc/backup' -d
'{"artifactPassword":<<password used for protecting the cloning
artifacts>>,"artifactsZipLocation":<<Location where Artifacts has to be
stored. (OPTIONAL: specify if the Artifacts need to be stored in any
location other than /tmp)>>}'
```

For example,

```
curl -k -u weblogic:password -H 'Content-Type: application/json' -X POST
'https://oamadmin1-dc1.poc.com:7002/oam/services/rest/mdc/dc/backup' -d
'{"artifactPassword":"password","artifactsZipLocation":"/scratch"}
```

Part VI

Managing Access Manager SSO, Policies, and Testing

Administrators manage single-sign on (SSO) with Access Manager, configure Access Manager policies, test connectivity and policies, and configure centralized session logout.

Testing your single sign-on connection and policies is also described. This section contains the following chapters:

- [Understanding Single Sign-On with Access Manager](#)
- [Managing Authentication and Shared Policy Components](#)
- [Understanding Credential Collection and Login](#)
- [Using Password Policy](#)
- [Managing Policies to Protect Resources and Enable SSO](#)
- [Validating Connectivity and Policies Using the Access Tester](#)
- [Configuring Centralized Logout for Sessions Involving OAM WebGates](#)

Using Passwordless Authentication with OAM

This section provides an overview of passwordless authentication and its configuration in OAM.

- [About Passwordless Login](#)
- [Configuring Passwordless Login with OAM](#)

About Passwordless Login

OAM provides passwordless authentication, which allows you to bypass the standard web-form-based authentication when using a mobile device. Passwordless authentication allows access to the protected resource without the need for entering the username and password everytime. However, the first time login is through the standard login form.

During the first time while accessing the protected resource, you are redirected to the standard login form. After successful login, you can enable passwordless notification-based authentication.

The next time (and subsequently) when you access the protected page and are required to login, a message is displayed (instead of the standard login page) mentioning that a push notification is sent to your mobile device. To authenticate, you must open the Oracle Mobile Authenticator (OMA) app on your registered mobile device and allow access. You are then redirected to the protected page.

OMA is a mobile app and must be installed on the mobile device, and registered with OAM. For more information about OMA, see [Configuring the Oracle Mobile Authenticator](#).

OAM provides Adaptive Authentication Plugin, Passwordless Plugin, module, and scheme for configuring passwordless login. See [Configuring Passwordless Login with OAM](#).

You can customize the following pages for the Passwordless Scheme and Second Factor Authentication (SFA) using the custom pages framework:

- Login Page
- Challenge Page
- Challenge Choice Page
- Challenge Answer Page
- Waiting Page (Intermediate Page)

See, Developing Custom Pages for details.

Configuring Passwordless Login with OAM

Passwordless authentication allows you to bypass the standard web-form-based authentication when using a mobile device. You can configure passwordless authentication using OAM.

This section provides steps to configure passwordless login:

1. [Enabling Adaptive Authentication Service](#)
2. [Configuring the Adaptive Authentication Service Plugin](#) to support Push Notification
3. [Setting Credentials for iOS and Android](#) and configuring certificates for Apple push notification service and OMA.
4. [Configuring Passwordless Authentication Module and Scheme](#)
5. [Protecting Resources with Passwordless Scheme](#)

Enabling Adaptive Authentication Service

Adaptive Authentication Service must be enabled for the features, such as, passwordless login to work.

To enable the Adaptive Authentication Service

1. Log in to the Oracle Access Management Console

```
https://hostname:port/oamconsole/
```
2. From the Welcome page, click **Configuration** and then click **Available Services**
3. Under **Application Security**, click **Enable Service** beside the Adaptive Authentication Service (or confirm that the green Status check mark displays).
A Confirmation window is displayed.
4. Click **Enable Service**.

Configuring Passwordless Authentication Module and Scheme

Configure the parameters for the passwordless authentication module and scheme. Passwordless authentication plugin is available as part of the OAM installation and is not required to be configured separately.

This section provides details about the parameters that needs to be configured in the passwordless authentication module and scheme.

- [Passwordless Authentication Module](#)
- [Passwordless Authentication Scheme](#)

Passwordless Authentication Module

Passwordless authentication module provides UserIdentificationPlugin, UserAuthenticationPlugin, and PasswordlessPlugin as individual steps in the passwordless authentication process.

To configure the Passwordless authentication module in the Oracle Access Management Console:

1. Log into the Oracle Access Management Console as System Administrator.
2. From the Application Security Launch Pad, click **Authentication Modules** under **Plugins**.
3. From the Authentication Modules tab, search for **PasswordlessModule**.
4. Update Passwordless authentication module properties as follows:

Table 33 UserIdentification Step

Step Details	Description
KEY_IDENTITY_STORE_REF	The name of the registered Identity Store containing the module users. Default: The registered Default Store.
KEY_LDAP_FILTER	Add the LDAP filter to the KEY_LDAP_FILTER attribute. Only standard LDAP attributes can be used when defining an LDAP search filter. For example: (uid={KEY_USERNAME})
KEY_SEARCH_BASE_URL	Base URL for user searches. For example: dc=us,dc=example,dc=com

Table 34 UserAuthentication Step

Step Details	Description
KEY_PROP_AUTHN_EXCEPTION	Enable or disable the propagation of LDAP errors.
KEY_IDENTITY_STORE_REF	The name of the registered Identity Store containing the module users. Default: The registered Default Store.

Table 34 (Cont.) UserAuthentication Step

Step Details	Description
KEY_ENABLE_AUTHN_FAILOVER	If the parameter is <code>false</code> , the <code>userpassword</code> is removed from the authentication context so that it cannot be used in subsequent plugins. If set to <code>true</code> , the password can be used in subsequent plugins for further user authentication. Default: <code>false</code>
KEY_PROP_AUTHN_LEVEL	When set to a particular value, that value is set as the <code>OAMAuthnLevelPrincipal</code> in the subject of the authenticated user.

Table 35 Passwordless Step

Step Details	Description
IdentityStoreRef	The name of the registered Identity Store containing the module users. Default value is the registered Default Store.
PushProxyProtocol	Proxy protocol.
PushProxyHost	Name of the proxy host if notifications are sent to the server using a proxy.
PushProxyPort	Proxy port if notifications are sent to the server using a proxy.
PushTitleMsg	Title for the message sent to the user's OMA application on user's device.
PushExpiryTimeMs	Time after which the push notification is considered expired, from the time the push notification is sent to the server. Default value is <code>60000</code> , in milliseconds.
PushAPNsProdServer	If set to <code>true</code> , the APNS production server is used to send notifications. Default value is <code>false</code> .
URL_ACTION	The type of servlet action needed for redirecting the user to the specific password page for expiry and warning pages. Value can be one of the following: <ul style="list-style-type: none"> • <code>REDIRECT_POST</code> • <code>REDIRECT_GET</code> • <code>FORWARD</code> Default is <code>FORWARD</code> .
URL_REDIRECT	The url to be redirected to, for showing the user the passwordless waiting page. This field is required for the passwordless pages customizations. Default value is <code>/oam/pages/passwordless.jsp</code>

Table 35 (Cont.) Passwordless Step

Step Details	Description
PasswordlessGroup	<p>The passwordless feature can be restricted to certain groups of users. This field can be used to restrict only certain users to be able to use the passwordless feature.</p> <p>Multiple groups can be specified separated using , (comma- the default value for the PasswordlessGroupDelimiter) or any PasswordlessGroupDelimiter as configured.</p> <p>Default is empty. It means that all the users in the configured idstore have passwordless feature enabled.</p>
PasswordlessGroupDelimiter	<p>The delimiter used to specify multiple groups for PasswordlessGroups.</p> <p>Default value is comma (,).</p>
DenyDisabledLockedUsers	<p>When this value is set to true, users who are in disabled and locked states cannot use passwordless feature.</p> <p>These users will be moved to password-based authentication always.</p> <p>Default value is false.</p> <p>This is required when a user has been using passwordless-based authentication and is suddenly disabled using OAM password management module, because of which authentication is denied for that user.</p>
CookieMaxAge	<p>The maximum age of the cookie related to passwordless related cookie.</p> <p>After this time (since last passwordless login), the user is required to login again with the password.</p> <p>Default value is 8640000, in seconds.</p>
CookieName	<p>Name of the passwordless cookie that gets set for passwordless authentication.</p> <p>Default value is ORA_OAM_BTHGOES</p>
CookieValidateBrowserFP	<p>Controls whether the passwordless cookie is validated for the browser's user agent string.</p> <p>Default value is true.</p>
CookieValidateIpAddress	<p>Specifies whether the passwordless cookie is validated for the ipaddress where it was initially set.</p> <p>Default value is true.</p>
CookieIssuedExpiration	<p>Time, after which, when the cookie is configured as invalid, from the time the cookie is considered expired.</p> <p>Default value is 7776000000, in milliseconds.</p>

Table 35 (Cont.) Passwordless Step

Step Details	Description
CookieRefreshExpiration	<p>Time between, when the cookie is refreshed and then the time when cookie is considered expired.</p> <p>Default value is 864000000, in milliseconds.</p> <p>This value is set as part of the cookie value, and cannot be changed from the browser's side.</p>
CookieDomain	<p>The domain, which is used while constructing the passwordless cookie. This is used for extra security.</p> <p>Default value is Empty String.</p>
PushPurgeExpiredRequest	<p>Allows the server to purge the expired push notification requests so as to maintain the storage for push notifications and not have expired requests that are not required.</p> <p>Default value is <i>false</i>.</p>
PushPollTimeMs	<p>The time that the browser uses to wait till it again sends a request to OAM server to check if the notification has been acted upon.</p>
HandleFailedCounter	<p>Enables the server to handle notifications that are rejected consistently, for added security. When enabled, the OAM server keeps track of notifications that have been rejected and if it is more than <i>MaxFailureCounter</i>, then the user is forced to do a normal password based authentication.</p> <p>Default value is <i>false</i>.</p>
EnableFailedCounterOnExpiry	<p>Optional. When the notification goes unanswered, a failure counter can be enabled. Once enabled, the counter keeps track on number of times the notification has gone unanswered.</p> <p>When it has gone unanswered more than the <i>MaxFailureCounter</i>, then the user needs to do a authentication using their actual password.</p> <p>Default value is <i>false</i>.</p>
MaxFailureCounter	<p>Maximum number of times the notification can be failed or unanswered, before which a password based authentication is re-triggered.</p> <p>Default value is 5.</p>
ForceBiometric	Not Used.

Passwordless Authentication Scheme

Set the challenge parameters in the passwordless authentication scheme to bypass the login form.

The passwordless authentication scheme is available with all the necessary configurations required for the passwordless login to work. There are no additional configurations required.

However, to customize the passwordless waiting page and the first-page-before-authentication, you must update the following parameters of the passwordless authentication scheme:

- Challenge URL
- Context value
- Context Type

Ensure the `initial_command` is set to `NONE` under the challenge parameters to bypass the login form.

For details on all the parameters in the authentication scheme, see [Authentication Schemes and Pages](#)

Protecting Resources with Passwordless Scheme

Complete the configuration for passwordless login by assigning the passwordless authentication scheme to the protected resource policy.

1. From the **Application Security** Launch Pad, select **Application Domains** under **Access Manager**.
2. Search and open the required **Application Domain**.
3. Open the **Authentication Policy** tab and click **Protected Resource Policy**.
4. Select the **PasswordlessScheme** from the **Authentication Scheme** dropdown list and click **Apply**.

21

Understanding Single Sign-On with Access Manager

You can familiarize yourself with the elements that comprise Access Manager single sign-on (SSO). A SSO provides an administrator with the foundation to begin developing policies. The following topics provide information about:

- [Access Manager Single Sign-On Components](#)
- [Access Manager Policy Model](#)
- [Anatomy of an Application Domain and Policies](#)
- [Policy Conditions and Rules](#)
- [Understanding SSO Cookies](#)
- [Configuring Single Sign-On with Access Manager](#)



Note:

Unless explicitly stated, information in this chapter is the same for all agent types and Access Manager credential collectors.

See [Introduction to Centralized Logout for Access Manager](#).

21.1 Access Manager Single Sign-On Components

Login is the action a user takes to authenticate and gain access to a protected application. Single sign-on (SSO) is the process that gives users the ability to access multiple protected resources (Web pages and applications) with a single authentication. SSO is enabled by Access Manager to eliminate the need for additional or different logins to access other applications at the same (or lower) authentication level during the same session. Access Manager converges several SSO architectures (including Identity Federation for Partner Networks, and Service Oriented Architecture) and provides SSO through a common SSO Engine for consistent service across multiple protocols. The Oracle Identity Management Infrastructure stores user identities in the identity store referenced in the policy.

 **Note:**

Contextual data is the information that is presented to or collected by Access Manager at various stages of user interaction. These stages include authentication, authorization, enterprise SSO, federation, adaptive authentication, token validation, session creation, and so on. The information itself might comprise a user's device fingerprints, IP address, antivirus and firewall protection, assertion and so on. Components that play the role of contextual data providers and asserters when integrated with Access Manager include Enterprise Single Sign-on, Identity Federation, Oracle Adaptive Access Manager.

Table 21-1 summarizes the components that support or enforce Access Manager policies, and where to find more information about these, if needed.

 **Note:**

Default Access Manager behavior is to deny access when a resource is not protected by a policy that explicitly allows access. To delegate authentication tasks to Access Manager, agents must reside with the relying parties and must be registered with Access Manager. Registering an agent sets up the required trust mechanism between the agent and Access Manager SSO.

Table 21-1 Summary: SSO Components

Component	Description
Applications	<p>Applications can delegate authentication and authorization to Access Manager and accepts headers from a registered Agent.</p> <p>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.</p>
<ul style="list-style-type: none"> OAM Server Oracle Access Management Console (installed on WebLogic AdminServer) 	<p>Non-administrative users first gain access by entering the URL of a protected resource, which returns the SSO login page.</p> <p>See Also: Understanding Credential Collection and Login .</p> <p>Administrative users access the console to author policies by typing the URL: <code>https://host:port/oamconsole</code>. Although, default policies can be generated automatically during Agent registration, as described in Registering and Managing OAM Agents.</p> <p>See Also: Managing Policies to Protect Resources and Enable SSO.</p>
Policy Enforcement Agents	<p>OAM Agents (Webgate or Access Client)</p> <p>See Also: Introduction to Agents and Registration.</p>
Credential Collectors and Communication Channels	<ul style="list-style-type: none"> Authentication with the default embedded credential collector (ECC) occurs across the HTTP (HTTPS) channel Authentication with the optional detached credential collector (DCC) occurs across the Oracle Access Protocol (OAP) channel Authorization occurs across the Oracle Access Protocol (OAP) channel <p>See Also: Table 22-4</p>

Table 21-1 (Cont.) Summary: SSO Components

Component	Description
SSO Engine	Manages the session lifecycle, facilitates global logout across all relying parties in the valid session, and provides consistent service across multiple protocols. See Also: Maintaining Access Manager Sessions and Configuring Centralized Logout for Sessions Involving OAM WebGates
Proxy support for legacy systems	<ul style="list-style-type: none"> OAM Proxy supports legacy Access Manager implementations by acting as a legacy Access Server. See: <ul style="list-style-type: none"> Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security OAM Proxy Metrics and Tuning
Access Policies	Registered agents rely on Access Manager authentication, authorization, and token issuance policies to determine who gets access to protected applications (defined resources). Note: Default Access Manager behavior is to deny access when a resource is not protected by a policy that explicitly allows access. See Also: Managing Policies to Protect Resources and Enable SSO
Policy Store	Database in production environments (otherwise, oam-config.xml). See Also: Managing Data Sources
Cryptographic keys and Key Storage	One key is generated and used per registered OAM Webgate. See Also: #unique_372/unique_372_Connect_42_BHCGCCJG .
Cookies	See: Understanding SSO Cookies .

Single sign-on can be implemented as introduced in [Table 21-2](#), which includes pointers to additional information.

Table 21-2 Introduction to SSO Implementations

SSO Type	Description
Single Network Domain SSO	You can set up Access Manager single sign-on for resources within a single network domain (<i>example.com</i> , for example). This includes protecting resources belonging to multiple WebLogic administration domains within a single network domain. Single Network Domain SSO is the subject of this book.
Multiple Network Domain SSO	Access Manager 12c supports cross-network-domain single sign-on out of the box. See Also: Multiple Network Domain SSO .
Application SSO	Application single sign-on allows users who have been authenticated by Access Manager to access applications without being re-authenticated. See Also: Application SSO and Access Manager
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. See Also: Multiple WebLogic Server Domain SSO

Table 21-2 (Cont.) Introduction to SSO Implementations

SSO Type	Description
Reverse-Proxy SSO	This SSO implementation type is supported with a few configuration differences. See Also: Reverse-Proxy SSO
SSO with Mixed Release Agents	Access Manager seamlessly supports registered 12c OAM agents (Webgates and programmatic access clients).

21.1.1 Multiple Network Domain SSO

With Access Manager, this is a standard feature. When OAM 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 Access Manager deployment), Oracle recommends that you use Identity Federation.

Access Manager supports cross-network-domain single sign-on out of the box. During single sign-off with Access Manager:

- The SSO cookie set by 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 Server for session clearing.



See Also:

[Configuring Centralized Logout for OAM WebGates](#)

21.1.2 Application SSO and Access Manager

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 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](#)".

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 Access Manager 12c. If the policy includes a redirect URL that is hosted by a Web server not protected by Access Manager, header responses are not applied.

For example, when a user authenticates, she might be redirected to a portal index page:

```
http://example.com/authnsuccess.htm
```

For authentication failure, an authentication action might redirect the user to an error page or a self-registration script:

```
http://example.com/authnfail.htm
```

21.1.3 Multiple WebLogic Server Domain SSO

Access Manager 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 Management 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.

21.1.4 Reverse-Proxy SSO

Reverse-Proxy SSO is a supported configuration.

Following are certain caveats associated with this configuration:

Caveats

If you are going to use a reverse proxy in a single sign-on configuration, be sure to perform one of the following tasks. Otherwise, the reverse proxy hides the client's IP address:

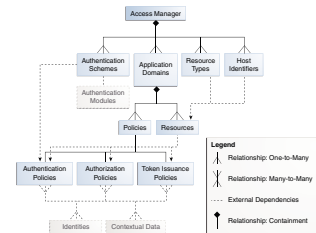
- Either to set the `IPvalidation` parameter to `false`
- Or add the proxy IP address to the `IPValidationExceptions` list in the Webgate registration

21.2 Access Manager Policy Model

Access Manager distills the policy models of Oracle Access Manager into a single Access Manager policy model.

[Figure 21-1](#) illustrates the main elements of the Access Manager 12c policy model including the shared policy components, an individual Application Domain, and external dependencies.

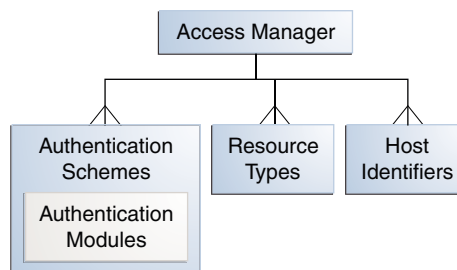
Figure 21-1 Access Manager 12c Policy Model



Shared Policy Components

Shared policy components are global and can be used in one or more Application Domains. [Figure 21-2](#) illustrates the shared components for Access Manager policies.

Figure 21-2 Access Manager Shared Policy Components



[Table 21-3](#) describes the global, shared components in an Access Manager policy.

Table 21-3 Access Manager Global, Shared Policy Components

Component	Description
Resource Types	<p>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.</p> <p>Any number of resources can belong to a specific resource type. However, each resource that is added to a policy must be defined as a single type:</p> <ul style="list-style-type: none"> • HTTP • wl_authen • TokenServiceRP <p>See Also:</p> <ul style="list-style-type: none"> • Managing Authentication and Shared Policy Components : Managing Resource Types
Host Identifiers	<p>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. With Access Manager, all possible host 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.</p> <p>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.</p> <p>Host identifiers can be generated automatically during Agent registration and are used to seed the Resource definition and default authentication and authorization policies in the new Application Domain.</p> <p>Alternatively: Administrators can create a host identifier definition for use in one or more Application Domains.</p> <p>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.</p> <p>See Also: "About Host Identifiers".</p>

Table 21-3 (Cont.) Access Manager Global, Shared Policy Components

Component	Description
Authentication Scheme	<p>A named component that defines the challenge mechanism, level of trust, and the underlying authentication module or plug-in required to authenticate a user. Several default schemes provided with Access Manager and Administrators can define their own schemes.</p> <p>Authenticating a user's identity with Access Manager refers to running a pre-defined set of processes to verify the digital identity of the user. One authentication scheme can be assigned to multiple authentication policies. However, each authentication policy can have only one authentication scheme assigned to it.</p> <p>Note: Authentication schemes are defined globally to ensure that a small number of Administrators define them in a consistent, secure way.</p> <p>See Also: "Managing Authentication Schemes"</p>
Authentication Modules and Plug-ins	<p>The smallest executable unit of an authentication scheme. The authentication module determines the exact procedure to be followed and the method for challenging the user for credentials.</p> <p>Authentication involves determining which credentials a user must supply when requesting access to a resource, gathering credentials, and returning a response that is based on the results of credential validation.</p> <p>All authentication processing relies on an authentication module to define the rules governing requirements and transmission of information to the backend authentication scheme. All information collected by the plug-in and saved in the context is available to the plug-in through the authentication process. Context data can also be used to set cookies or headers in the user's login page</p> <p>A number of plug-ins and several pre-defined modules are provided. Oracle strongly recommends using plug-ins, which you can configure and orchestrate as needed to provide multi-step authentication.</p> <p>See Also:</p> <ul style="list-style-type: none"> • "Managing Native Authentication Modules" • "Orchestrating Multi-Step Authentication with Plug-in Based Modules "

Access Manager Policy Components

Access Manager default behavior denies access when a resource is not protected by a policy that explicitly allows access. [Table 21-4](#) describes policy components you configure to allow access and where you can find the details.



See Also:

["Anatomy of an Application Domain and Policies"](#)

Table 21-4 Access Manager Policy Components

Component	Description
Application Domain	<p>Each Application Domain provides a logical container for resources, and the associated policies that dictate who can access these resources. An application domain can be created automatically during Agent registration or manually using the console.</p> <p>See Also: "Anatomy of an Application Domain and Policies"</p>
Resource Definitions	<p>Based on a defined host identifier, Administrators can add specific resources to an Application Domain and apply policies to protect those resources.</p> <p>See Also: "Adding and Managing Policy Resource Definitions".</p>
Authentication Policy	<p>Each resource defined in an Application Domain can be protected by only one authentication policy. Each authentication policy requires one authentication scheme.</p> <p>One authentication policy can protect many resources. However, each resource can be protected by only one authentication policy.</p> <p>See Also: "Defining Authentication Policies for Specific Resources"</p>
Authorization Policies	<p>Each resource assigned to an Application Domain can be protected by only one authorization policy. Each policy can include one or more conditions and a rule. Authorization policies can also contain success responses.</p> <p>One authorization policy can protect many resources. However, each resource can be protected by only one authorization policy.</p> <p>See Also: "Defining Authorization Policies for Specific Resources".</p>
Token Issuance Policy	<p>By default, only a container for Token Issuance Policies is provided in a generated Application Domain. No Conditions or Rules are generated automatically. You must add these manually.</p> <p>See Also: "Token Issuance Policy Pages".</p>
Policy Responses	<p>Available for all policy types, Authentication and Authorization success Responses can be defined within respective policies to be applied after policy evaluation.</p> <p>See Also: "Introduction to Policy Responses for SSO".</p>
Rule	<p>Available for only Authorization and Token Issuance Policies.</p> <p>Each Authorization policy includes a rule that defines whether the policy allows or denies access to resources protected by the policy.</p> <p>The rule references Authorization conditions, described next.</p> <p>See Also: "Introduction to Authorization Policy Rules and Conditions".</p>
Condition	<p>Available for only Authorization and Token Issuance Policies.</p> <p>Each Authorization policy rule references conditions that define to whom the rule applies, if there is a time Condition, and how evaluation outcomes are to be applied.</p> <p>Conditions are declared outside of rules and are referenced within a rule.</p> <p>See Also: "Introduction to Authorization Policy Rules and Conditions".</p>

21.3 Anatomy of an Application Domain and Policies

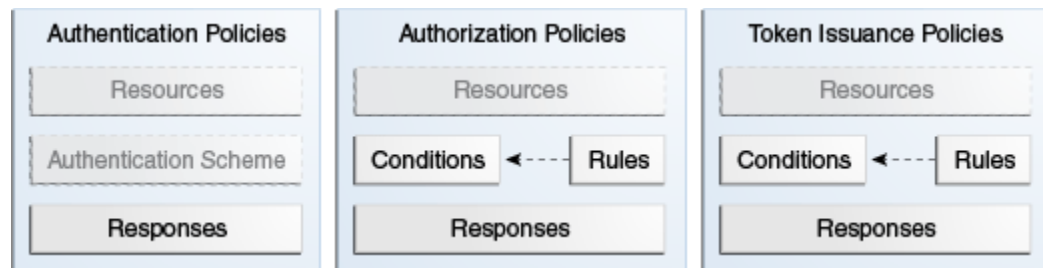
Access Manager enables you to control who can access resources based on policies defined within an Application Domain. Users attempt to access a protected resource by entering a URL in a browser, by running an application, or by calling some other external business logic. When a user requests access to a protected resource, the

request is evaluated according to policies that discriminate between authenticated users who are authorized and those who are not authorized for access to a particular resource. Application domains do not have any hierarchical relationship to one another. 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. Within each Application Domain, specific resources are identified for protection by specific policies that govern access. Authentication and authorization policies include Administrator-configured responses that are applied upon successful evaluation. Authorization policies include Administrator-configured conditions and rules that define how evaluation is performed, and responses to be applied upon successful evaluation.

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.

Figure 21-3 shows an expanded view of policies within an Application Domain, as well as how the shared elements are used in an Application Domain.

Figure 21-3 Anatomy of Access Manager Policies



For more information, see the following topics:

- [Resource Definitions for Policies](#)
- [About Authentication Policies](#)
- [About Authorization Policies](#)
- [About Token Issuance Policies](#)

21.3.1 Resource Definitions for Policies

The term *resource* represents a document, or entity, or pieces of content stored on an OAM Server and available for access by a large audience.

Clients communicate with the OAM Server to request a resource using a particular protocol (HTTP or HTTPS, for example), which corresponds to an existing Resource Type. Every HTTP Resource Type must be associated with a host identifier. However, non-HTTP Resource Types are associated with a specific name (not a host identifier).

With Access Manager, each resource must be defined as within the Resources container in an Application Domain before it can be associated with a specific policy.



Note:

Only resources defined in the Resources container can be associated with policies in the Application Domain.

For more information, see "[Adding and Managing Policy Resource Definitions](#)".



Note:

To protect pieces of content on a page, Oracle recommends using Oracle Entitlements Server.

21.3.2 About Authentication Policies

Administrators can create an authentication policy to apply to specific resources within an Application Domain.

Each authentication policy:

- Identifies the specific resources covered by this policy, which must be defined on the Resources tab of this policy and in the Resources container for the Application Domain
- Specifies the authentication scheme that provides the challenge method to be used to authenticate the user
- Specifies the Success URL (and the failure URL) that redirects the user based on the results of this policy evaluation
- Defines optional Responses that identify post-authentication actions to be carried out by the Agent.

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 Oracle Access Manager 12c, which provided data passage to (and between) applications by redirecting to URLs in a specific sequence.

Policy responses are optional. These must be configured by an Administrator and are applied to specific resources defined within the Application Domain. For more information, see "[Introduction to Policy Responses for SSO](#)".

Authentication Policy Evaluation Results

To authenticate a user, Access Manager presents the user's browser with a request for authentication credentials based on the challenge method defined by the authentication scheme for this policy.

After policy evaluation, the result is returned and the user is redirected based on that result:

- Success (allow access) redirects to the requested URL
- Failure, (deny access) redirects to a generic error page

 **Note:**

Policy evaluation results can be overridden policy by policy.

 **See Also:**

- ["Authentication Policy Pages"](#)
- ["Managing Run Time Policy Evaluation Caches"](#)

21.3.3 About Authorization Policies

Authorization is the process of determining if a user has a right to access a requested resource.

A user might want to see data or run an application program protected by a policy, for example.

Administrators can create an authorization policy to specify the conditions under which a subject or identity has access to a particular resource. The requested resource must belong to an Application Domain and must be included within a specific authorization policy.

Each authorization policy:

- Identifies the specific resources covered by this policy, which must be defined on the Resources tab of this policy and in the Resources container for the Application Domain
- Specifies the Success URL (and the failure URL) that redirects the user based on the results of this policy evaluation
- Identifies specific Allow or Deny Rules based on defined conditions for this policy and resources. See [Table 21-5](#) for an overview of Condition types.
- Defines optional Responses that identify post-authorization actions to be carried out by the Agent, as described in [Introduction to Policy Responses for SSO](#).

 **See Also:**

[Policy Conditions and Rules](#)

21.3.4 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. The client can be either a Requester Partner or an end user.

Unless explicitly stated, information on Application Domains and authorization policies applies equally to Token Issuance policies.

**Note:**

During automatic policy generation, no Token Issuance Policies are created; only the container for Token Issuance Policies is generated automatically.

21.4 Policy Conditions and Rules

Conditions can be specified only within Authorization and Token Issuance policies. Rules define the Allow or Deny specification that determines the overall effect of the policy.

Unless explicitly stated, information on policy Conditions and Rules applies equally to authorization and Token Issuance policies.

Conditions

Conditions are used in conjunction with Rules that specify Allow or Deny access, based on defined Conditions. [Table 21-5](#) identifies available condition types.

Table 21-5 Condition Types

Type	For more information, see ...
Identity	"Introduction to Authorization Policy Rules and Conditions" .
IP4 Range	"Defining IP4 Range Conditions" .
Temporal	"Defining Temporal Conditions" .
Attribute	"Defining Attribute Conditions" .
True	Effectively "Allow All". Oracle recommends this be used as the default option in cases where you need to let in any authenticated use. In this case, you do not need any particular conditions to be satisfied at authorization time. This replaces the Use Implied Constraints flag the previous release of Access Manager, which similarly lets policy evaluation complete with an Allow result when no specifically-defined constraints were present.

Each Authorization and Token Issuance policy can contain one or more condition objects. There can be more than one instance of a type of condition in a policy (the previous policy model allowed only one instance of a class in a policy).

Conditions are similar to earlier Access Manager 12c authorization constraints. However, constraints included Allow or Deny specifications and conditions do not.

Rules

Rules are new constructs in the policy model. Each Rule defines the Allow or Deny specification that determines the overall effect of the policy. Rules also define how the outcomes of each Condition evaluation is to be combined. Conditions are referenced in rules and declared outside of rules.

Within a Rule, evaluation outcomes can be combined as follows:

- **Simple Mode:** Accepts a list of condition names that are combined based on the value of a combiner that allows either All conditions to be met or Any one condition

to be met to return "true" for the evaluation. [Previously, ALL allowed constraints while ANY denied them.]

- **Expression mode:** Allows the user to specify a Boolean expression to combine conditions using condition names and special characters (comma, vertical bar, ampersand and exclamation point: , |& and !).



Note:

A policy in which there are one or more conditions that are not part of either an Allow or Deny Rule is treated as a valid policy.

For more information about Conditions and Rule, see [Managing Policies to Protect Resources and Enable SSO](#).

21.5 Understanding SSO Cookies

SSO cookies are set or cleared during user login.

The following sections provide more information on SSO cookies:

- [Single Sign-On Cookies During User Login](#)
- [Single Sign-On Server and Agent Cookies](#)
- [Support for SameSite=None Attribute in OAM Cookies](#)

21.5.1 Single Sign-On Cookies During User Login

Single Sign-On Cookies can be set or cleared during user login.

[Table 21-6](#) describes the cookies in detail.

Table 21-6 SSO Cookies

SSO Cookie Set at User Login	Set By	Description
OAM_ID cookie	OAM Server Embedded Credential Collector	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. See Also: " OAM_ID cookie ".
OAMAuthnCookie	OAMWebgate	Set by each OAM Webgate that is contacted. Protected by the key known to the respective OAM Webgate and the OAM Server. A valid OAMAuthnCookie is required for a session. Note: If the user accesses applications protected by different OAM Webgates, you will have multiple OAMAuthnCookies. See " OAMAuthnCookie for OAM Webgates ".

Table 21-6 (Cont.) SSO Cookies

SSO Cookie Set at User Login	Set By	Description
OAM_REQ	OAM Server Embedded Credential Collector	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. See " OAM_REQ Cookie ".
OAMRequestContext	OAM Webgate	Set or cleared by the OAMWebgate and protected by the key known to the respective OAMWebgate and the OAM Server. With Internet Explorer browser: --When RequestContextCookieExpTime is not set, OAMRequestContext is a transient cookie. --When RequestContextCookieExpTime is set, the OAMRequestContext cookie expires by the time set using the "Expires" directive. This requires a time sync between the client host and Web server host. With all other (non-IE) browsers, when RequestContextCookieExpTime is not set OAMRequestContext expires in 5 minutes by default or by the time set using the "Max-Age" directive. See Also: " OAMRequestContext " Table 15-2
DCCCtxCookie	Detached Credential Collector	For detached credential collector (DCC)--similar to OAM_REQ created by embedded credential collector (ECC). See " DCCCtxCookie "

For details about configuring authentication and authorization policies, see [Managing Policies to Protect Resources and Enable SSO](#).

21.5.2 Single Sign-On Server and Agent Cookies

The following sections provide information on the SSO server and agent cookies.

- [OAM_ID cookie](#)
- [OAMAuthnCookie for OAM Webgates](#)
- [OAM_REQ Cookie](#)
- [OAMRequestContext](#)
- [DCCCtxCookie](#)
- [DCCCtxCookie_COUNT](#)

21.5.2.1 OAM_ID cookie

The OAM_ID cookie is scoped to the OAM Server. OAM_ID is generated by the OAM Server when the user is challenged for credentials, and submitted to the server on

every redirect to the server. OAM_ID is protected by 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.

21.5.2.2 OAMAuthnCookie for OAM Webgates

There is one OAMAuthnCookie_<host:port>_<random number> set by each OAM 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](#).

Cookie Expiration: For OAM Webgate and OAMAuthnCookie, expiration is controlled by the tokenValidityPeriodparameter, which controls the valid token (or cookie) time.

This key is known to both the OAM 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.

21.5.2.3 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 How to Launch the Command-Line Interface in *Administering Oracle Fusion Middleware*.

 **See Also:**

- [WLST Command Reference for WebLogic Server](#)
- [Table 21-6](#)

21.5.2.4 OAMRequestContext

The OAMRequestContext is set or cleared by the 12c Resource WebGate and protected by the key known to the respective 12c WebGate and the OAM Server.

This cookie is configured to store the state about the user's original request to a protected resource while his credentials are collected and authentication performed.

- With Internet Explorer browser:
 - When RequestContextCookieExpTime is not set, OAMRequestContext is a transient cookie.
 - When RequestContextCookieExpTime is set, the OAMRequestContext cookie expires by the time set using the "Expires" directive. This requires a time sync between the client host and Web server host.
- With all other (non-IE) browsers, when RequestContextCookieExpTime is not set OAMRequestContext expires in 5 minutes by default or by the time set using the "Max-Age" directive.

 **See Also:**

RequestContextCookieExpTime in [Table 15-2](#)

21.5.2.5 DCCCtxCookie

The DCCCtxCookie comes into play only with the Detached Credential Collector (DCC). The DCCCtxCookie is used by DCC to save various context information required during authentication.

It includes information necessary to reconstruct the original request upon completion of authentication, to maintain server affinity, and to perform iterative multi-step authentication.

By default, DCCCtxCookie is set when the DCC is first redirected away to collect credentials based on the authentication scheme (when the browser is first redirected to the login form with a form-based authentication scheme).

With the DCC, once authenticated the OAM server issues a DCC master session token to the DCC in the authenticate response. DCC then sets a host- based DCC cookie using the token and:

- **If DCC cookie Presented During Authentication:** DCC decrypts the token using a DCC key, and performs partial token validation locally (integrity check, token validity period check). If it passes, DCC performs complete token validation for timeout aspects over the OAP channel against the OAM Server.

- **If no DCC Cookie:** This indicates a first time authentication which initiates credential collection, performs sanity and syntactic checks on the credential and submits to OAM Server for validation.

**See Also:**

["Configuring OAM WebGate and Authentication Policy for DCC"](#)

21.5.2.6 DCCctxCookie_COUNT

DCCctxCookie_COUNT cookie maintains the number of cookies generated after splitting the DCCctxCookie. While the count is determined by the DCCctxCookie_COUNT cookie, the configuration parameter, MaxSplittedCookieSize fixes the size of each DCCctxCookie cookie. By default, the size is set to 4 KB.

This feature enables the Webgate to handle large DCCctxCookie cookies. When the size of the DCCctxCookie exceeds 4 KB, some browsers do not support and the user authentication fails. Set appropriate value for the Webgate configuration parameter, MaxSplittedCookieSize in order to split the DCCctxCookie cookie into multiple cookies each of size 4 KB or less. The number of cookies generated after the split of DCCctxCookie cookie is tracked by DCCctxCookie_COUNT cookie.

For example, when the requested URL is too long, the size of DCCctxCookie cookie becomes 6 KB. With the default setting, the DCCctxCookie splits into two cookies as follows:

```
DCCctxCookie_HOST:PORT_1 (size: 4 KB)
```

```
DCCctxCookie_HOST:PORT_2 (size: 2 KB)
```

You may encounter a 'bad request' error while handling a long URL. To avoid this error, edit httpd.conf file and add LimitRequestFieldSize parameter. The default value for LimitRequestFieldSize is 8 KB.

21.5.3 Support for SameSite=None Attribute in OAM Cookies

OAM adds SameSite=None attribute to all the cookies set by WebGate and OAM Server.

Support for SameSite=None Attribute in OAM Cookies

OAM adds SameSite=None attribute to all the cookies set by WebGate and OAM Server.

 **Note:**

- You must also download and upgrade to the latest WebGate Patch for this feature to work. For details, see the note [Support for SameSite Attribute in Webgate \(Doc ID 2687940.1\)](https://support.oracle.com) at <https://support.oracle.com>.
- See also the note [Oracle Access Manager \(OAM\): Impact Of SameSite Attribute Semantics \(Doc ID 2634852.1\)](https://support.oracle.com) at <https://support.oracle.com>.

Optional Configurations on OAM Server

- If SSL/TLS is terminated on Load Balancer (LBR) and OAM server is not running in SSL/TLS mode, set the following system property in **setDomainEnv.sh**: -
`Doam.samesite.flag.value=None;secure`
Alternatively, you can propagate SSL/TLS context from the LBR or Web Tier to OAM Server. For details, see Doc ID 1569732.1 at <https://support.oracle.com>.
- To disable the inclusion of `SameSite=None` by OAM Server, set the following system property in **setDomainEnv.sh**: `-Doam.samesite.flag.enable=false`
- To set `SameSite=None` for non-SSL/TLS HTTP connections, set the following system property in **setDomainEnv.sh**: -
`Doam.samesite.flag.enableNoneWithoutSecure=true`

Example - To add the system properties to `setDomainEnv.sh`:

1. Stop all the Administration and Managed Servers.
2. Edit the `$OAM_DOMAIN_HOME/bin/setDomainEnv.sh`, and add the properties as shown:

```
EXTRA_JAVA_PROPERTIES="-Doam.samesite.flag.enable=false $
{EXTRA_JAVA_PROPERTIES}"
export EXTRA_JAVA_PROPERTIES
```

3. Start the Administration and Managed Servers.

Optional Configurations for WebGate

- If SSL/TLS is terminated on LBR and OAM Webgate WebServer is not running in SSL/TLS mode, set the **ProxySSLHeaderVar** in the **User Defined Parameters** configuration to ensure that WebGate treats the requests as SSL/TLS. For details, see [User-Defined WebGate Parameters](#).
- To disable inclusion of `SameSite=None` by OAM WebGate, set `SameSite=disabled` in the **User Defined Parameters** configuration on the console. This is a per-agent configuration.
- To set `SameSite=None` for non-SSL HTTP connections, set `EnableSameSiteNoneWithoutSecure=true` in the **User Defined Parameters** configuration on the console. This is a per-agent configuration.

 **Note:**

In deployments using mixed SSL/TLS and non-SSL/TLS components: For non-SSL/TLS access, OAM Server and Webgate do not set `SameSite=None` on cookies. Some browsers (for example, Google Chrome) do not allow `SameSite=None` setting on non-secure (non-SSL/TLS access) cookies, and therefore, may not set cookies if a mismatch is found.

Therefore, it is recommended that such mixed SSL/TLS and non-SSL/TLS deployments are moved to SSL/TLS Only deployments to strengthen the overall security.

21.6 Configuring Single Sign-On with Access Manager

Administrators can perform configuration of single sign-on with Access Manager, manage resource types, host identifiers, and add resources and policies.

For each task, a link to additional information is included.

1. Review all topics in this chapter to get familiar with the Access Manager SSO policy model.
2. Configure a single sign-on logout URL for each application you want to protect, using documentation for your specific application.
3. Install and register an Agent on each Web server that is hosting an application to protect using either method. See:
 - [Introduction to Agents and Registration](#)
 - [Registering and Managing OAM Agents](#)
4. Proceed to manage resource types, host identifiers, authentication schemes, and modules:
 - [Managing Authentication and Shared Policy Components](#)
5. Locate an existing Application Domain (or start a fresh one) and add resources and policies, as described in:
 - [Managing Policies to Protect Resources and Enable SSO](#)

Managing Authentication and Shared Policy Components

Administrators can configure shared policy components, manage authentication schemes, and deploy and manage plug-ins for authentication.

- [Prerequisites to Managing Authentication and Shared Policy Components](#)
- [Configuring Shared Policy Components](#)
- [Managing Resource Types](#)
- [Managing Host Identifiers](#)
- [Understanding Authentication Methods and Credential Collectors](#)
- [Managing Native Authentication Modules](#)
- [Orchestrating Multi-Step Authentication with Plug-in Based Modules](#)
- [Deploying and Managing Individual Plug-ins for Authentication](#)
- [Managing Authentication Schemes](#)
- [Extending Authentication Schemes with Advanced Rules](#)
- [Configuring Challenge Parameters for Encrypted Cookies](#)
- [Configuring Authentication POST Data Handling](#)
- [Long URL Handling During Authentication](#)
- [Using Application Initiated Authentication](#)

22.1 Prerequisites to Managing Authentication and Shared Policy Components

The Oracle Access Management Console and at least one OAM Server must be installed and running within a WebLogic Server domain, and Access Manager must be running with at least two registered Agents.

Oracle recommends that you review information in [Understanding Single Sign-On with Access Manager](#) before performing activities in this section.

22.2 Configuring Shared Policy Components

You can configure shared policy components required for use in Access Manager authentication policies that protect resources and enable single sign-on.



See Also:

[Understanding Single Sign-On with Access Manager](#)

1. Confirm that the desired resource type is defined, as described in this chapter:
 - [Managing Resource Types](#)
2. Confirm that a host identifier definition named for the agent was created during agent registration, (or create one yourself), as described in:
 - [Managing Host Identifiers](#)
3. Gain comprehension about credential collection with Access Manager:
 - [Understanding Authentication Methods and Credential Collectors](#)
4. Learn about and use the authentication plug-ins that enable multi-step authentication:
 - [Orchestrating Multi-Step Authentication with Plug-in Based Modules](#)
 - [Deploying and Managing Individual Plug-ins for Authentication](#)
5. Create and manage authentication schemes that you can add to authentication policies, as described in:
 - [Managing Authentication Schemes](#)
 - [Configuring Challenge Parameters for Encrypted Cookies](#)
6. Set up your own global password policy for either the default embedded or optional detached credential collector (unless specified, tasks apply to both ECC and DCC, with minor changes noted in the discussion):
 - [Accessing Password Policy Configuration Page](#)
 - [Managing Global Password Policy](#)
 - [Configuring Password Policy Authentication](#)
 - [DCC: Configuring OAM WebGate and Authentication Policy for DCC](#)
 - [Completing Password Policy Configuration](#)
7. Proceed to [Managing Policies to Protect Resources and Enable SSO](#) to set up authentication policies.

22.3 Managing Resource Types

Administrators can add a resource to an application domain, search a defined resource type, or create a defined resource type.

- [Resource Types and Their Use](#)

- [Resource Type Page](#)
- [Searching for a Specific Resource Type](#)
- [Creating a Custom Resource Type](#)

22.3.1 Resource Types and Their Use

When adding a resource to an Application Domain, Administrators must choose from a list of defined Resource Types.

Oracle-provided resource types include:

- HTTP
- wl_authen
- TokenServiceRP

Administrators can configure additional resource types, and define operations on both Oracle-provided and custom resource types. A particular resource can be defined to use a subset of the declared operations, or all of them (which includes any new operators defined on the resource's type subsequently). Administrators cannot remove custom resource types or operations for which resources have been created. Oracle-provided resource types and operations are marked as read-only within the policy store and cannot be removed.



Note:

Changes to the operation list of a resource type is not allowed if a resource of that type exists.

22.3.2 Resource Type Page

In the Oracle Access Management Console, resource types are organized with other Components under the Policy Configuration tab. The navigation tree shows Oracle-provided resource types: HTTP, wl_authen, and TokenServiceRP.



Note:

Pre-defined resource types cannot be deleted. Pre-defined operations are shown with a lock icon and cannot be deleted. Additional operations can be created, edited, or deleted as needed.

The HTTP resource type, shown in [Figure 22-1](#), is used for Web applications protected by Access Manager and accessed using internet protocols (HTTP or HTTPS).

Figure 22-1 Default HTTP Resource Type Definition

HTTP Resource Type Duplicate Apply

Use the following screen to create a Resource Type. Custom Resource Types will be listed along with the default Resource Types when adding resources to an Authentication or Authorization policy.

Name HTTP

Resources for representing web resources used with

Description HTTP and HTTPS protocols.

...

This is a predefined resource type which cannot be deleted, nor can any operations initially defined on it. However, additional operations can be created, edited or deleted as needed. Predefined operations are shown with a lock icon.

Operations + x

Operation
CONNECT
OPTIONS
POST
PUT
TRACE
GET
HEAD
DELETE

The `wl_authen` resource type is shown in [Figure 22-2](#). It is used for Fusion Middleware applications that use one of the following Access Manager Identity Assertion Provider configurations described in the *Securing Applications with Oracle Platform Security Services*:

- Identity Asserter
- Identity Asserter with Oracle Web Services Manager
- Authenticator function

Figure 22-2 Default Resource Type wl_authen

wl_authen Resource Type Duplicate Apply

Use the following screen to create a Resource Type. Custom Resource Types will be listed along with the default Resource Types when adding resources to an Authentication or Authorization policy.

Name wl_authen

Resources for representing WebLogic Authentication

Description schemes.

...

This is a predefined resource type which cannot be deleted, nor can any operations initially defined on it. However, additional operations can be created, edited or deleted as needed. Predefined operations are shown with a lock icon.

Operations + x

Operation
LOGIN

The `TokenServiceRP` resource type represents the Token Service Relying Party, as shown in [Figure 22-3](#). The operation for this resource type is Issue.

Figure 22-3 Default Resource Type TokenServiceRP Resource Type

TokenServiceRP Resource Type Duplicate Apply

Use the following screen to create a Resource Type. Custom Resource Types will be listed along with the default Resource Types when adding resources to an Authentication or Authorization policy.

Name TokenServiceRP

Resources for representing Token Service Relying

Description Party.

⋮

This is a predefined resource type which cannot be deleted, nor can any operations initially defined on it. However, additional operations can be created, edited or deleted as needed. Predefined operations are shown with a lock icon.

Operations + x

Operation
🔒 ISSUE

Table 22-1 describes the elements in each resource type definition.

Table 22-1 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.
Operations	Optional. Policies that govern a particular resource apply to all specified operations defined for the resource. Add (or remove) operations for this resource type as a string and the operations will be available when you define a resource of this type within an Application Domain. There is no limit to the number of operations that can be added to the resource type. <ul style="list-style-type: none"> • Get • Post • Put • Head • Issue (TokenServiceRP) • Login (wl_authen) • Delete • Trace • Options • Connect • <i>Other</i> (available with Oracle Access Manager 10 is not supported in 11g). <p>Remote Registration: During automatic policy creation, specified operations are supported. During automatic policy creation with no operations specified, then All operations defined for that type are supported.</p> <p>See Also: Resource Types and Their Use and Resources in an Application Domain.</p>

22.3.3 Searching for a Specific Resource Type

Users with valid Administrator credentials can to locate a defined resource type.



See Also:

["SSO Agent Search Page"](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Resource Types** in the Access Manager section.
3. In the **Name** field, 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**.

4. In the results table, you can:
 - **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.
 - **Reorder Columns:** Select a **View** menu item to alter the appearance of the results table.

22.3.4 Creating a Custom Resource Type

Users with valid Administrator credentials can create a defined resource type.

For instance, you can define a custom resource type that applies to as few as one or two (or more) operations. Any defined custom resource type is listed with default resource types when adding resources to an authentication or authorization policy.



See Also:

- ["Resource Types and Their Use"](#)
- ["Defining Resources in an Application Domain"](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.

2. In the Application Security console, click **Resource Types** in the Access Manager section.
3. Click **Create Resource** type.
4. In the page that appears, enter the following information:
 - **Name:** A unique name that identifies this resource type.
 - **Description:** Optional.
 - **Operations:** Click **+** in the **Operations** table, type the operation name into the field provided. Repeat as needed to define all operations for this resource type.
 - **Reconfigure Table:** Select a **View** menu item to alter the appearance of the results table.
5. Click **Apply** to submit this custom resource definition.
6. Add this resource definition to an Application Domain as described in "[Adding and Managing Policy Resource Definitions](#)".

22.4 Managing Host Identifiers

Administrators can create, modify, and remove host identifiers manually.

- [About Host Identifiers](#)
- [About Virtual Web Hosting](#)
- [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](#)

22.4.1 About Host Identifiers

Access Manager policies protect resources on computer hosts. Within Access Manager, the computer host is specified independently using a host identifier.

[Table 22-2](#) 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 22-2 Host Identifiers Examples

Sample 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.

Table 22-2 (Cont.) Host Identifiers Examples

Sample Host Identifier	Description
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.

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 Access Manager 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 Management 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, Oracle Access Management 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](#)

22.4.1.1 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 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](#)"), 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://example-wiki.uk.example.com/wikiexample
```

Here, `wikiexample` is the resource URL and `example-wiki.uk.example.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](#)"

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.

22.4.1.2 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](#)".

22.4.1.3 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 Access Manager.

You can identify Web server hosts to 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:

- example.com
- example.com:80
- www.example.com
- www.example.com:80
- 216.200.159.58
- 216.200.159.58:80

22.4.2 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.

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.

22.4.2.1 Configuring Virtual Hosting for Non-Apache Web Servers

Ensure that the Virtual Host box is checked on the OAM 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 `example2` in a URL:

```
http://example2
```

On the Web server, if one of the Web sites has a host named `example2`, the request is served by the matching virtual site.

In the Preferred Host field of the expanded OAM 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

22.4.2.2 Associating a Webgate for Apache with Virtual Hosts, Directories, or Files

Ensure that the Virtual Host box is checked on the OAM 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 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_exampleapps_ptcweb32/intermediateca.cer
    SSLCertificateFile Z:/Apps/sslcert_exampleapps_ptcweb32/
sslcert_myapps_ptcweb32.cer
    SSLCertificateKeyFile Z:/Apps/sslcert_exampleapps_ptcweb32/
sslcert_myapps_ptcweb32.key
    ErrorLog logs/proxysitel_log
    CustomLog logs/proxysitel_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

```

22.4.3 Host Identifier Page

A host identifier is automatically created when an Agent (and application) are registered using either the Oracle Access Management 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 Management 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 22-4 illustrates the Create 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 22-4 Create Host Identifier Page

Access Manager >

Create Host Identifier Host Identifier Apply

A Host Identifier must be configured for each Web site or virtual Web site that you want to protect with Access Manager. Host identifier simplifies the identification of a Web server host by defining all possible hostname variations.

* Name

Description

Host Name Variations + X

Host Name	Port
No host is added to this Host Identifier	

Table 22-3 describes the host identifier definitions.

Table 22-3 Host Identifier Definitions

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.
Host Name Variations	<ul style="list-style-type: none"> Host Name: A list of the various host names or permutations that users might use when accessing the application. See also: "Host Identifier Variations" and "Host Identifier Guidelines". Port: The Web server port used by each host or permutation

22.4.4 Creating a Host Identifier

Users with valid Administrator credentials can 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.

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"](#)
- ["About Virtual Web Hosting"](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Host Identifiers** in the Access Manager section.
3. Click **Create Host Identifier**.
4. On the Create Host Identifier page, fill in the:
 - a. **Name**
 - b. **Description**
 - c. **Host Name Variations:** Add (or remove) host name and port variations in the Operations list.

Add: Click the **Add (+)** button, then enter a new host name and port combination to identify variables that map to the Host Identifier Name.

Remove: Click a host name, then click the **Delete** button to remove it.
5. Repeat step 3c as needed to identify all variations of this host that users can access.
6. Click **Apply** to submit the new definition (or close the page without applying changes).
7. Close the Confirmation window, and confirm the new definition is listed in the results table.

22.4.5 Searching for a Host Identifier Definition

Users with valid Administrator credentials can search for a specific host identifier.

 **Note:**

During Delete, if the Host Identifier is associated with a resource, you are prompted with an alert. Without any association, the Host Identifier is deleted successfully.

**See Also:**

["SSO Agent Search Page"](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Host Identifiers** in the Access Manager section.
3. In the Search Host Identifiers page **Name** field, enter a name (or a partial name with wild card (*)), or leave the Name field blank to show all Host Identifiers. For example:
`my_h*`
4. Click the **Search** button to initiate the search and display results in a table, then:
 - **View or Edit:** Double-click the name in the Search Results table to display the configuration page, then add or edit as usual.
 - **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 Search Results table to a full page (or from the View menu, click Detach).
 - **Reorder Columns:** From the View menu, select reorder Columns and use the arrows provided to reorder the columns.

22.4.6 Viewing or Editing a Host Identifier Definition

Users with valid Administrator credentials can 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:**

["Host Identifier Page"](#)

1. Locate the desired host identifier and view it as described in ["Searching for a Host Identifier Definition"](#).
2. On the Host Identifier page, modify information as needed ([Table 22-3](#)):
 - a. **Name**

- b. Description**
- c. Host Name Variations:** In the table provided:
 - Add (+) Host Name Variations:** Click the Add (+) button, then enter a new host name and port combination to identify variables that map to the Host Identifier Name.
 - Delete (X) Host Name Variations:** Click a host name, then click the Delete button to remove it.
3. Repeat step 3c as needed to add or remove variations.
4. Click **Apply** to submit the changes (or close the page without applying changes).
5. Dismiss the Confirmation window, and close the page when you finish.

22.4.7 Deleting a Host Identifier Definition

Users with valid Administrator credentials can 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.

If the Host Identifier is associated with a resource, you are alerted. Without any association, the Host Identifier is deleted.

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](#)" if you want to remove a single host identifier from an existing definition.

1. Locate and modify related resource definitions in any application domains that uses this host identifier. See "[Searching for a Resource Definition](#)".
2. Locate the desired host identifier as described in "[Searching for a Host Identifier Definition](#)".
3. **View:** Double-click the name in the results table to display the configuration page, and confirm this can be removed.
4. **Delete:** Click the Delete button in the tool bar to remove the selected item in the results table; confirm removal in the Confirmation window.

22.5 Understanding Authentication Methods and Credential Collectors

With Access Manager, authentication involves redirecting the requester (user) to a centralized component that performs authentication (known as the Credential Collector).

This section provides the following topics:

- [Authentication Methods Supported by Access Manager](#)
- [Embedded Credential Collector Versus Detached Credential Collector](#)
- [Authentication Event Logging and Auditing](#)

22.5.1 Authentication Methods Supported by Access Manager

Authentication is the process of proving that a user is who he or she claims to be. Authenticating a user's identity with Access Manager refers to running a pre-defined set of processes to verify the digital identity of the user. Using Access Manager, 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 or plug-ins. This section describes multi-level authentication and other authentication methods supported by Access Manager.

Multi-level Authentication

Access Manager enables Administrators to assign different authentication levels to different authentication schemes, and then choose which scheme protects which application. 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.

SSO capability enables users to access more than one protected resource or application with a single sign in. A user who is authenticated to access resources at level 2, is eligible to access resources protected at levels less than or equal to 2. 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).

For more information, see "[About Multi-Level and Step-Up Authentication](#)".



See Also:

["Multi-Step Authentication"](#)

Multi-Step Authentication

Multi-step authentication requires a custom authentication module composed of two or more authentication plug-ins that transmit information to the backend authentication scheme several times during the login process. All information collected by the plug-in and saved in the context will be available to the plug-in through the authentication process. Context data can also be used to set cookies or headers in the user's login page.

See "[Simple Form Versus Multi-Factor \(Multi-Step\) Authentication](#)".

Windows Native Authentication

Integrated Windows Native Authentication is supported for Webgate protected applications. This form of authentication relies on the Kerberos authentication module. For more information, see [Configuring Access Manager for Windows Native Authentication](#) .

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 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 Access Manager, see "[Managing Authentication Schemes](#)".

22.5.2 Embedded Credential Collector Versus Detached Credential Collector

Access Manager 12c supports the embedded credential collector (ECC) by default but also enables you to configure the latest WebGate to use as a detached credential collector (DCC, also known as an Authenticating WebGate). The centralized DCC presents the login page, collects the user credentials (userID and password, for example), and sends these to the OAM Server using the back channel Oracle Access Protocol (OAP). Additional credentials can be requested using the DCC.

 **Note:**

Use of ECC is the recommended approach for credential collection. See [Understanding Credential Collection and Login](#) for more details.

When OAM Server is configured to use the DCC, the ECC and its HTTP endpoints are disabled. The only HTTP communication is to the Oracle Access Management Console hosted by the WebLogic AdminServer in the domain where the OAM Server is deployed. Connectivity to the AdminServer can be controlled at the network level, for example, to disallow administration requests from outside the internal network.

- Allowing both the ECC and DCC to co-exist enables you to use authentication schemes and policies configured for use with either the ECC or the DCC. This

enables a fallback mechanism for resources that rely on the ECC, which includes the Oracle Access Management Console.

- Disabling (turning off) the ECC entirely prohibits access to resources that rely on the ECC mechanism, including the Oracle Access Management Console.

While the embedded and detached credential collectors (ECC and DCC, respectively) are essentially the same, compare the two in [Table 22-4](#).



See Also:

"[Understanding Credential Collection and Login](#)"

Table 22-4 Comparing the DCC and ECC

Feature	DCC	ECC
Deployment	<p>The Detached Credential Collector remains a logical part of the server and acts as a front channel communication endpoint of the OAM Server. However, the DCC also:</p> <ul style="list-style-type: none"> • Stands alone (detached from the OAM Server and does not require an application server). • Supports RSA SecurID passcode verification, get next token, create new pin workflows. • Authenticates Webgate with greater flexibility for server scale-out and attack resilience as well as credential collection UI construction, flow, and lifecycle management. 	<p>The Embedded Credential Collector is deployed with, and integral to, the OAM Server and part of the protocol binding layer.</p> <p>The ECC supports RSA SecurID passcode verification, get next token, create new pin workflows.</p>
DMZ Deployment	<p>Yes.</p> <p>The main benefit of a deployment using DCC in the DMZ is the termination of the end-user network connections within the public network, and the use of Oracle Access Protocol (Oracle's proprietary application network protocol) over mutually authenticated connections reaching the OAM Server. This offers a complete isolation of the OAM Server from the establishment of any unauthenticated network connection.</p> <p>Unauthenticated users cannot send malformed requests to the OAM Server.</p>	<p>No.</p> <p>A Defense in Depth strategy, at the infrastructure, is necessary to mitigate threats and risks facing a typical Enterprise.</p>
Communication channel	<p>DCC consumes HTTP/HTTPS requests from the user, then communicates with the OAM Server across the Oracle Access Protocol (back channel), which can be SSL-enabled.</p>	<p>ECC communicates with both the user and the OAM Server across HTTP/HTTPS.</p>

Table 22-4 (Cont.) Comparing the DCC and ECC


Feature	DCC	ECC
DCC login, error, and password pages	<p>Dynamic pages general login/logout and password policy with the DCC are excluded automatically through the OHS httpd.conf/webgate.conf file--you do not need to configure a policy to exclude these. See the Webgate host in <code>\$WEBGATE_HOME/webgate/ohs/oamssso/*</code>, <code>\$WEBGATE_HOME/webgate/ohs/oamssso-bin/*pl</code>, and <code>\$WEBGATE_HOME/webgate/ohs/oamssso-bin/templates/*</code> directory:</p> <ul style="list-style-type: none"> • Login page: <code>/oamssso-bin/login.pl</code> • Logout: <code>/oamssso-bin/logout.pl</code> • RSA SecurID login pages: <code>/oamssso-bin/securid.pl</code> <p>Note: Update the Perl location in the first line of the login, logout, and securid scripts in <code>/oamssso-bin</code>.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> See Also:</p> <ul style="list-style-type: none"> • Integrating RSA SecurID Authentication with Access Manager • Developing Custom Pages in Fusion Middleware Developer's Guide for Oracle Access Management. </div>	<p>Pages where the user enters her credentials arrive out of the box on the OAM Server and require no additional settings or changes.</p> <ul style="list-style-type: none"> • Login page: <code>/pages/login.jsp</code> • Logout page: <code>/pages/logout.jsp</code> • Error page: <code>/pages/servererror.jsp</code> • Multi-step: <code>/pages/mfa_login.jsp</code>
Perl Scripts for DCC-based Login and Logout	<p>Perl Scripts for DCC-based Login and Logout</p> <p>The path name of the Perl executable must be updated in Oracle-provided Perl scripts on the Webgate host <code>\$WEBGATE_HOME/webgate/ohs/oamssso-bin/*pl</code> to be consistent with the actual location.</p> <p>Unix: The <code>which</code> command finds Perl on the OAM Server. For example:</p> <pre>which perl /usr/bin/perl</pre> <p>However, Perl scripts themselves point to:</p> <pre>/usr/local/bin/perl</pre> <p>Windows: The default Perl Interpreter specified in Oracle-provided Perl scripts will not be available. You must update the Perl Interpreter path in these scripts to actual path to Perl on your system.</p>	N/A
Password enforcement	<p>Yes.</p> <p>See Configuring OAM WebGate and Authentication Policy for DCC</p>	<p>Yes</p> <p>See: Managing Global Password Policy</p>

Table 22-4 (Cont.) Comparing the DCC and ECC

Feature	DCC	ECC
Authentication scheme collection methods	DCC supports only Form Based Authentication.	ECC supports all challenge methods. The ECC collects user credentials based on the challenge method of the Authentication Scheme and sends it back to OAM Server for validation.
Custom Authentication Plug-ins and Challenge Methods	Yes; same as ECC.	All challenge methods and multi-step authentication (Password Policy and other custom authentication plugins) are supported.
Single Step (Simple Form) Authentication	Yes; same as ECC.	Yes. Both the DCC and ECC handle this, where: <ul style="list-style-type: none"> All credentials are supplied in one simple form Upon credential validation and authentication, either success or failure status is returned This can be retried upon failure
Multi-Step Authentication	Yes. Both the DCC and ECC handle complex multi-factor (multi-step, iterative, and variable) Authentication processing. In this case: <ul style="list-style-type: none"> Not all required credentials are supplied at once Depending on the authentication status, PENDING state, expected credentials and context data are returned, expecting those credentials to be supplied in the next round Each intermediate step, submit required credentials and context data to feed authentication engine, until a success or failure status returned The Authentication plug-in can have multiple steps configured See " Understanding Multi-Level and Step-Up Authentication "	Yes. Both the DCC and ECC handle complex multi-factor (multi-step, iterative, and variable) Authentication processing.

Table 22-4 (Cont.) Comparing the DCC and ECC

Feature	DCC	ECC
Authentication Processing	<p>The DCC does not restrict authentication functionality of the OAM Server in any way as compared to the ECC.</p> <p>The DCC:</p> <ol style="list-style-type: none"> 1. Handles authentication redirects from OAM Webgates. 2. Handles Form-based authentication, which consists of a challenge to the user for their credentials (simple form or multi-factor). 3. Decrypts the authentication request message from the agent using the agent key; performs basic integrity checks; validates request time; and extracts all parameters from the request including request context. 4. Constructs the authentication response message, including request context originally retrieved, encrypts obrar using the agent key. 5. Decrypts the logout redirect request using the agent key to trigger logout processing. 	<p>During authentication:</p> <ol style="list-style-type: none"> 1. The ECC handles the request coming to the protocol binding layer (PBL), which converts it and sends it to the SSO Engine. 2. The SSO Engine checks for a valid session and, if none, transfers control to the Authentication Engine. 3. The Authentication Engine checks for resource protection and fetches the authentication scheme associated with the resource. 4. The ECC interacts with the client, accepts the data, and submits this to the PBL.

Table 22-4 (Cont.) Comparing the DCC and ECC

Feature	DCC	ECC
Overriding the ECC	<p>To deploy the DCC and override the ECC, an Administrator must perform the following tasks to specify the relevant DCC URLs and forms.</p> <ul style="list-style-type: none"> • OAM Agent registration: Allow Credential Collector Operations (enable for DCC) • Authentication Module, Step Orchestration: Error (if Failure) • Authentication Scheme: Challenge Redirect URL (DCC host and port) • Authentication Scheme: Challenge URL /oamssso-bin/login.pl (DCC login pages) • Authentication Scheme: Challenge Method • Password Policy: Password Service URL for DCC (Default: /oamssso-bin/login.pl) <p>See Configuring OAM WebGate and Authentication Policy for DCC</p>	N/A
Logout Configuration	See "Configuring Logout When Using Detached Credential Collector-Enabled WebGate"	See "Configuring Centralized Logout for OAM WebGates"
Cookie/Token	<ul style="list-style-type: none"> • DCCctxCookie • OAM WebGate: OAMAuthnCookie • OAM WebGate: OAMRequestContext <p>See: "Single Sign-On Cookies During User Login"</p>	<ul style="list-style-type: none"> • OAM Webgate: OAMAuthnCookie • OAM Webgate: OAM_REQ • OAM Webgate: OAM_ID • OAM Webgate: OAMRequestContext <p>See: "Single Sign-On Cookies During User Login"</p>

22.5.3 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:**

- [Logging Component Event Messages](#)
- [Auditing Administrative and Run-time Events.](#)
- [Monitoring Oracle Access Management Performance and Access Manager Health](#)

22.6 Managing Native Authentication Modules

In Access Manager, each authentication scheme requires an authentication module.

Native authentication modules lack the flexibility to orchestrate two or more plug-ins to meet specialized authentication needs. Therefore, native authentication modules are targeted for deprecation in future releases. Oracle strongly recommends using plug-in based authentication modules as described "[Orchestrating Multi-Step Authentication with Plug-in Based Modules](#)".

This section provides the following information:

- [Native Access Manager Authentication Modules](#)
- [Viewing or Editing Native Authentication Modules](#)
- [Deleting a Native Authentication Module](#)

22.6.1 Native Access Manager Authentication Modules

Native Access Manager Authentication Modules include LDAP, LDAPNoPasswordAuthModule, Kerberos, X509, and, Custom Authentication Modules.

"[Table 22-5](#)" lists the Native Access Manager Authentication Modules and description.

Table 22-5 Native Authentication Modules

Module Name	Description
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. See Also: " Native LDAP Authentication Modules ".
LDAPNoPasswordAuthModule	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. See Also: " Native LDAP Authentication Modules ".
Kerberos	Identifies the key tab file and krb5.configuration file names and Principal. Use this plug-in when configuring Access Manager for Windows Native Authentication, as described in Configuring Access Manager for Windows Native Authentication . See Also: " Native Kerberos Authentication Module ".

Table 22-5 (Cont.) Native Authentication Modules

Module Name	Description
X509	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. See Also: " Native X.509 Authentication Module ".
Custom Authentication Modules	This type of module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This type of module generally uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function. Depending on the success or failure action defined for each plug-in, another authentication plug-in is called. See Also: " Pre-populated Plug-ins for Configuring Access Manager with Multi-Step Authentication ", and Developing Custom Authentication Plug-ins in the <i>Developer's Guide for Oracle Access Management</i> for details about developing and deploying plug-ins, custom authentication modules, and schemes that use custom modules.

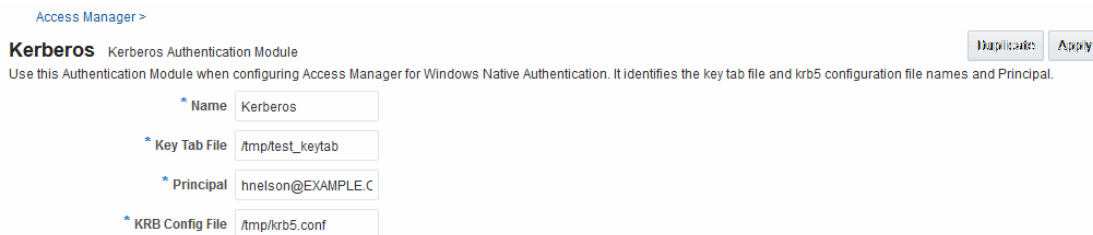
 **See Also:**

- "[Credential Challenge Methods](#)"
- To create custom authentication plug-ins, see [Developing Custom Authentication Plug-ins](#) in the *Developer's Guide for Oracle Access Management*.

22.6.1.1 Native Kerberos Authentication Module

The pre-configured Kerberos authentication module is illustrated in [Figure 22-5](#). Additional details follow the figure.

Figure 22-5 Native Kerberos Authentication Module



Access Manager >

Kerberos Kerberos Authentication Module Duplicate Apply

Use this Authentication Module when configuring Access Manager for Windows Native Authentication. It identifies the key tab file and krb5 configuration file names and Principal.

* Name

* Key Tab File

* Principal

* KRB Config File

[Table 22-6](#) describes the definition of the native Kerberos authentication module. You can use the existing, pre-configured Kerberos authentication module or create one of your own.

Table 22-6 Native 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	<p>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: <code>/etc/krb5.keytab</code>.</p> <p>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.</p> <p>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.</p>
Principal	Identifies the HTTP host for the principal in the Kerberos database, which enables generation of a keytab for a host.
KRB Config File	<p>Identifies the path to the configuration file that controls certain aspects of the Kerberos installation. A <code>krb5.conf</code> file must exist in the <code>/etc</code> directory on each UNIX node that is running Kerberos.</p> <p><code>krb5.conf</code> 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).</p>

22.6.1.2 Native LDAP Authentication Modules

Oracle provides two LDAP authentication modules: LDAP and LDAPNoPasswordAuthModule.

Both modules have the same requirements (Name and User Identity Store), as illustrated in [Figure 22-6](#). Additional details follow the figure.

Figure 22-6 Native LDAP Authentication Module

LDAP LDAP Authentication Module Duplicate Apply

Use the LDAP Authentication module for Basic and Form challenge methods. It matches the credentials (username and password) of the user who requests a resource to a user definition stored in an LDAP directory server.

* Name

* User Identity Store

[Table 22-7](#) describes the elements in an LDAP authentication module. The same elements and values are also used in LDAPNoPasswordAuthnModule.

 **Note:**

These standard LDAP Authentication Modules are targeted for deprecation. Future enhancements will not be available in standard modules. Oracle strongly recommends using plug-in based modules.

Table 22-7 Native LDAP Authentication Modules Definition

Element	Description
Name	A unique name for this module.
User Identity Store	<p>The designated LDAP user identity store must contain any user credentials required for authentication by this module. The LDAP store must be registered with Access Manager.</p> <p>See Also: "Registering and Managing User Identity Stores".</p> <p>Multiple identity store vendors are supported. Upon installation, there is only one User Identity Store, which is also the designated 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. The authentication scheme <code>OAMAdminConsoleScheme</code> relies on the LDAP module for Administrator Roles and credentials.</p> <p>See Also: "About using the System Store for User Identities" and "Administrator Lockout".</p>

22.6.1.3 Native X.509 Authentication Module

Access Manager provides a pre-configured X.509 authentication module as a default. Administrators can also create new X.509 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.

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 Access Manager, the computer hosting the Oracle Access Management 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 22-7 Native X.509 Authentication Module

X509 X509 Authentication Module Duplicate Apply

X509 Authentication Module indicate which attribute of the client's X.509 certificate should be validated against the user attribute in LDAP.

* Name

* Match LDAP Attribute

* X509 Cert Attribute

* Cert Validation Enabled

* OCSP Enabled

OCSF Server Alias

OCSF Responder URL

OCSF Responder Timeout ^ v

Table 22-8 describes the requirements of the native X.509 authentication module.



Note:

This standard Authentication Module is targeted for deprecation. Future enhancements will not be available in standard modules. Oracle strongly recommends using plug-in based modules.

Table 22-8 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 searched against given the X509 Cert Attribute value. For example, if the certificate subject EMAIL is me@example.com and it must be matched against the "mail" LDAP Attribute, an LDAP query must search LDAP against the "mail" attribute with a value "me@example.com (cn). Default: cn
X509 Cert Attribute	Defines the certificate attribute to be used to bind the public key (attributes within subject, issuer scope to be extracted from the certificate: subject.DN, issuer.DN, subject.EMAIL, for example). See Also. Match LDAP Attribute earlier in this table.

Table 22-8 (Cont.) X509 Authentication Module Definition

Element	Description
Cert Validation Enabled	Enables (or disables if not checked) X.509 Certificate validation. When enabled, the OAM Server performs the certificate validation (rather than having the WebLogic server intercept and validate the certificate before passing it to the OAM Server). Access Manager performs the entire certificate path validation.
OCSP Enabled	Enables (or disables when not checked) the Online Certificate Status Protocol. Values are either <code>true</code> or <code>false</code> . For example: OCSP Enabled: <code>true</code> 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 OCSP Responder pointing to CA certificates in .oamkeystore file--a mapping between the aliased name and the actual instance name or the IP address of the OCSP Responder instance.
OCSP Responder URL	Provides the URL of the Online Certificate Status Protocol responder. For example, OpenSSL Responder URL: <code>http://localhost:6060</code>
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.

22.6.2 Viewing or Editing Native Authentication Modules

Users with valid Administrator credentials can 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 if needed.

Note:

By default, the `LDAP` module is used in the authentication scheme that protects the Oracle Access Management Console. To ensure Administrator access, the `LDAP` module must point to the User Identity Store that is designated as the System Store. If you change the designated System Store, be sure to change the `LDAP` Module to reference the newly designated System Store.

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Authentication Modules** in the **Plug-ins** section.
3. In the Search Results list, select the desired module to open its properties page.

4. On the properties page, modify information as needed:
 - Kerberos Module: See [Table 22-6](#)
 - LDAP Module: See [Table 22-7](#)
 - X.509 Module: See [Table 22-8](#) and [Table 22-14](#)
5. Click **Apply** to submit the changes and close the Confirmation window (or close the page without applying changes).
6. Add the updated authentication module to authentication schemes (or change to another authentication module in each authentication scheme that references this module), as described in "[Managing Authentication Schemes](#)".

22.6.3 Deleting a Native 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 native module.

Prerequisites

In each authentication scheme that references the module to be deleted, specify another authentication module.

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Authentication Modules** in the **Plug-ins** section.
3. Optional: Open the module to verify this is the module to remove, then close the page.
4. In the Search Results list, click the desired module name, then click the **Delete** button.
5. Confirm removal (or dismiss the confirmation window to retain the module).

22.7 Orchestrating Multi-Step Authentication with Plug-in Based Modules

Authentication involves determining which credentials a user must supply when requesting access to a resource, gathering credentials, and returning a response that is based on the results of credential validation.

All authentication processing relies on an authentication module to define the rules governing requirements and transmission of information to the backend authentication scheme. All information collected by the plug-in and saved in the context is available to the plug-in through the authentication process.

Context data can also be used to set cookies or headers in the user's login page.

**Note:**

Oracle strongly recommends using authentication plug-ins to create custom authentication modules.

This section provides the following topics:

- [Simple Form Versus Multi-Factor \(Multi-Step\) Authentication](#)
- [Access Manager Plug-ins for Multi-Step Authentication Modules](#)
- [Pre-populated Plug-ins for Configuring Access Manager with Multi-Step Authentication](#)
- [Example: Leveraging SubjectAltName Extension Data and Integrating with Multiple OCSP Endpoints](#)
- [Creating a Custom Authentication Module using Bundled Plug-ins](#)
- [Steps and Plug-ins in Customized Step-up Authentication Module](#)
- [Configuring an HTTPToken Extractor Plug-in](#)
- [JSON Web Token Plug-in](#)

**See Also:**

To create custom authentication plug-ins, see *Developing Custom Authentication Plug-ins* in the *Oracle Fusion Middleware Developer's Guide for Oracle Access Management*.

22.7.1 Simple Form Versus Multi-Factor (Multi-Step) Authentication

Simple form-based authentication is the default and does not require additional configuration unless you want to customize forms. Authentication plug-ins provide processing that meets your specific multi-step authentication needs.

Simple form-based authentication relies on the default embedded or optional detached credential collector and Web forms that process user logins with Access Manager authentication mechanisms. With simple form-based authentication:

- All credentials are supplied in one simple form.
- Upon credential validation and authentication, either success or failure status is returned.
- Authentication can be retried upon failure.

**See Also:**

[Developing Custom Pages](#) for details about customizing login pages and forms

For dynamic, multi-step authentication, Access Manager provides a number of plug-ins with which you can design and orchestrate your own customized authentication modules. Both the

ECC and DCC handle complex multi-factor (multi-step, iterative, and variable) Authentication processing, where:

- Not all required credentials are supplied at once
- Depending on the authentication status, PENDING state, expected credentials and context data are returned, expecting those credentials to be supplied in the next round
- Each intermediate step, submit required credentials and context data for the authentication engine, until a success or failure status is returned.
- The Authentication plug-in can have multiple steps configured.



Note:

Administrators can install multiple user identity stores for Access Manager. Each identity store can rely on a different LDAP provider. Each authentication plug-in can be configured to use a different user identity store.

Table 22-9 provides more information about these two forms of authentication.

Table 22-9 Simple Form versus Multi-Step Authentication

Authentication Method	Description
Simple form-based authentication	<p>Simple form-based authentication relies on Credential Collectors (both ECC and DCC) and Web forms that process user logins using Access Manager authentication mechanisms. This is the default and does not require additional configuration unless you want to customize forms.</p> <p>See Also: Developing Custom Pages for details about customizing login pages and forms</p>

Table 22-9 (Cont.) Simple Form versus Multi-Step Authentication

Authentication Method	Description
Multi-Step Authentication	<p>Multi-step authentication requires a custom authentication module composed of two or more authentication plug-ins that transmit information to the backend authentication scheme several times during the login process. All information collected by the plug-in and saved in the context will be available to the plug-in through the authentication process. Context data can also be used to set cookies or headers in the user's login page.</p> <p>Multi-Step authentication relies on:</p> <ul style="list-style-type: none"> • Authentication Chaining: You can chain multiple authentication plug-ins in a new authentication module, and add the module to an authentication scheme. • Challenge Mechanism: Controls the way in which the required credentials are collected. Currently, this is tied to the authentication scheme. Both the ECC and DCC use the same challenge mechanisms. • Credential Collection: Either the ECC or DCC can be used for multi-step authentication. (DCC provides greater flexibility for interactions with users or programmatic entities when collecting authentication-related information that involves several methods to establish the user's identity). <p>See Also: Configuring OAM WebGate and Authentication Policy for DCC "Steps and Plug-ins in Customized Step-up Authentication Module" Developing Custom Authentication Plug-ins for details about custom authentication plug-ins</p>

22.7.2 Access Manager Plug-ins for Multi-Step Authentication Modules

Plug-ins operate with either the default embedded credential collector (ECC) or the optional detached credential collector (DCC-enabled WebGate). Each authentication plug-in provides an individual piece of functionality that you can use alone or string together into a series of steps. The lifecycle of a plug-in centers around the ability to add and use the plug-ins to build features and work flows that act as extensions to the OAM Server. Each plug-in is deployed as a JAR file and each plug-in's configuration requirements must be given in XML format. You can create custom plug-in based authentication modules using existing Access Manager. To create your own plug-ins, see *Developing Custom Authentication Plug-ins* in the *Oracle Fusion Middleware Developer's Guide for Oracle Access Management*.



Note:

Standard (native) Authentication Modules are targeted for deprecation; future enhancements will not be available in the standard modules. Oracle strongly recommends using plug-in based modules as described in "[Orchestrating Multi-Step Authentication with Plug-in Based Modules](#)".

Figure 22-8 shows plug-ins available out of the box. These plug-ins, and any that you create using the SDK and import, appear in a list when you add steps to build a custom authentication module.

Figure 22-8 Access Manager Plug-ins for Customized Authentication Modules

Row#	Plug-in Name	Description	Activation Status	Type
1	UserIdentificationPlugIn		Activated	Authentication
2	UserAuthenticationPlugIn		Activated	Authentication
3	UserPasswordPolicyPlugIn		Activated	Authentication
4	UserForgotPasswordPlugIn		Activated	Authentication
5	FedUserProvisioningPlugIn		Activated	User Provisioning
6	TAPAssertionPlugIn		Activated	Assertion
7	TAPIdentifyPlugIn		Activated	Assertion
8	KerberosTokenIdentifier		Activated	Authentication
9	KerberosTokenAuthenticator		Activated	Authentication
10	MTModelIdentifierPlugIn		Activated	Authentication
11	FedModelIdentifierPlugIn		Activated	Authentication
12	SIMFedModelIdentifierPlugIn		Activated	Authentication
13	BasicFedModelIdentifierPlugIn		Activated	Authentication
14	SIMBasicFedModelIdentifierPlugIn		Activated	Authentication
15	FedProgramaticAuthnPlugIn		Activated	Authentication
16	X509CredentialExtractor		Activated	Authentication
17	TAPRequestPlugIn		Activated	Authentication
18	TAPUserAuthenticationPlugIn		Activated	Authentication
19	TenantDisambiguationPlugIn		Activated	Authentication
20	FedAuthnRequestPlugIn		Activated	Authentication
21	FedUserAuthenticationPlugIn		Activated	Authentication
22	RSA SecurID PlugIn		Activated	Authentication
23	HTTPTokenExtractor		Activated	Authentication
24	UserAuthnLevelCheckPlugIn		Activated	Authentication
25	CredentialCollectorPlugIn		Activated	Authentication
26	IPFUserPasswordPolicyPlugIn		Activated	Authentication
27	PersistentLoginPlugIn		Activated	Assertion
28	IPFUserForgotPasswordPlugIn		Activated	Authentication
29	TOTPPlugIn		Activated	Authentication
30	GenericStatePlugIn		Activated	Authentication
31	EssoProvisioningPlugIn		Activated	Authentication
32	FedAttributeRequestPlugIn		Activated	Authentication
33	OAuthTokenResponsePlugIn		Activated	Assertion
34	SFAPPlugIn		Activated	Authentication
35	CredentialChallengePlugIn		Activated	Authentication
36	TAPResponseValidationPlugIn		Activated	Authentication

The Name generally defines the component that relies on the plug-in. The Description is optional. The Type column indicates the purpose of the plug-in. Activation Status lets you know if this is active and ready to use.

 **See Also:**

Developing Custom Authentication Plug-ins in the *Oracle Fusion Middleware Developer's Guide for Oracle Access Management* for details about building your own custom plug-ins. You can import new plug-ins, distribute, activate, deactivate, and remove custom plug-ins.

Whether you use an Oracle-provided plug-in or create one of your own, adding a plug-in when you create a custom authentication module is the same. Each custom module requires the following types of information:

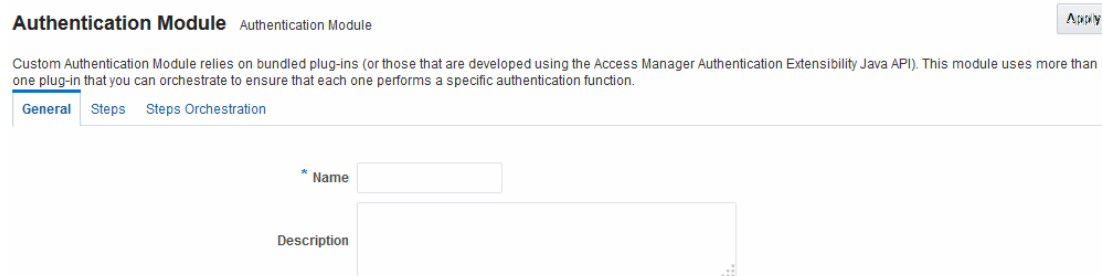
- **General** identifies the unique name and optional description for the individual plug-in.
- **Steps** identify the specific plug-ins to use, and their execution order, based on the configuration details of each plug-in (including the user identity store to use).
- **Step Orchestration** specifies the action to be taken on success or on failure or on error.

 **Note:**

When multi-factor authentication is used, the `UserIdentificationPlugin` should be invoked in the last pass during the authentication process.

Figure 22-9 shows the Custom Authentication Module within the Access Manager section of the System Configuration tree. Each module has three configuration tabs.

Figure 22-9 Creating Custom Authentication Modules: General



Authentication Module Authentication Module Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General Steps Steps Orchestration

* Name

Description

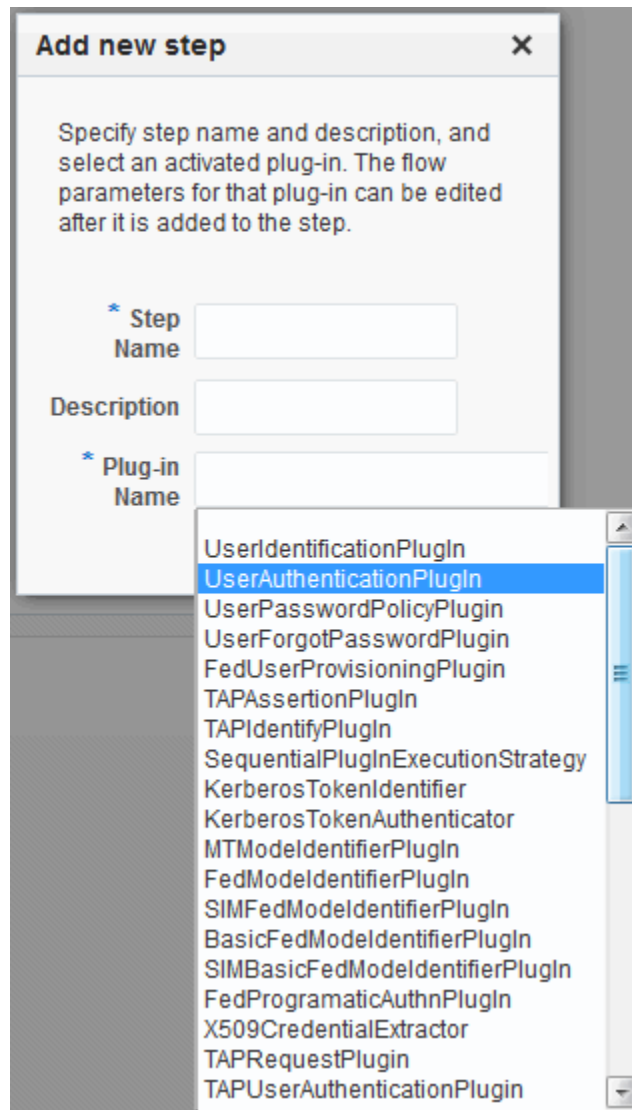
Table 22-10 describes the content of the General tab.

Table 22-10 General tab

Element	Description
Name	A unique name up to 60 characters.
Description	Optional; up to 250 characters.

Clicking the Steps tab opens a fresh page where you can add a new step. 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 22-10 Adding a Step and Associating a Plug-in



[Table 22-11](#) describes the information required when adding a new step. Each step requires a plug-in and each plug-in requires specific details for proper operation.

Table 22-11 Add New Step Entries, Steps Results Table, and Details Section

Element	Description
Step Name	The unique name you enter to identify this step, up to 60 characters.
Description	The optional description for this step, as entered when adding the step (up to 250 characters).

Table 22-11 (Cont.) Add New Step Entries, Steps Results Table, and Details Section

Element	Description
Plugin Name	The plug-in that you select for a particular step from the list of imported and activated plug-ins. See Also: To create custom plug-ins, see <i>Developing Custom Authentication Plug-ins</i> in the <i>Oracle Fusion Middleware Developer's Guide for Oracle Access Management</i> .
Step Details	Plug-in configuration details must be specified to ensure proper operation. Details might differ depending the chosen plug-in and its requirements. See Also: Table 22-12 .

[Table 22-12](#) describes the Plug-in Parameter Details required by Oracle-provided plug-ins. Absent from this table are the plug-in exceptions (those plug-ins with no initial parameters): `KerberosTokenIdentifier`, `FedAuthnRequestPlugIn`, and `FedUserAuthenticationPlugIn`.

Table 22-12 Parameter Details for Various Plug-ins

Plug-in Parameter	Display Name	Description
KEY_IDENTITY_STORE_REF	Identity Store Name	Most plug-ins require this attribute to ensure that the appropriate user identity store is called during authentication. The following plug-in uses only this property: <ul style="list-style-type: none"> • <code>TAPAssertionPlugIn</code> For additional Details required by plug-ins that employ this property, see: <ul style="list-style-type: none"> • <code>UserIdentificationPlugIn</code> • <code>UserAuthenticationPlugIn</code> • <code>UserPasswordPolicyPlugIn</code> • <code>TAPUserAuthenticationPlugIn</code> • <code>TenantDisambiguationPlugIn</code>

Table 22-12 (Cont.) Parameter Details for Various Plug-ins

Plug-in Parameter	Display Name	Description
CredentialCollectorPlugIn	CredentialCollectorPlugIn	<p>This plugin allows the administrator to configure which credentials will be collected for authentication. Credentials to be collected are configured as step parameters. The plugin validates these parameters and renders the UI to collect the credentials. After user input, the plugin parses the credential parameters and builds the user context with credential objects.</p> <p>NOTE: Plugin error responses are set to the context if the credentials are invalid and the plugin returns failure.</p> <p>The plugin supports the collection of 4 credentials as step level parameters.</p> <ol style="list-style-type: none"> 1. CRED_PARAM_1 2. CRED_PARAM_2 3. CRED_PARAM_3 4. CRED_PARAM_4 <p>The following example illustrates how to collect a username and password.</p> <pre>CRED_PARAM_1= {ID=KEY_USERNAME}, {DISPLAY_NAME=KEY_USERNAME},{TYPE=text} {ID=KEY_PASSWORD}, {DISPLAY_NAME=KEY_PASSWORD}, {TYPE=password}</pre> <p>Where ID, DISPLAY_NAME and TYPE are constants.</p>
Actiontype	Action Type	Indicates if the plugin wants to REDIRECT or FORWARD to the login page to collect credentials.
loginPageURL	Login Page URL	The URL to which the user will be forwarded or redirected for credential collection.
NO_OF_CREDENTIALS		The number of credentials provided for the plugin instance. If the number of instances is more than 4, the user must update the oam-config file to add additional CRED_PARAMS as plugin parameters.
UserIdentificationPlugIn	UserIdentificationPlugIn	This native plug-in maps the user to a specific LDAP user record.
KEY_LDAP_FILTER	LDAP Filter	The search filter required to identify the user. LDAP attributes are used when defining an LDAP search filter.
KEY_SEARCHBASE_URL	LDAP Searchbase	The search base required for the query. The node in the directory information tree (DIT) under which user data is stored; the highest possible base for all user data searches.

Table 22-12 (Cont.) Parameter Details for Various Plug-ins

Plug-in Parameter	Display Name	Description
UserAuthenticationPlugIn	UserAuthenticationPlugIn	This native plug-in authenticates the supplied username/password credentials against an LDAP directory.
KEY_PROP_AUTHN_EXCEPT ION	Propagate LDAP errors	Enables (or disables) propagation of LDAP errors. UserAuthenticationPlugIn employs this attribute.
UserAuthnLevelCheckPlugIn	UserAuthnLevelCheckPlugIn	This native plug-in shall determine if the user has been authenticated to the authentication level X - where the value of X is provided by the plugin parameter AUTHN_LEVEL_FOR_PLUGIN. For example, it checks the current Authentication Level of the user with the value specified. In addition, the plug-in specifies a list of parameters to collect depending on whether the Authentication Level check succeeded or failed.
AUTHN_LEVEL_FOR_PLUGIN	AUTHN_LEVEL_FOR_PLUGIN	Specify the authentication level as an integer. Multiple steps can use UserAuthnLevelCheckPlugIn. However, each Step must have a unique name and AUTHN_LEVEL_FOR_PLUGIN. See Also: "Steps and Plug-ins in Customized Step-up Authentication Module"
UserPasswordPolicyPlugIn	UserPasswordPolicyPlugIn	
PLUGIN_EXECUTION_MODE	Mode of Operation	The execution mode of UserPasswordPolicyPlugIn. UserPasswordPolicyPlugIn is supported only when using LDAP based authentication modules. It does not work with non LDAP authentication modules. Depending upon the configuration, can operate either alone or with other default plug-ins. Values are one of the following: <ul style="list-style-type: none"> • PSWDONLY: Default. The most preferred configuration where only the password status is determined. The ID and authentication must be performed using the UserIdentification and UserAuthentication Plugins. • AUTHWITHPSWD: Both authentication and password are performed using this plug-in. • AUTHONLY: Only the user identification and authentication is performed using this plug-in
NEW_USERPSWD_BEHAVIOR	Force Password Change on First Login	Configures retroactive behavior of the new-user password-policy. Used with UserPasswordPolicyPlugIn. Values are either: <ul style="list-style-type: none"> • FORCECHANGEPASSWORD: Forces a password change. • NOFORCEPASSWORDCHANGE: The password policy change does not affect user passwords that are already set. Default: FORCECHANGEPASSWORD

Table 22-12 (Cont.) Parameter Details for Various Plug-ins

Plug-in Parameter	Display Name	Description
DISABLED_STATUS_SUPPORT	Disabled Account Status Support	Specifies whether the disabled status is to be supported and acted upon in this password service. Valid values are either True or False. Default: TRUE
URL_ACTION	Password Management Action URL	Specifies the URL to which the user is sent for password management. The type of servlet action needed for redirecting the user to the specific password page for expiry and warning pages. Values can be either: <ul style="list-style-type: none"> REDIRECT_POST REDIRECT_GET FORWARD Default: REDIRECT_POST
FedUserProvisioningPlugin		
KEY_USER_RECORD_ATTRIBUTE_LIST	List of User Attributes	For Federation. Comma-separated list of assertion attributes required to create the user record.
KEY_PROVIDERID_ATTRIBUTE_NAME	Partner Attribute Name	For Federation. The attribute name of the LDAP user record whose value will be set to the Partner's Identity Provider ID when provisioning the user. This field is optional and if empty, the Partner's Identity Provider ID will not be set in the LDAP user record.
KEY_USERID_ATTRIBUTE_NAME	User UserID Attribute	For Federation. Name of the attribute in the assertion attributes that is used as the LDAP UserID.
TAPIdentifyPlugIn		
KEY_TAP_RETURN_ATTRIBUTE	Username Mapping Attribute	Name of the attribute used for account linking by TAPIdentifyPlugIn.
SequentialPlugInExecutionStrategy		
StrategyName	Orchestration Strategy	Name of the plugin orchestration strategy required by SequentialPlugInExecutionStrategy.
KerberosTokenAuthenticator		
KEY_KEYTAB_FILE	Location of Keytab file	Name of the file containing Kerberos principals and encrypted keys required by KerberosTokenAuthenticator
KEY_PRINCIPAL	OAM Service Principal	Your OAM Account SPN, required by KerberosTokenAuthenticator.
KEY_KRB_CONFIG_FILE	Location of Kerberos Configuration file	Location of the Kerberos configuration properties file, required by KerberosTokenAuthenticator.
KEY_DOMAIN_DNS2DN_MAP	AD Domain DNS Names to DN Mapping	Comma-separated list of Active Directory DNS Domains to DN mappings required by KerberosTokenAuthenticator.
X509CredentialExtractor		

Table 22-12 (Cont.) Parameter Details for Various Plug-ins

Plug-in Parameter	Display Name	Description
KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT	User Mapping Attribute	X509 certificate Attribute to be used for user mapping required by X509CredentialExtractor.
KEY_IS_CERT_VALIDATION_ENABLED	Certificate Validation	Enable or disable X.509 certificate validation, required by X509CredentialExtractor.
KEY_IS_ENFORCE_EKU_ENABLED	Enable Extended Key Usage (EKU) Validation	Enable or Disable validation for Extended Key Usage (EKU) extensions in the client certificate: <ul style="list-style-type: none">• YES• NO (Default)• TRUE• FALSE

**Note:**

An empty or incorrect value is handled as NO.

For details, see [X.509 Authentication Using Extended Key Usage \(EKU\)](#)

Table 22-12 (Cont.) Parameter Details for Various Plug-ins

Plug-in Parameter	Display Name	Description
KEY_ENFORCE_KEY_USAGES	Object Identifier (OID) Mapping	Specify a comma separated list of object identifiers (OID) that needs to be present in the EKU extension of the client certificate.

 **Note:**

KEY_IS_ENFORCE_EKU_ENABLED **must** be set to YES/TRUE.

By default, the OID 1.3.6.1.5.5.7.3.2 is specified. This OID is used for client authentication through TLS web client.

 **Note:**

The values must be specified as OIDs only. Any other values, in any other format is not recognized and the authentication fails.

For Smart Card Login, specify the OID 1.3.6.1.4.1.311.20.2.2

To specify multiple OIDs, use a **comma (,)** between the OIDs. For example:
1.3.6.1.5.5.7.3.2,1.3.6.1.4.1.311.20.2.2.

 **Note:**

Both the values must be present in the EKU extension of the client certificate. If either of these values is not present, the authentication fails.

See also, [Global OID Reference Database](#) for OID details.

TAPRequestPlugin

Table 22-12 (Cont.) Parameter Details for Various Plug-ins

Plug-in Parameter	Display Name	Description
TAPS2PVersion	Integration Protocol Version	Token version for Integration.
TAPPartnerId	Integration PartnerId	Integration Partner Identifier.
TAPChallengeURL	Partner Integration Endpoint URL	Remote Partner End Point URL.
TAPUserAuthenticationPlugin		
KEY_USERNAME_ATTRIBUTE	Username Mapping Attribute	Name of the attribute used for account linking required by TAPUserAuthenticationPlugin
KEY_CHECK_TOKEN_EXPIRY	Enable Token Expiration Checking	Enable or disable Integration token expiration.
TenantDisambiguationPlugin		
KEY_FEDERATED_TENANTS	FederatedTenantNames	Optional names of tenants (comma separated) for whom federated authentication is enabled. Plugin will check with Federation engine if tenant names are not mentioned.
RSA SecurID Plugin		
username	Username Parameter	Name of the username plugin parameter required by RSA SecurID Plugin.
passcode	Passcode Parameter	Name of the passcode plugin parameter required by RSA SecurID Plugin.
nexttoken	Next Token Parameter	Name of the next token plugin parameter required by RSA SecurID Plugin.
newpin	New PIN Parameter	Name of the new pin plugin parameter required by RSA SecurID Plugin.
confirmnewpin	Confirm New PIN Parameter	Name of the confirm new pin plugin parameter required by RSA SecurID Plugin.
HTTPTokenExtractor		
KEY_HEADER_PROPERTY	HTTP Header Names	Comma separated list of HTTP Headers. See Configuring an HTTPToken Extractor Plug-in .
KEY_COOKIE_PROPERTY	HTTP Cookie Names	Comma separated list of Cookies. See Configuring an HTTPToken Extractor Plug-in .

Figure 22-11 illustrates the Steps tab and Details section for a custom authentication module. When adding Steps, there is no data to display in the table. However, when you add one or more Steps to the table, the Details sections are populated.

Figure 22-11 Plug-in Based Authentication Module Steps and Details

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View + × Detach

Step Name	Description	Plug-in Name
ESSO_UI_Step		UserIdentificationPlugin
ESSO_UA_Step		UserAuthenticationPlugin
ESSO_PROV_Step		EssoProvisioningPlugin

Save Cancel

Step Details

Step Name ESSO_UI_Step

Description

Plug-in Name UserIdentificationPlugin

Plug-in Parameters

KEY_IDENTITY_STORE_REF

KEY_LDAP_FILTER

KEY_SEARCH_BASE_URL

Figure 22-12 illustrates the Steps Orchestration tab of a custom authentication module, which is populated by information for each defined step (and the action you choose for each operational condition).

Figure 22-12 Steps Orchestration for Plug-in Based Authentication Modules

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General Steps **Steps Orchestration**

You can specify the initial step

* Initial Step

Name	Description	On Success	On Failure	On Error
ESSO_UI_Step		ESSO_UA_Step	failure	failure
ESSO_UA_Step		ESSO_PROV_Step	failure	failure
ESSO_PROV_Step		success	failure	failure

Table 22-13 describes the elements on the Steps Orchestration tab. 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 22-13 Steps Orchestration Tab

Element	Description
Initial Step	Choose the starting step from those listed. The list includes only those steps defined for this module.
Name	Each step added to this module 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. A list provides actions you can choose: <ul style="list-style-type: none"> • Success • Failure • <i>StepName</i> (activates the next step)
OnFailure	The action selected for failure of this step. A list provides actions you can choose: <ul style="list-style-type: none"> • Success • Failure • <i>StepName</i> (activates the next step)
OnError	The action selected for an error when executing this step. A list provides actions you can choose: <ul style="list-style-type: none"> • Success • Failure • <i>StepName</i> (activates the next step)

22.7.3 Pre-populated Plug-ins for Configuring Access Manager with Multi-Step Authentication

The following topics describe several of the native Custom modules provided with pre-populated plug-ins. You can use these to orchestrate your own custom authentication modules:

- [KerberosPlugin](#)
- [LDAPPlugin](#)
- [X509Plugin](#)
- [Password Policy Validation Module and Plug-ins](#)



See Also:

- [Table 22-12](#)
- [Example: Leveraging SubjectAltName Extension Data and Integrating with Multiple OCSP Endpoints](#)
- ["Steps and Plug-ins in Customized Step-up Authentication Module"](#)

KerberosPlugin

Use this plug-in when configuring Access Manager for Windows Native Authentication, as described in [Configuring Access Manager for Windows Native Authentication](#).

Figure 22-13 shows the KerberosPlugin module that is bundled with 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 "kerberos ticket".

Figure 22-13 KerberosPlugin

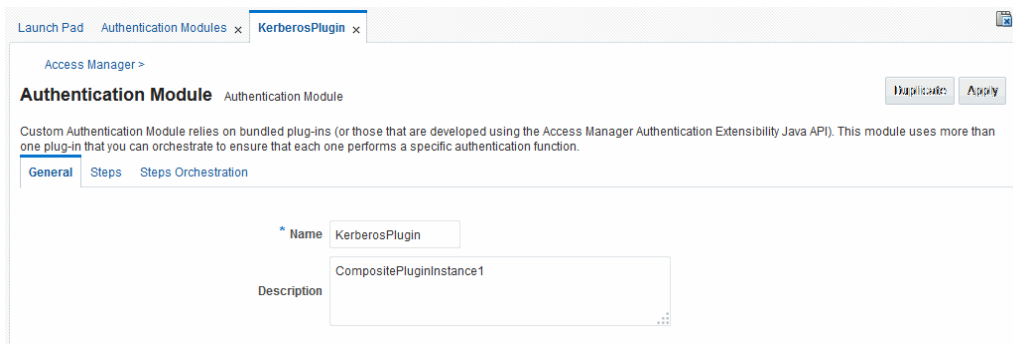


Figure 22-14 shows the default steps and details. Figure 22-15 shows the orchestration of the steps and conditions.

Figure 22-14 Default KerberosPlugin Steps and Details

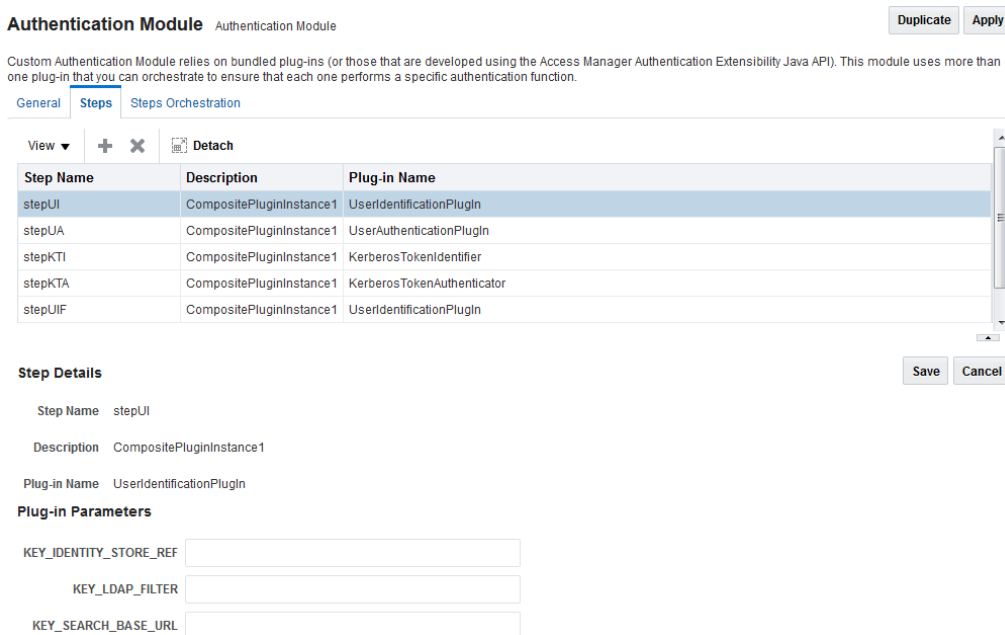


Figure 22-15 Default KerberosPlugin Steps and Orchestration

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General Steps **Steps Orchestration**

You can specify the initial step

* Initial Step

Name	Description	On Success	On Failure	On Error
stepKTI	KerberosTokenIdentifier	<input type="text" value="stepKTA"/>	<input type="text" value="stepUI"/>	<input type="text" value="failure"/>
stepKTA	KerberosTokenAuthenticator	<input type="text" value="stepUIF"/>	<input type="text" value="failure"/>	<input type="text" value="failure"/>
stepUIF	UserIdentificationPlugin	<input type="text" value="success"/>	<input type="text" value="failure"/>	<input type="text" value="failure"/>
stepUI	UserIdentificationPlugin	<input type="text" value="stepUA"/>	<input type="text" value="failure"/>	<input type="text" value="failure"/>
stepUA	UserAuthenticationPlugin	<input type="text" value="success"/>	<input type="text" value="failure"/>	<input type="text" value="failure"/>

LDAPPlugin

Figure 22-16 shows the LDAPPlugin module that is bundled with Access Manager. By default, LDAPPlugin has 2 steps, shown in Figure 22-17. Figure 22-18 shows the default orchestration of steps for LDAPplugin.

Figure 22-16 LDAPPlugin

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

* Name

Description

Figure 22-17 Default LDAPPlugin Steps and Details

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View

Step Name	Description	Plug-in Name
stepUI	CompositePluginInstance1	UserIdentificationPlugin
stepUA	CompositePluginInstance1	UserAuthenticationPlugin

Step Details

Step Name stepUI

Description CompositePluginInstance1

Plug-in Name UserIdentificationPlugin

Plug-in Parameters

KEY_IDENTITY_STORE_REF

KEY_LDAP_FILTER

KEY_SEARCH_BASE_URL

Figure 22-18 Default Orchestration of Steps for LDAPplugin

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** **Steps Orchestration**

You can specify the initial step

* Initial Step

Name	Description	On Success	On Failure	On Error
stepUI	UserIdentificationPlugin	stepUA <input type="button" value=""/>	failure <input type="button" value=""/>	failure <input type="button" value=""/>
stepUA	UserAuthenticationPlugin	success <input type="button" value=""/>	failure <input type="button" value=""/>	failure <input type="button" value=""/>

X509Plugin

Figure 22-19 shows the X509Plugin module that is bundled with 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 22-20 shows default steps and details for this plug-in. Figure 22-21 shows the default orchestration of steps for the X509Plugin.

Figure 22-19 X509Plugin

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

* Name

Description

Figure 22-20 X509Plugin Default Steps and Details

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View

Step Name	Description	Plug-in Name
stepX509	CompositePluginInstance1	X509CredentialExtractor
stepUI	CompositePluginInstance1	UserIdentificationPlugin

Step Details Save Cancel

Step Name

Description

Plug-in Name

Plug-in Parameters

KEY_IS_CERT_VALIDATION_ENABLED

KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT

With this plug-in, the root and sub CA certificates must be added to the \$DOMAIN_HOME/config/fmwconfig/amtruststore because the X509CredentialExtractor plug-in loads certificates from this location.

Table 22-14 lists the stepX509 values for Subject and Subject Alternative Names. Such processing is only supported when the X509Plugin is used.



See Also:

- [Table 22-12](#)
- [Example: Leveraging SubjectAltName Extension Data and Integrating with Multiple OCSP Endpoints](#)

Table 22-14 X509 Step Details (KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT)

issuer.D	Subject
subject.	EDIPI Note: EDIPI refers to the Electronic Data Interchange Personal Identifier.
subjectAltName.	OTHER_NAME (FASC-N) Note: FASC-N refers to the Federal Agency Smart Credential Number.
subjectAltName.	RFC822_NAME
subjectAltName.	UNIFORM_RESOURCE_IDENTIFIER

Figure 22-21 Default Orchestration for X509Plugin Steps

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can [orchestrate](#) to ensure that each one performs a specific authentication function.

General Steps **Steps Orchestration**

You can specify the initial step

* Initial Step

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



See Also:

["Example: Leveraging SubjectAltName Extension Data and Integrating with Multiple OCSP Endpoints"](#)

Password Policy Validation Module and Plug-ins

Oracle provides a Password Policy Validation Module that employs the following plug-ins as individual steps in the authentication process:

- User Identification Step
- User Authentication Step
- User Password Status Step

Figure 22-22 shows the Steps tab. Additional details follow the figure.

Figure 22-22 Password Policy Validation Module Plug-ins

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View + Detach

Step Name	Description	Plug-in Name
User Identification Step		UserIdentificationPlugIn
User Authentication Step		UserAuthenticationPlugIn
User Password Status St...		UserPasswordPolicyPlugIn

-

Step Details Save Cancel

Step Name User Identification Step

Description

Plug-in Name UserIdentificationPlugIn


Plug-in Parameters

KEY_LDAP_FILTER

KEY_IDENTITY_STORE_REF

KEY_SEARCH_BASE_URL

Figure 22-23 shows the Steps Orchestration page for the Password Policy Validation Module plug-ins, which is self explanatory.

 **See Also:**

- [Table 22-12](#)
- [Accessing Password Policy Configuration Page](#)

Figure 22-23 Steps Orchestration: Password Policy Validation Plug-ins

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

You can specify the initial step

* Initial Step

Name	Description	On Success	On Failure	On Error
User Identification Step		User Authentication Step	failure	failure
User Authentication Step		User Password Status Step	User Password Status Step	failure
User Password Status Step		success	failure	failure

22.7.4 Example: Leveraging SubjectAltName Extension Data and Integrating with Multiple OCSP Endpoints

Access Manager 11g support for personal identity verification (PIV) cards (a United States Federal smart card), is to use FASC-N and EDIPI attributes from the SubjectAltName extension to map the user during X.509 authentication. While multiple OCSP providers are not supported, you can use an OCSP Gateway or write a custom authentication plug-in that uses the OSDT OCSP APIs to validate against multiple OCSP providers.

The following functionality is available only with the X.509 Plug-in (not the X.509 Authentication module). The Plug-in configuration specifies the LDAP attribute to which the extracted attribute from the X.509 client certificate will be mapped.

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the **Plug-ins** section, click **Authentication Modules**.
3. In the list of modules, select the **X509Plugin** module.
4. In the page that opens, click **Duplicate** and fill out the fields as follows:



See Also:

["Creating a Custom Authentication Module using Bundled Plug-ins"](#)

General Tab:

- a. Name: *CustomX509Plugin*.
- b. Description: *Custom Plug-in for X509*.

Steps Tab:

- a. Click + to add a step to the plug-in.
- b. Enter a Name and Description, then select the *X509CredentialExtractor* plug-in.

Step Details:

- a. KEY_IS_CERT_VALIDATION_ENABLED `true`.
- b. KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT (Table 22-14):
`subject.EDIPI, subjectAltName.OTHER_NAME (FASC-N),
subjectAltName.RFC822_NAME,
subjectAltName.UNIFORM_RESOURCE_IDENTIFIER`
- c. Click the **Save** button.

Add Another Plug-in:

- a. Click + to add a different plug-in.
- b. Enter the Name, Description, and select *UserIdentificationPlugin*

Step Details for Second Plug-in:

- a. Set `KEY_IDENTITY_STORE_REF` to the required identity store.
 - b. Add the LDAP filter to the `KEY_LDAP_FILTER` attribute. For example:


```
(&(uid=
Unknown macro: {subject.CN}
)(mail=
Unknown macro: {subject.E}
))
```
 - c. Add the user search base, if required, to the `KEY_SEARCH_BASE_URL` attribute.
 - d. Click the **Save** button.
 - e. Proceed to Step Orchestration tab (Step2).
5. **Orchestrate Steps:**
 - a. **Initial Step:** Select the `X509CredentialPlugin` Step from the drop down.
 - b. **On Success:** `X509CredentialPlugin` step, select the `UserIdentificationPlugin` Step from the drop down list.
 - c. **On Success:** `UserIdentificationPlugin` step, select `Success` from the drop down list.
 - d. **On Failure:** Select `Failure` for both `X509CredentialPlugin` and `UserIdentificationPlugin` steps.
 - e. **On Error:** Select `Failure` for both `X509CredentialPlugin` and `UserIdentificationPlugin` steps.
 - f. Click the **Apply** button and review the confirmation window stating that the plug-in has been created successfully.
 6. Set up the Certificate Validation Module for Certificate Validation and Revocation using OCSP.

 **See Also:**

["Certificate Validation and Revocation"](#)

- a. In the Oracle Access Management Console, click **Configuration** at the top of the window.
- b. In the Configuration console, click **Certificate Validation**.
- c. In the Certificate Revocation list, confirm that **Enabled** is checked, then click **Save**.
- d. In the OCSP/CDP section, enable OCSP, enter the OCSP URL and the Subject of the OCSP Server's certificate, then click **Save**.
- e. On the command line, use the Java keytool application to import the trusted certificates into the `$DOMAIN_HOME/config/fmwconfig/amtruststore` keystore, as trusted certificate entries.

 **Note:**

Initially the keystore is empty; its password is set the first time the Java keytool application is used.

22.7.5 Creating a Custom Authentication Module using Bundled Plug-ins

Users with valid Administrator credentials can create custom authentication module that uses one or more authentication plug-ins.

This procedure outlines general steps for any authentication module (with sample information to configure an authentication X509 module for use with the Online Certificate Status Protocol (OCSP) to maintain the security of a server and other network resources).

See Also:

- ["Example: Leveraging SubjectAltName Extension Data and Integrating with Multiple OCSP Endpoints"](#)
- ["Steps and Plug-ins in Customized Step-up Authentication Module"](#)

Prerequisite

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

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, select **Create Custom Authentication Module** from the **Create (+)** drop-down list in the **Plug-ins** section.
3. In the page that appears, enter the Name and optional Description. For example: *CustomX509Plugin* and *Plugin for X509*, respectively.
Click **Apply** to save general information.
4. **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 (*X509CredentialExtractor*, for instance) 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.
5. **Define Step Details:** Use appropriate values for requested parameters (Table 22-11, Table 22-12, Table 22-19 and ["Example: Leveraging SubjectAltName Extension Data and Integrating with Multiple OCSP Endpoints"](#)):
 - a. Click a *StepName* in the table to reveal required details, enter appropriate values for the requested details.

b. Validate User Cert using OCSP:

Confirm that `KEY_IS_CERT_VALIDATION_ENABLED` is set to `true`.

Add the certificate attributes to be extracted with `KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT` (Table 22-14):

```
subject.EDIPI
subjectAltName.OTHER_NAME (FASC-N)
subjectAltName.RFC822_NAME
subjectAltName.UNIFORM_RESOURCE_IDENTIFIER
```

c. Click the Save button.**d. Repeat to configure each step appropriately.****e. Ensure that users are provisioned in any user identity stores assigned in the steps.****6. Orchestrate Steps:** See Table 22-13 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.****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 Steps c through f to orchestrate operations for each plug-in this module.****h. Review your orchestration.****7. 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.

8. In the navigation tree, confirm the new Custom Authentication Module is listed, and then close the page when you finish.**9. Use your custom module in an authentication scheme, as described in "Managing Authentication Schemes".**

22.7.6 Steps and Plug-ins in Customized Step-up Authentication Module

The processing that occurs with a customized step-up authentication module is driven by the steps and plug-ins described in Table 22-15. For more information, see Table 22-12.

Table 22-15 Steps and Plug-ins in a Customized Step-up Authentication Module

Step #	Step Name	Plug-in Name	Description
1	StandardLevelCheck-2	UserAuthnLevelCheckPlugin	<p>Configurable with the LevelCheck Rule and credentials parameters associated with the SUCCESS or FAILURE outcome resulting from the check.</p> <p>This plugin communicates with the authentication engine to determine the current authentication level of the user and compares it with the plugin level parameter AUTHN_LEVEL_FOR_PLUGIN. It interacts with a custom credential collector and checks the current Authentication Level of the user against the value specified. For example, if 2 is specified for X:</p> <ul style="list-style-type: none"> • Authentication Level \geq X returns ExecutionStatus.SUCCESS and proceeds to the next step; for example it will check for higher level authentication. • Authentication Level $<$ X returns ExecutionStatus.FAILURE and proceeds to the next step in the plugin; for example it will collect the standard credentials for level 2 (username and password). <p>Specifies parameters to collect depending on whether the Authentication Level check succeeded or failed:</p> <ul style="list-style-type: none"> • ON SUCCESS, go to SensitiveLevelCheck-6 • ON FAILURE, go to CollectUserNamePassword • ON ERROR, Failure

Table 22-15 (Cont.) Steps and Plug-ins in a Customized Step-up Authentication Module

Step #	Step Name	Plug-in Name	Description
2	CollectUserNamePassword	CredentialCollectorPlugIn	<p>This plugin interacts with the credential collector (CustomReadServlet) to allow the administrator to configure the credentials collected for authentication. Credentials to be collected are configured as step parameters. The plugin validates these parameters and renders the UI to collect them.</p> <p>The user provides the credentials that need to be collected in the step parameter. In this example, since in previous step user was not authenticated to level 2, he will be prompted to enter a user name and password.</p> <ul style="list-style-type: none"> loginPageURL: /CustomRead/Servlet (generic credential collector for UserAuthnLevelCheckPlugIn to render the interface to acquire plug-in specified credentials. No_OF_CREDENTIALS: 4 CRED_PARAM_4 CRED_PARAM_3 CRED_PARAM_2: {ID=KEY_PASSWORD}, {DISPLAY_NAME=KEY_PASSWORD}, {TYPE=password} CRED_PARAM_1: {ID=KEY_USERNAME}, {DISPLAY_NAME=KEY_USERNAME }, {TYPE=text} actiontype: FORWARD <p>Credentials to be collected should be specified in this format only for the credential collector to render the UI interface.</p> <p>Also specifies action on:</p> <ul style="list-style-type: none"> ON SUCCESS, go to UserIdentificationProcess ON FAILURE, Failure ON ERROR, Failure
3	UserIdentificationProcesses	UserIdentificationPlugIn	<p>Out of the box plug-in that maps the user to a specific LDAP user record:</p> <ul style="list-style-type: none"> ON SUCCESS, go to UserAuthenticationStep ON FAILURE, Failure ON ERROR, Failure
4	UserAuthenticationSteps	UserAuthenticationPlugIn	<p>Out of the box plug-in that authenticates the supplied username and password credentials against an LDAP directory.</p> <ul style="list-style-type: none"> ON SUCCESS, go to SensitiveLevelCheck-6 ON FAILURE go to CollectSensitiveLevelCreds ON ERROR, Failure

Table 22-15 (Cont.) Steps and Plug-ins in a Customized Step-up Authentication Module

Step #	Step Name	Plug-in Name	Description
5	SensitiveLevelCheck-6	UserAuthnLevelCheckPlugIn	This plugin communicates with the authentication engine to determine the current authentication level of the user and compares it with the plugin level parameter AUTHN_LEVEL_FOR_PLUGIN. It interacts with a custom credential collector and checks the current Authentication Level of the user against the value specified. Specifies parameters to collect depending on whether the check succeeded or failed: <ul style="list-style-type: none"> ON SUCCESS, Success ON FAILURE, go to CollectSensitiveLevelCreds ON ERROR, Failure
6	CollectSensitiveLevelCreds	CredentialCollectorPlugIn	This plugin renders the UI for collecting credentials for level 6 authentication. This is similar to CollectUserNamePwd. <ul style="list-style-type: none"> ON SUCCESS, go to ValidateSensitiveLevelCreds ON FAILURE, Failure ON ERROR, Failure CRED_PARAM_1: {ID=securitycode}, {DISPLAY_NAME=form_securecode},{TYPE=text}
7	ValidateSensitiveLevelCreds	SubjectSetPlugIn	This custom developed plug-in validates the security code against the server. <ul style="list-style-type: none"> ON SUCCESS, Success ON FAILURE, Failure ON ERROR, Failure

After defining and orchestrating plug-ins in an authentication module, you can use the module in an authentication scheme and use the scheme in a policy.



See Also:

"Creating a Custom Authentication Module using Bundled Plug-ins"

22.7.7 Configuring Step-up Authentication

You can define step-up authentication using plug-ins within a customized module.

In this example, there are users who need standard level access to pages on the corporate portal and those who need access to sensitive information. For standard applications, authentication credentials include username and password. For sensitive applications, credentials include username, password, and a security code (the later obtained with a custom plugin that validates the code).

1. Create or edit a custom authentication module for step up authentication:

Authentication Module Authentication Module Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General Steps Steps Orchestration

* Name

Description

2. Define your custom authentication module based on the Steps shown here.

Authentication Module Authentication Module Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General Steps Steps Orchestration

View

Step Name	Description	Plug-in Name
StandardLevelCheck-2		UserAuthnLevelCheckPlugin
CollectUserNamePassword		CredentialCollectorPlugin
UserIdentificationProcess		UserIdentificationPlugin
UserAuthenticationStep		UserAuthenticationPlugin
SensitiveLevelCheck-6		UserAuthnLevelCheckPlugin
CollectSensitiveLevelCreds		CredentialCollectorPlugin
ValidateSensitiveCreds		UserAuthenticationPlugin

Save Cancel

Step Details

Step Name

Description

Plug-in Name

Plug-in Parameters

loginPageURL

NO_OF_CREDENTIALS

CRED_PARAM_4

CRED_PARAM_3

CRED_PARAM_1

actiontype

CRED_PARAM_2

3. Orchestrate your Steps and Plug-ins as shown here and described in [Table 22-15](#).

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

✔ **Confirmation** ✕

Authentication Module created successfully.

General Steps **Steps Orchestration**

You can specify the initial step

* Initial Step: StandardLevelCheck-2

Name	Description	On Success	On Failure	On Error
StandardLevelCheck-2		SensitiveLevelCheck-6	CollectUserNamePassword	failure
CollectUserNamePassword		UserIdentificationProcess	failure	failure
UserIdentificationProcess		UserAuthenticationStep	failure	failure
UserAuthenticationStep		SensitiveLevelCheck-6	failure	failure
SensitiveLevelCheck-6		success	CollectSensitiveLevelCreds	failure
CollectSensitiveLevelCreds		ValidateSensitiveCreds	failure	failure
ValidateSensitiveCreds		success	failure	failure

- Sensitive Scheme:** Create or edit an Authentication Scheme for sensitive applications that uses your customized step-up authentication module. For example:

See Also:

["Managing Authentication Schemes"](#)

Create Authentication Scheme Authentication Scheme Set As Default Apply

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme must also include a defined Authentication Module.

* Name: Sensitive-Auth-Scheme-1

Description:

* Authentication Level: 6

Default:

* Challenge Method: FORM

Challenge Redirect URL: /oam/server

* Authentication Module: Step-Up-Authn-Module

* Challenge URL: /CustomReadServlet

* Context Type: customHtml

* Context Value: /sensapp/custom.html

Challenge Parameters: initial_command=NONE

- Lower-Level Scheme:** Create or edit an Authentication Scheme for the lowest level applications using your customized step-up authentication module F. or example:

Create Authentication Scheme Authentication Scheme Set As Default Apply

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme must also include a defined Authentication Module.

1 * Name: Lower-Auth-Scheme-1

2 Description: [Empty]

3 * Authentication Level: 2

4 Default:

5 * Challenge Method: FORM

6 Challenge Redirect URL: /oam/server

7 * Authentication Module: Step-Up-Authn-Module


8 * Challenge URL: /CustomReadServlet

9 * Context Type: customHtml

10 * Context Value: /sensapp/custom.html

11 Challenge Parameters: initial_command=NONE

- 6. **Sensitive Policy:** Create or edit an Authentication Policy for sensitive-level resources using your customized step-up Authentication Scheme. For example:

 **See Also:**

[Managing Policies to Protect Resources and Enable SSO](#)

Create Authentication Policy Authentication Policy Apply

Authentication Policy defines the type of verification that must be performed to provide a sufficient level of trust for Access Manager to grant access to the user making the request. A single policy can be defined to protect one or more resources in the Application Domain.

* Name: Level-6-Sensitive

Description: [Empty]

* Authentication Scheme: Sensitive-Auth-Scheme-1

Success URL: [Empty]

Failure URL: [Empty]

Resources Responses Advanced Rules

Resource Type	Host Identifier	Resource URL	Query String
This Policy does not protect any Resources			

- 7. **Lower-Level Policy:** Create or edit an Authentication Policy for the lowest level resources using your customized step-up Authentication Scheme. For example:

Create Authentication Policy Authentication Policy Apply

Authentication Policy defines the type of verification that must be performed to provide a sufficient level of trust for Access Manager to grant access to the user making the request. A single policy can be defined to protect one or more resources in the Application Domain.

* Name: Success URL:

Description:

* Authentication Scheme: Failure URL:

Resources Responses Advanced Rules

Resource Type	Host Identifier	Resource URL	Query String
This Policy does not protect any Resources			

- 8. Verify:** Verify your resources and the policies that protect them.

22.7.8 Configuring an HTTPToken Extractor Plug-in

You can configure an HTTPToken Extractor plug-in.

To configure:

1. Create a sample plug-in that will re-direct the user to the authenticating application.
The authenticating application will authenticate the user and set the user name in the HTTP header or cookie.
2. Create a custom authentication module that will access any applicable plug-ins.
For example, if you add the plug-in created in the previous step and the HTTPToken Extractor and User identification plug-ins, successful authentication occurs when the process for all three plug-ins has been successfully completed.
3. Add values for the header name and the user search filter properties.
The KEY_HEADER_PROPERTY is set in the HTTPToken Extractor plug-in while KEY_LDAP_FILTER is set in the UI plug-in. For example:
 - KEY_HEADER_PROPERTY = *cookieorheadername*
 - KEY_LDAP_FILTER = (uid={*cookieorheadername*})

Note:

The user should be present in the data store which is being used.

22.7.9 JSON Web Token Plug-in

Oracle Access Manager (OAM) provides complete but different access management solutions for both users and applications. After OAM authentication, SSO tokens are issued which can be used with WebGates or products like Oracle API Gateway. These tokens are specific to OAM though and there are often business requirements where Web services or REST services need to be protected. While OAM tokens can be used to protect web services, they are usually protected by standard tokens. A JSON Web Token (JWT) is one of the standard tokens that is widely used.

In the RSPS2 release, OAM introduced complete support for an OAuth authorization service provided by the Oracle Access Manager Mobile and Social (OAMMS) service. The OAuth Service issues a JSON Web Token (JWT) for accessing Web services subsequent to the user's authentication and/or authorization. Thus, a user can be authenticated with an OAM authentication mechanism and subsequently have both an OAM and a JWT for access to different resources. A typical scenario in which this can be used is when a WebGate protected application is accessed. The user is authenticated by an OAM authentication module and both an OAM token and a JWT are provided. The OAM token is used for access through the WebGate and the JWT can be used to access a Web service or a REST service when needed. (The Web service or REST service is protected by a product like Oracle API Gateway or Oracle Web Services Manager.)

A JSON Web Token Plug-in is now available in OAM. Use this JSON Web Token Plug-in when you need to protect REST or Web services with standard tokens. The JSON Web Token Plug-in issues both an OAM token and a Mobile and Social JWT that can be used for Web services access. Oracle API Gateway and Oracle Web Services Manager can use this JWT for Web services protection as well. See the following sections for additional details.

- [Understanding the JSON Web Token Plug-in](#)
- [Configuring the JSON Web Token Plug-In](#)

22.7.9.1 Understanding the JSON Web Token Plug-in

You can use JSON Web Token Plug-in in deployments.

The following flow describes how to use JSON Web Token Plug-in:

- Configure the Oracle Access Management WebGate to use both OAM authentication and the JSON Web Token Plug-in.
- When a user accesses a resource protected by the WebGate, the WebGate redirects the user to authenticate with Access Manager.
- Upon authentication, the plug-in identifies which OAuth service end point should generate the JWT. (OAuth service end points are unique and can be configured to point to a specific OAuth service profile within a specific Identity Domain.) Oracle Access Manager Mobile and Social creates the JWT and the plug-in returns it as a cookie. (The cookie name can be configured in the plug-in configuration.)
- The Web application intercepts the response and accesses the cookie so that it can be used later for Web service access. Depending on how the web application is deployed, there may be other options to retrieve the JWT. The user can now access the Web resource.
- When the Web resource needs to access a Web service, it extracts the OAM Mobile and Social JWT and sends it to the Oracle API Gateway.
- The Oracle API Gateway uses the Oracle OAuth Service REST API to validate the token. It then grants access to the Web service. The Oracle API Gateway can also validate the JWT locally without making a remote call to the OAuth service.

 **Note:**

Currently there is not a mechanism to pass scope to the OAuth service while issuing a JWT with OAM authentication. Consequently, the token should be considered to have global scope.

Both the OAM token timeout and the JWT timeout can be set to the same value to have the same validity. The OAM tokens and JWT are not linked, so they cannot be terminated using single logout.

22.7.9.2 Configuring the JSON Web Token Plug-In

You can configure the JSON Web Token Plug-in.

You will be creating a custom authentication module.

1. In the Oracle Access Management Console, click Application Security at the top of the window.

The Search Authentication Modules screen is displayed.

2. Select **Create Custom Authentication Module** from the **Create (+)** drop-down menu in the Plug-ins section.

The General tab is displayed.

3. Enter a name (and optional description) for the custom authentication module.

For this example, we name the module JWToken AuthnModule.

4. Click the **Steps** tab and the **+** (plus sign) to add a new step.

The Add New Step dialog is displayed. Three new steps will be added.

5. Specify a step name (and optional description), select an activated plug-in from the Plug-in name drop down list and click OK.

For this example, the values are StepUI and UserIdentificationPlugin. The flow parameters for that plug-in can be edited after it is added to the step

6. Enter values for the UserIdentificationPlugin parameters and click Save.

7. Click the **+** (plus sign) to add a second step, enter the name StepUA, select **UserAuthenticationPlugin** from the drop down list and click OK.

8. Enter values for the UserAuthenticationPlugin parameters and click Save.

9. Click the **+** (plus sign) to add a third step, enter the name StepOAuth, select **OAuthTokenResponsePlugin** from the drop down list and click OK.

10. Enter values for the OAuthTokenResponsePlugin parameters and click Save.

11. Click the **Steps Orchestration** tab to configure the orchestration of the steps in the following order.

- a. StepUI
- b. StepUA
- c. StepOAuth

12. Click **Apply** and close the Custom Authentication Module tab.

13. Click **Authentication Schemes** from the Launch Pad.
14. Select **LDAPScheme** and click **Duplicate**.
A Copy of LDAPScheme screen displays.
15. Change the value of Name to JWTToken AuthnScheme and the value of Authentication Module to JWTToken AuthnModule.
16. Click **Save**.
17. Configure an Authentication policy with the newly defined JWTToken AuthnScheme Authentication Scheme.

22.7.10 X.509 Authentication Using Extended Key Usage (EKU)

OAM supports validation of Extended Key Usage (EKU) extensions in a client certificate.

OAM validates the X.509 client certificate acquired through two-way TLS using path validation. In addition to this, it also supports the validation of EKU extensions in a client certificate.

By default, the validation of EKU extensions in a client certificate is disabled. You can enable this feature by setting the value of `KEY_IS_ENFORCE_EKU_ENABLED` to `TRUE/YES` in the `X509CredentialExtractor` plugin.

The EKU extensions in a client certificate are validated against the corresponding object identifiers (OID) that you specify under `KEY_ENFORCE_KEY_USAGES` in the `X509CredentialExtractor` plugin. For more information, see [Table 22-12](#)

The X.509 authentication succeeds only if the EKU extensions in a client certificate includes all the OIDs specified in the `KEY_ENFORCE_KEY_USAGES` parameter of the `X509CredentialExtractor` plugin.



Note:

`KEY_IS_ENFORCE_EKU_ENABLED` must be set to `TRUE/YES`.

The following sections provide details about the X.509 Authentication Module, Scheme and configurations for EKU Validation:

- [X.509 Authentication Module for EKU Validation](#)
- [X509 Steps Orchestration for EKU](#)
- [X509 Scheme for EKU](#)
- [Protecting Resources with X509 for EKU Scheme](#)

22.7.10.1 X.509 Authentication Module for EKU Validation

Create a new Authentication module and configure the following steps: `UserIdentificationPlugin` and `X509CredentialExtractor`.

To configure the X.509 authentication module for EKU validation in the Oracle Access Management Console:

1. Log into the Oracle Access Management Console as System Administrator.

2. From the Application Security Launch Pad, click **Authentication Modules** under **Plug-ins**.
3. From the Authentication Modules tab, click **Create Authentication Module** and then **Custom Authentication Module**.
4. Add a name and description for the authentication module under the General tab. For example, **X509**.
5. Add the X509 authentication module steps as follows:
 - a. Add the UserIdentificationPlugin step and set following parameters:

Table 22-16 UserIdentification Step

Step Details	Description
KEY_IDENTITY_STORE_REF	The name of the registered Identity Store containing the module users. Default: The registered Default Store.
KEY_LDAP_FILTER	Add the LDAP filter to the KEY_LDAP_FILTER attribute. Only standard LDAP attributes can be used when defining an LDAP search filter. For example: (uid={KEY_USERNAME})
KEY_SEARCH_BASE_URL	Base URL for user searches. For example: dc=us,dc=example,dc=com


- b. Add the X509CredentialExtractor step and set the following parameters:

Table 22-17 X509CredentialExtractor Step

Parameters	Display Name	Description
KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT	User Mapping Attribute	X509 certificate Attribute to be used for user mapping required by X509CredentialExtractor.
KEY_IS_CERT_VALIDATION_ENABLED	Certificate Validation	Enable or disable X.509 certificate validation, required by X509CredentialExtractor.

Table 22-17 (Cont.) X509CredentialExtractor Step

Parameters	Display Name	Description
KEY_IS_ENFORCE_EKU_ENABLED	Enable Extended Key Usage (EKU) Validation	<p>Enable or Disable validation for Extended Key Usage (EKU) extensions in the client certificate:</p> <ul style="list-style-type: none"> • YES • NO (Default) • TRUE • FALSE

 **Note:**

An empty or incorrect value is handled as NO.

For details, see [X.509 Authentication Using Extended Key Usage \(EKU\)](#)

Table 22-17 (Cont.) X509CredentialExtractor Step

Parameters	Display Name	Description
KEY_ENFORCE_KEY_USAGES	Object Identifier (OID) Mapping	Specify a comma separated list of object identifiers (OID) that needs to be present in the EKU extension of the client certificate.

 **Note:**

KEY_IS_ENFORCE_EKU_ENABLED must be set to YES/TRUE.

By default, the OID 1.3.6.1.5.5.7.3.2 is specified. This OID is used for client authentication through TLS web client.

 **Note:**

The values must be specified as OIDs only. Any other values, in any other format is not recognized and the authentication fails.

For Smart Card Login, specify the OID 1.3.6.1.4.1.311.20.2.2

To specify multiple OIDs, use a **comma (,)** between the OIDs. For example: 1.3.6.1.5.5.7.3.2,1.3.6.1.4.1.311.20.2.2.

 **Note:**

Both the values must be present in the EKU extension of the

Table 22-17 (Cont.) X509CredentialExtractor Step

Parameters	Display Name	Description
		client certificate. If either of these values is not present, the authentication fails.
		See also, Global OID Reference Database for OID details.

22.7.10.2 X509 Steps Orchestration for ECU

Configure the Orchestration steps for the Authorization flow.

1. Click the **Steps Orchestration** subtab
2. From the **InitialStep list**, choose the X509_EKU step.
3. Set **On Success**, **On Failure**, and **On Error** for each of the steps as shown in the following example:

Table 22-18 X509 Step Orchestration for ECU

Name	Description	On Success	On Failure	On Error
X509_EKU	X509CredentialExtra ctor Step	User_Identification	Failure	Failure
User_Identification	UserIdentificationPlu gin Step	Success	Failure	Failure

22.7.10.3 X509 Scheme for ECU

Create a new X509 Authentication Scheme.

To create a new Authentication Scheme:

1. From the **Application Security** Launch Pad, click **Authentication Schemes** under **Access Manager**.
2. From the **Authentication Schemes tab**, click **Create Authentication Scheme**.
3. Set all the parameters as described in [Authentication Schemes and Pages](#). The following figure shows a Sample Authentication Scheme Page:

Figure 22-24 Sample Authentication Scheme Page

Access Manager >

Create Authentication Scheme Authentication Scheme

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme must also include a defined Authentication Module.

*** Name** X509_EKU_Scheme

Description X509 Scheme for EKU

*** Authentication Level** 5

Default

*** Challenge Method** X509

Challenge Redirect URL /oam/serve/

*** Authentication Module** X509

*** Challenge URL** /oam/CredCollectServlet/X509

*** Context Type**

Challenge Parameters

Set As Default Apply

22.7.10.4 Protecting Resources with X509 for EKU Scheme

Complete the configuration for the X509 for EKU, by assigning the X509_EKU_Scheme to the protected resource policy.

1. From the **Application Security** Launch Pad, select **Application Domains** under **Access Manager**.
2. Search and open the required **Application Domain**.
3. Open the **Authentication Policies** tab and click **Protected Resource Policy**.
4. Select the **X509_EKU_Scheme** from the **Authentication Scheme** drop-down list and click **Apply**.

22.8 Deploying and Managing Individual Plug-ins for Authentication

Administrators can create custom plug-ins and add, validate, distribute, and activate them for defining customized multi-step authentication.

This section provides the following topics:

- [About Managing Your Own Authentication Plug-ins](#)
- [Making Custom Authentication Plug-ins Available for Use](#)
- [Checking an Authentication Plug-in's Activation Status](#)
- [Deleting Your Custom Authentication Plug-ins](#)
- [Plug-ins Page](#)
- [Plug-in Details Page](#)

22.8.1 About Managing Your Own Authentication Plug-ins

You can create custom authentication plug-ins and use them to define customized multi-step authentication modules.

Create custom plug-ins as specified in the Developing Custom Authentication Plug-ins in the *Fusion Middleware Developer's Guide for Oracle Access Management*. After development, the plug-in must be deployed on the AdminServer, as a JAR file, which is validated automatically. After validation, you can configure and distribute the plug-in using the Oracle Access Management Console.

The server processes the XML configuration file within the plug-in JAR file to extract data about the plug-in. After the plug-in is imported, you can see and modify the various plug-in states based on information available from the AdminServer.

[Plug-ins Page](#) illustrates the Plug-ins Node under the Common Configuration section of the System Configuration tab. [Plug-in Details Page](#) reflects configuration details for the selected plug-in in the table.

22.8.2 Making Custom Authentication Plug-ins Available for Use

Users with valid Administrator credentials can add, validate, distribute, and activate a custom plug-in.

Prerequisites

Developing a custom plug-in as described in the Developing Custom Authentication Plug-ins in the *Fusion Middleware Developer's Guide for Oracle Access Management*.

1. Import the Plug-in:

- a. In the Oracle Access Management Console, click **Application Security** at the top of the window
- b. In the Application Security Console, click **Authentication Plug-ins** in the **Plug-ins** section.
- c. In the page that appears, click **Import Plug-in**.
- d. In the Import Plugin dialog box, click **Browse** and select your plug-in JAR file.
- e. Review the message in the dialog box, then click **Import**.

2. Configure Parameters: Expand the **Plugin Details** section, click **Configuration Parameters**, and enter appropriate information as needed.

3. Distribute the Plug-in to OAM Servers:

- a. In the Plug-ins table, select the target plug-in.
- b. Click **Distribute Selected**, then check the plug-in's **Activation Status** tab.

4. Activate the Plug-in (and the custom plugin implementation class) so it is ready to be used by OAM Server:

- a. In the Plug-ins table, select the target plug-in.
- b. Click **Activate Selected**, then check the plug-in's Activation Status.

5. Perform the following tasks as needed:

- [Checking an Authentication Plug-in's Activation Status](#)
- [Deleting Your Custom Authentication Plug-ins](#)
- [Creating a Custom Authentication Module using Bundled Plug-ins](#)

22.8.3 Checking an Authentication Plug-in's Activation Status

Users with valid Administrator credentials can activate a custom plug-in and check the status of activation.

Prerequisites

[Making Custom Authentication Plug-ins Available for Use](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window
2. In the Application Security console, click **Authentication Plug-ins** in the **Plug-ins** section.
3. In the Plug-ins table, select the target plug-in.
4. **Server Instance Name:** Expand the Plug-in Details section and click **Activation Status** to display the location and status of the plug-in.
5. Perform the following tasks as needed:
 - [Deleting Your Custom Authentication Plug-ins](#)
 - [Creating a Custom Authentication Module using Bundled Plug-ins](#)

22.8.4 Deleting Your Custom Authentication Plug-ins

Users with valid Administrator credentials can deactivate and then delete a custom plug-in.

When an Administrator deletes a custom authentication plug-in, its name is not removed from the list of plug-ins. To delete the plug-in (for the purpose of re-importing the same plug-in later), the Administration must stop the WebLogic Server and edit the oam-config.xml manually.

Prerequisites

The plug-in must have been added and available in the console

1. In the Oracle Access Management Console, click **Application Security** at the top of the window
2. In the Application Security console, click **Authentication Plug-ins** in the **Plug-ins** section.
3. **Deactivate the Plug-in:** You must perform this before removing a plug-in.
 - a. In the Plug-ins table, select the target plug-in.
 - b. Click **Deactivate Selected**, then check the plug-in's Activation Status.
4. **Delete the Deactivated Plug-in:**
 - a. In the Plug-ins table, select the target plug-in.
 - b. Click **Delete Selected**.
 - c. Stop the WebLogic Administration Server, locate and edit oam-config.xml manually to remove the deactivated plug-in, and then restart the WebLogic Administration Server.
5. Perform the following tasks as needed:

- [Making Custom Authentication Plug-ins Available for Use](#)
- [Creating a Custom Authentication Module using Bundled Plug-ins](#)

22.8.5 Plug-ins Page

Provides information about the Plug-ins Node under the Common Configuration section of the System Configuration tab, and the Plug-ins page.

The Plug-ins page includes a tool bar with command buttons, most of which operate on the plug-in that is selected in the table. The table provides information about the existing custom plug-ins and their state.

Figure 22-25 Plug-ins Page

Plug-ins

Use the following screen to set up custom Plug-ins to extend Authentication functionality for Oracle Access Manager with Oracle Security Token Service.

Row	Plug-in Name	Description	Activation Status	Type	Last updated On	Last updated by
12	SIMFedModelIdentifierPlugIn		Activated	Authentication		
13	BasicFedModelIdentifierPlugIn		Activated	Authentication		
14	SIMBasicFedModelIdentifierPlugIn		Activated	Authentication		
15	FedProgramaticAuthnPlugIn		Activated	Authentication		
16	X509CredentialExtractor		Activated	Authentication		

Total Rows: 36

Administrators control plug-in states using the command buttons, across the table, at the top of the Plug-ins page, as described in [Table 22-19](#).

Table 22-19 Custom Plug-ins Actions

Action Button	Description
Import Plugin...	<p>Uploads the plugin information to the OAM_FILE_ARTIFACTS table in the database and also adds the plug-in JAR file to the AdminServer <code>\$DOMAIN_HOME/oam/plugins</code> and begins plug-in validation.</p> <ul style="list-style-type: none"> • Same JAR Name: If the new plug-in JAR name (in <code>\$DOMAIN_HOME/oam/plugins</code>) matches an existing plug-in JAR name (in <code>\$DOMAIN_HOME/config/fmwconfig/oam/plugins</code>), Oracle Access Manager extracts new configuration metadata from the XML file in the JAR (in <code>\$DOMAIN_HOME/oam/plugins</code>) and checks the version of the new plug-in. • XML Version: If the new plug-in XML version (in <code>\$DOMAIN_HOME/oam/plugins</code>) is greater than the existing XML version (in <code>\$DOMAIN_HOME/config/fmwconfig/oam/plugins</code>), validation is successful. Otherwise, "invalid plugin name with invalid version" is returned and the new plug-in JAR is removed (from <code>\$DOMAIN_HOME/oam/plugins</code>). • Different JAR Name: If the new plug-in JAR name (in <code>\$DOMAIN_HOME/oam/plugins</code>) is different then existing plug-in JAR names (in <code>\$DOMAIN_HOME/config/fmwconfig/oam/plugins</code>), the new plug-in JAR is uploaded and validation is successful. <p>On Success: Status is reported as "Uploaded" (even if an OAM Server is down). If all registered OAM Servers report "Uploaded", then the status on AdminServer is also "Uploaded".</p> <p>See Also: Developing Custom Authentication Plug-ins in the <i>Fusion Middleware Developer's Guide for Oracle Access Management</i>.</p>
Distribute Selected	<p>Downloads the imported plugin from the database to the <code>\$DOMAIN_HOME/config/fmwconfig/oam/plugins</code> directory on the AdminServer only</p> <ul style="list-style-type: none"> • Propagates the plug-in to all registered OAM Servers. • Starts the distribution listener and notification mechanism between AdminServer and OAM Servers • Distributes the plug-in JAR from database to each OAM Server node under <code>\$DOMAIN_HOME/config/fmwconfig/oam/plugins</code> <p>On Success: Status is reported as "Distributed" (even if an OAM Server is down). If all registered OAM Servers report "Distributed", then the status on AdminServer is also "Distributed".</p> <p>On Failure: Status is reported as "Distribution Failed"</p>
Activate Selected	<p>After successful distribution the plug-in can be activated on all registered OAM Servers.</p> <p>Activation:</p> <ul style="list-style-type: none"> • Starts the Message listener and notification mechanism between AdminServer and OAM Servers • AdminServer sends Activate notification to all registered OAM Servers <p>On Success: Status is reported as "Activated" (even if an OAM Server is down). If all registered OAM Servers report "Activated", then the status on AdminServer is also "Activated".</p> <p>On Failure: Status is reported as "Activation Failed"</p> <p>Following activation on all OAM Servers, the plug-in can be used and executed in any authentication module construction or orchestration.</p>

Table 22-19 (Cont.) Custom Plug-ins Actions

Action Button	Description
Deactivate Selected	<p>Following plug-in activation, an Administrator can choose to deactivate the plug-in: if the plug-in is not used in any authentication module or scheme, for example. The selected plug-in from all registered OAM Servers.</p> <p>Deactivate:</p> <ul style="list-style-type: none"> Starts the Distribution listener and notification mechanism between AdminServer and OAM Servers Removes the plug-in JAR from AdminServer and each registered OAM Server (<i>\$DOMAIN_HOME/config/fmwconfig/oam/plugins</i>) AdminServer sends De-activation notification to all registered OAM Servers <p>On Success: Status is reported as "Deactivated" (even if an OAM Server is down). If all registered OAM Servers report "Deactivated" then the status on AdminServer is also "Deactivated". Plug-in configuration is removed from oam-config.xml.</p> <p>Note: After deactivation, the plug-in cannot be used or executed in any authentication module or orchestration.</p> <p>On Failure: Status is reported as "Deactivation Failed"</p>
Remove Selected	<p>Following plug-in deactivation, an Administrator can delete the selected plug-in. During this process, Access Manager:</p> <p>Delete:</p> <ul style="list-style-type: none"> Starts the Distribution listener and notification mechanism between AdminServer and OAM Servers Removes the plug-in JAR from AdminServer and each registered OAM Server (<i>\$DOMAIN_HOME/config/fmwconfig/oam/plugins</i>) <p>On Success: Status is reported as "Removed" (even if an OAM Server is down). Plugin configuration continues to remain in oam-config.xml with the updated status "Removed".</p> <p>On Failure: Status is reported as "Removal Failed"</p>

Table 22-20 describes elements in the Plug-ins status table.

Table 22-20 Plugins Status Table

Element	Description
Plugin Name	Extracted from the Plugin name element of the XML metadata file.
Description	Extracted from the description element of the XML metadata file.
Activation Status	Reported activation status based on information from AdminServer.
Type	Extracted from the type element of the XML metadata file.
Last Updated on	Extracted from the creation date element of the XML metadata file.
Last Updated by	Extracted from the author element of the XML metadata file.

22.8.6 Plug-in Details Page

Provides information about the Plug-in Details page.

In the Plug-in Details section of the page, the Activation Status is maintained by the AdminServer, as shown in [Figure 22-26](#).

Figure 22-26 Plugin Details: Activation Status of Selected Plug-in

Plug-in Details: UserIdentificationPlugIn

Server Instance Name	Plug-in Activation Status
oam_server1	Activated

Depending on your plug-in, various configuration details are extracted from the configuration element of the XML metadata file to populate Configuration Parameters in the Plugin Details section. Examples are shown in [Table 22-21](#); see also, [Table 22-12](#).

Table 22-21 Example of Plugin Details Extracted from XML Metadata File

Configuration Element	Description
DataSource	<pre>- <configuration> - <AttributeValuePair> <Attribute type="string" length="20">DataSource</Attribute> <mandatory>true</mandatory> <instanceOverride>false</instanceOverride> <globalUIOverride>true</globalUIOverride> <value>jdbc/CISCO</value> </AttributeValuePair> </configuration></pre>
Kerberos Details	<p>Defines the following Kerberos details:</p> <p>KEY_KEYTAB_FILE, KEY_PRINCIPAL, KEY_KRB_CONFIG_FILE</p>
User Identification Details	<p>Defines the User Identity Store and LDAP filter parameters. for this plug-in to use:</p> <p>KEY_IDENTITY_STORE_REF, KEY_LDAP_FILTER</p>
User Authentication Details	<p>Defines the User Identity Store for this plug-in to use:</p> <p>KEY_IDENTITY_STORE_REF</p>
X.509 Details	<p>Defines the X.509 certificate details for this plug-in to use:</p> <p>KEY_CERTIFICATE_ATTRIBUTE_TO_EXTRACT, KEY_IS_CERT_VALIDATION_ENABLED</p>

22.9 Managing Authentication Schemes

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 "[Deploying and Managing Individual Plug-ins for Authentication](#)").

When you register a partner (either using the Administration Console or the remote registration tool), the Application Domain that is created is seeded with a policy that uses the authentication scheme that is set as the default scheme. You can choose any of the existing authentication schemes as the default for use during policy creation.

You can also create a new authentication scheme, copy an existing definition to use as a template, modify a definition, or delete the definition. The copy uses a default name that is based on the original. For example, if you copy the scheme named *KerberosScheme*, the copy is named *Copy of KerberosScheme*.

This section is divided into the following topics:

- [Authentication Schemes and Pages](#)
- [Understanding Multi-Level and Step-Up Authentication](#)
- [Creating an Authentication Scheme](#)
- [Viewing, Editing, or Deleting an Authentication Scheme](#)
- [Searching for an Authentication Scheme](#)

22.9.1 Authentication Schemes and Pages

All authentication schemes include the same elements with differing values.

[Figure 22-27](#) shows the default LDAPScheme page as an example.

Figure 22-27 Default LDAPScheme Page

LDAPScheme Authentication Scheme Set As Default Duplicate Apply

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme must also include a defined Authentication Module.

Name LDAPScheme

Description LDAP Scheme

Authentication Level 2

Default

Challenge Method FORM

Challenge Redirect URL /oam/server/

Authentication Module LDAP

Challenge URL /pages/login.jsp

Context Type default

Context Value /oam

Challenge Parameters

[Table 22-22](#) 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 22-22 Authentication Scheme Definition

Element	Description
Name	The unique name for this scheme, which appears in the navigation tree. See Also: " Pre-configured Authentication Schemes "
Description	The optional description, up to 200 characters, that explains the use of this scheme.

Table 22-22 (Cont.) Authentication Scheme Definition

Element	Description
Authentication Level	<p>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.</p> <p>The trust level is expressed as an integer value between 0 (no trust) and 99 (highest level of trust).</p> <p>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 25-1.</p> <p>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.</p> <p>See Also: "About Multi-Level and Step-Up Authentication".</p>
Default	A non-editable box that is checked when the Set as Default button is clicked.
Challenge Method	<p>One challenge method must be selected from those listed as available:</p> <ul style="list-style-type: none"> • Form • Basic (LDAP) • X509 (Certificate) • WNA (Windows Native Authentication) • None • DAP <p>See Also: "Credential Challenge Methods"</p>
Challenge Redirect URL	<p>This URL declares the end point referencing the Credential Collector (ECC or DCC). For example:</p> <p>ECC: <code>/oam/server</code></p> <p>DCC: <code>http://<dcc-host:port>/</code></p> <p>See Also:</p> <ul style="list-style-type: none"> • "About Host Identifiers" • Configuring OAM WebGate and Authentication Policy for DCC
Authentication Module	<p>Identifies the pre-configured authentication module to be used to challenge the user for credentials. The module or plug-in specified here identifies the exact user identity store to be used.</p> <ul style="list-style-type: none"> • FederationMTPlugin • FederationPlugin • Kerberos Plugin (Authentication Modules and Custom Authentication Module) • MTLDAPBasic • MTLDAPPlugin • OIFMTLDAPPlugin • Password Policy Validation Module • TAPModule • X509 Plugin (under the X509 Authentication Modules node) <p>See Also "Managing Native Authentication Modules" and "Orchestrating Multi-Step Authentication with Plug-in Based Modules".</p>

Table 22-22 (Cont.) Authentication Scheme Definition

Element	Description
Challenge URL	<p>This URL is associated with the designated Challenge Method (FORM, for instance).</p> <p>FORM-based, out of the box authentication scheme (LDAPScheme and LDAPNoPasswordValidationScheme), Challenge URL is "/pages/login.jsp". The context type and context values are used to build the final URL.</p> <p>X509-based Challenge URL takes the form: <code>https://<i>managed_server_host</i>:<i>managed_server_ssl_port</i>/oam/CredCollectServlet/X509</code></p> <p>Note: The default Challenge URL is based on the credential collector embedded with the OAM Server (ECC). If you are using detached credential collector-enabled Webgate and related DCC pages installed with WebGate, see Configuring OAM WebGate and Authentication Policy for DCC.</p>
Challenge Parameters	Supported challenge parameters are discussed in " Challenge Parameters for Authentication Schemes ".
For schemes using Challenge Method FORM, X509, or DAP	Only Schemes with the Challenge Method of FORM, X509, or DAP include the following additional elements. Other schemes use defaults that require no change.

Table 22-22 (Cont.) Authentication Scheme Definition

Element	Description
Context Type	<p>Used to build the final URL for the Embedded Credential Collector (ECC only, DCC does not use this) based on the following possible values:</p> <ul style="list-style-type: none"> 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 by the ECC. 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: <pre>challenge_url = "/customlogin.jsp" contextType="customWar" contextValue="/contextroot of custom application"</pre> 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: <pre>challenge_url = "/CustomReadServlet" contextType="customHtml" contextValue="html file location"</pre> 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: <pre>challenge_url = "http://host:port/externallogin.jsp" contextType="external" contextValue=Not applicable</pre> <p>See Also: "About Custom Login Pages" and Managing Global Password Policy</p>
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 22-22](#). All custom login pages must meet the following requirements:

- Custom login pages require two form fields (username and password). Access Manager supports custom forms as described in *Developing Applications with Oracle Access Management*.
- 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 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](#)
- [Credential Challenge Methods](#)
- [Challenge Parameters for Authentication Schemes](#)



See Also:

Developing Applications with Oracle Access Management for details about customizing login pages and messages.

22.9.1.1 Pre-configured Authentication Schemes

Pre-configured Authentication Schemes such as BasicFAScheme and KerberosScheme are available with Access Manager.

"[Table 22-23](#)" identifies the pre-configured authentication schemes available with Access Manager and some specific details of each. For more information about challenge parameters, see "[Table 22-23](#)".

Table 22-23 Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
AnonymousScheme	Authentication Level: 0 Challenge Method: None Authentication Module: AnonymousModule	Leaves unprotected specific 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.
BasicFAScheme Only for Oracle Fusion Applications	For Fusion Applications	For specific information about this authentication scheme, refer to the Oracle Fusion Applications Technology Library located on the Oracle Technology Network (OTN) web site: http://www.oracle.com/technetwork

Table 22-23 (Cont.) Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
BasicScheme	Authentication Level: 1 Challenge Method: Basic Authentication Module: LDAP	Protects 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.
FAAuthScheme Only for Oracle Fusion Applications	Authentication Level: 2 Challenge Method: FORM Authentication Module: LDAP Context: customWar Context Value: / fusion_apps	For specific information about this authentication scheme, refer to the Oracle Fusion Applications Technology Library located on the Oracle Technology Network (OTN) web site: http://www.oracle.com/technetwork
FederationMTScheme Only for Oracle Fusion Applications	Authentication Level: 2 Challenge Method: FORM Authentication Module: FederationMT Plugin Context Type: customWar Context Value: / fusion_apps Challenge Parameters: initial_command=NONE is_rsa=true	For specific information about this authentication scheme, refer to the Oracle Fusion Applications Technology Library located on the Oracle Technology Network (OTN) web site: http://www.oracle.com/technetwork See Also: " Challenge Parameters for Authentication Schemes ".

Table 22-23 (Cont.) Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
FederationScheme Only for Identity Federation Federation 11.1.2.	Authentication Level: 2 Challenge Method: FORM Authentication Module: FederationPlugin Context Type: customWar Context Value: /oam Challenge Parameters: initial_command=NONE is_rsa=true	See Also: Managing Oracle Access Management Identity Federation .
Kerberos Scheme	Authentication Level: 2 Challenge Method: WNA Authentication Module: Kerberos Context Type: customWar Context Value: /fusion_apps	Protects Access Manager-related resources (URLs) for most directory types based on a Windows Native Authentication challenge method and valid WNA credentials in Active Directory.

Table 22-23 (Cont.) Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
LDAPNo Password Validation Scheme	<p>Authentication Level: 2</p> <p>Challenge Method: FORM</p> <p>Authentication Module: LDAPNoPasswordAuthModule</p> <p>Context Type: default</p> <p>Context Value: /oam</p> <p>Note: LDAPNoPasswordAuthModule is similar to the DAP (asserter) mechanism.</p>	<p>Protects Access Manager-related resources (URLs) for most directory types based on a form challenge method.</p> <p>Used with the Identity Asserter for SSO when you have resources in a WebLogic Container. See <i>Securing Applications with Oracle Platform Security Services</i>.</p>
LDAPScheme	<p>Authentication Level: 2</p> <p>Challenge Method: FORM</p> <p>Authentication Module: LDAP</p> <p>Context Type: customWar</p> <p>Context Value: /fusion_apps</p>	<p>Protects Access Manager-related resources (URLs) for most directory types based on a form challenge method.</p>
<p>enableCoexistMode and disableCoexistMode in the Oracle Fusion Middleware WebLogic Scripting Tool Command Reference.</p>		

Table 22-23 (Cont.) Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
OAMAdminConsoleScheme	<p>Authentication Level: 2</p> <p>Challenge Method: FORM</p> <p>Authentication Module: LDAP</p> <p>Context Type: default</p> <p>Context Value: /oam</p>	Authentication scheme for Oracle Access Management Console.
OIFScheme	<p>Authentication Level: 2</p> <p>Challenge Method: DAP</p> <p>Authentication Module: DAP</p> <p>Context Type: External</p>	<p>This scheme delegates authentication to OIF, after which, Oracle Identity Federation sends back a token that is asserted by the OAM Server.</p> <p>The Delegated Authentication Protocol (DAP) challenge method is used to delegate authentication to a third-party (OIF in this case).</p> <p>Challenge Parameters: TAPPartnerId=OIFDAPPartner</p> <p>See Also: "Challenge Parameters for Authentication Schemes".</p>
For Identity Federation 11.1.1, use FederationScheme		
PasswordPolicyValidationScheme	<p>Authentication Level: 2</p> <p>Challenge Method: FORM</p> <p>Authentication Module: Password Policy Validation Module</p> <p>Context: External</p>	Enables password policy evaluation.
TAPResponseOnlyScheme	<p>Authentication Level: 2</p> <p>Challenge Method: DAP</p> <p>Authentication Module: DAP</p>	

Table 22-23 (Cont.) Pre-configured Authentication Schemes

Scheme Name	Specifications	Purpose
		<ol style="list-style-type: none"> 1. From the IAM Suite Application Domain, Protected Higher Level 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. <p>Challenge Parameters: TAPPartnerId=TAPPartnerName</p>
X509Scheme	Authentication Level: 5 Challenge Method: X509 Authentication Module: X509	<p>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.</p> <p>Note: This scheme relies on SSL to deliver the use's X.509 certificate to the OAM Server.</p>

22.9.1.2 Credential 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.

Access Manager provides the following credential challenge methods for use in an authentication scheme:

- [FORM](#)
- [BASIC](#)
- [X509](#)
- [WNA](#)
- [NONE](#)
- [DAP](#)

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. 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. Thus, a Basic challenge relies on the LDAP Authentication Module and user identity store associated with that module.

 **Note:**

If a URL is protected by Access Manager using Basic Authentication with OID configured as the identity store, OID defined users can not log in. To resolve this, add the following line before the closing `</security-configuration>` tag in the `config.xml` file.

```
<enforce-valid-basic-auth-credentials>>false
  </enforce-valid-basic-auth-credentials>
```

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 Access Manager:

- 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:**

[Configuring Access Manager for Windows Native Authentication](#) for details about integration with 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 Access Manager-specific URLs that you do not want to protect.

DAP

The Delegated Authentication Protocol (DAP) challenge method is required for OIFScheme (Oracle Identity Federation 11.1.1 integration) with the DAP authentication module and external context type ([Table 22-22](#)). The DAP challenge mechanism indicates that Access Manager does an assertion of the token that it receives, which differs from the standard challenge "FORM" mechanism with the external option.

DAPModule is an assertion module, though it is specialized for this one application and does not appear in the list of Authentication Modules in the Oracle Access Management Console.

The DAP challenge mechanism delegates authentication to a third party (Identity Federation in this case). The challenge_url points to the Identity Federation Server URL. When a resource is protected by this scheme, the OAM Server redirects to the Identity Federation Server URL for credential collection. OAM Server does not perform the credential collection or validation in this case. 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. Access Manager receives and decrypts this token, checks whether the user is a valid user in the default identity store for Oracle Access Management. If the user is valid, Access Manager gives access to the resource.

The DAPToken is encrypted and decrypted with a key that is shared between Access Manager and Identity Federation. The DAPToken is built from the Access Manager side.

The 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 Access Manager and Identity Federation. Access Manager provides a WLST command (`registerOIFDAPPartner`), that takes the keystore location generated by the Identity Federation store and retrieves the keys and stores it on the Identity Federation side.

22.9.1.3 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.

The syntax for specifying any challenge parameter is:

```
<parametername>=<value>
```

This syntax is not specific to any Webgate release. Authentication schemes are independent of Webgate release.

[Table 22-24](#) identifies the pre-configured schemes with challenge parameters.

Table 22-24 Challenge Parameters in Pre-configured Schemes

Pre-configured Schemes	Challenge Parameter
BasicSessionlessScheme	CookieLessMode=true Primarily used for clients that do not support URL redirect or cookies.
FederationMTScheme	initial_command=NONE Primarily used for Fusion Applications that support multiple factor authentication. is_rsa=true Used with RSA multi-step authentication, as described in Integrating RSA SecurID Authentication with Access Manager and Process Overview: Multi-Step Authentication <i>Developing Applications with Oracle Access Management</i> .
FederationScheme For Identity Federation 11.1.2 only. Use OIFScheme for Oracle Identity Federation 11.1.1.	Primarily used for clients that do not support URL redirect or cookies. Context Value: /fusion_apps Challenge Parameters: initial_command=NONE is_rsa=true
OIFScheme For Oracle Identity Federation 11.1.1 only. Use FederationScheme for Identity Federation 11.1.2.	Primarily used for clients that do not support URL redirect or cookies. TAPPartnerId=OIFDAPartner This scheme delegates authentication to Oracle Identity Federation 11.1.1, after which, Federation sends back a token that is asserted by the OAM Server.
TapScheme	TAPPartnerId=TAPPartnerName

An authentication scheme can collect context-specific information before submitting the request to the Access Server. Context-specific information can be in the form of an external call for information. [Table 22-25](#) lists user-defined challenge parameters you can use in Authentication Schemes.

Table 22-25 User-Defined Challenge Parameters for Authentication Schemes

Challenge Parameter	Definition
initial_command =NONE	<p>Required to enable the plug-in to indicate which credentials are to be collected.</p> <p>For example, for Form-based authentication, the framework typically expects to collect "username" and "password" (submitted from the login page). However, you might want credentials from different fields of the login page; "form_username" and "form_password" for example. Setting this challenge parameter shifts initial control from the login page to the plug-in, which decides the parameters to collect from the login page then appropriately forwards or redirects to the page.</p> <p>Default: blank (not set)</p>
action=	<p>The actions parameter identifies the URL to which the HTML form is posting when you do not want to use the hard coded ECC default <code>/oam/server/auth_cred_submit</code>.</p> <p>Note: ECC does not use the <code>action=</code> parameter. When the <code>action=</code> challenge parameter is not specified, both the DCC and ECC use the default: <code>/oam/server/auth_cred_submit</code>.</p> <p>See Also: "Configuring the PasswordPolicyValidationScheme "</p>
creds= DCC Only	<p>Supported by the detached credential collector (DCC) only.</p> <p>In the following 11g example, username and password are the names of relevant fields in the login form:</p> <pre>creds=username password</pre> <p>NOTE: Format of this challenge parameter has changed since the 10g release.</p> <p>The Web server source (server parameter) takes precedence over other sources. This prevents the request data, which is under control of the user, from overriding Web server data. For example, a <code>remote_user</code> cookie sent from a user will not override a <code>remote_user</code> variable set by the Web server</p> <p>Generally, when the user submits a login form that is protected by an authentication scheme with a Form-based challenge method, the DCC processes the credentials that were specified with this <code>creds=</code> parameter.</p> <p>For forms using <code>METHOD=POST</code> processing, the browser sends a POST request to the Web server with the credential data from the form in the body of the request. If the form uses <code>METHOD=GET</code>, the browser sends a GET request with query string parameters with the same names as those specified on the <code>creds</code> parameter. Oracle recommends that you use POST processing, if possible.</p> <p>Note: You can specify the <code>creds</code> parameter with the other types of challenge methods. For a plug-in to make use of the <code>creds</code> parameter, you specify what is passed in the <code>obMap</code> credentials parameter of the <code>ObUserSession</code> object, as described in the <i>Developing Applications with Oracle Access Management</i>.</p> <p>See Also: "Configuring the PasswordPolicyValidationScheme "</p>
extracreds= DCC Only	<p>Supported by the DCC only. Specifies optional parameters which, if present, are made available to the authentication plug-in for collection during each iteration of a multi-step authentication using the DCC.</p> <p>The <code>extracreds</code> parameter uses the same syntax as the <code>creds</code> parameter: <code>extracreds=separated qualified or unqualified names [{any cookie header server query post}:] <name></code>. However, the value <code>any</code> is used by <code>extracreds</code> only. For example:</p> <pre>extracreds=[{any cookie header server query post}:] <name></pre> <p>See Also: "Configuring the PasswordPolicyValidationScheme "</p>

Table 22-25 (Cont.) User-Defined Challenge Parameters for Authentication Schemes

Challenge Parameter	Definition
OverrideRetryLimit=0	<p>The number of tries that can override the RetryLimit for login.</p> <p>The value must be a positive integer.</p> <p>A value of zero (0) disables this function.</p> <p>See Also: "Configuring the PasswordPolicyValidationScheme "</p>
ChallengeRedirectMethod	<p>Authentication POST data preservation parameter for both the embedded credential collector (ECC) and the detached credential collector (DCC).</p> <p>Value: GET POST DYNAMIC</p> <p>Note: Preference is given first to the Authentication Scheme containing this parameter; second to the Webgate providing this user defined parameter. Otherwise, default behavior is Dynamic.</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>
MaxPreservedPostDataBytes	<p>Configure this Authentication Scheme challenge parameter (or user-defined Webgate parameter) for authentication POST-data preservation.</p> <p>Default: 8192 bytes</p> <p>Note: Preference is given first to the Authentication Scheme containing this parameter; second to the Webgate providing this user-defined parameter. Otherwise, default behavior is 8192 bytes.</p> <p>This parameter defines the maximum length of POST data that Webgate can preserve. If the size of inbound raw user POST data (or encrypted post data after processing), crosses this limit, POST data is dropped and the existing authentication flow continues. The event is logged as usual.</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>
MaxPostDataBytes= DCC Only	<p>Configure this Authentication Scheme challenge parameter to restrict the maximum number of bytes of POST data that is submitted as user credentials and sent to the OAM Server.</p> <p>Configure this challenge parameter for POST-data preservation by the DCC only to limit the maximum size of the POST data that can be posted as credentials on the form and sent to the OAM Server. DCC compares the value of the content-length header with the limit set.</p> <p>Default: 8192 bytes</p> <p>This challenge parameter requires a positive integer value.</p> <p>See Also: Table 15-2 "Configuring the PasswordPolicyValidationScheme " "Configuring Authentication POST Data Handling"</p>

Table 22-25 (Cont.) User-Defined Challenge Parameters for Authentication Schemes

Challenge Parameter	Definition
ssoCookie=	<p>Controls the OAMAuthnCookie cookie, as described in "Configuring Challenge Parameters for Encrypted Cookies".</p> <p>Default: ssoCookie=httponly ssoCookie=Secure</p> <p>Disable either setting: ssoCookie=disablehttponly ssoCookie=disableSecure</p> <p>Note: These parameters are configured differently depending on your credential collector configuration.</p> <ul style="list-style-type: none"> For detached credential collector-enabled OAM Webgates, set these parameters directly in the agent registration page. For non-DCC agents (Resource Webgates), these parameters are configured through user-defined challenge parameters in authentication schemes. <p>See Also: Table 22-32</p>
miscCookies=	<p>Controls other miscellaneous Access Manager internal cookies. By default, httponly is enabled for all other (miscellaneous) cookies.</p> <p>Default: miscCookies=httponly miscCookies=Secure</p> <p>Disable either setting: miscCookies=disablehttponly miscCookies=disableSecure</p> <p>Note: These parameters are configured differently depending on your credential collector configuration.</p> <ul style="list-style-type: none"> For detached credential collector-enabled Webgates, set these parameters directly in the agent registration page. For non-DCC agents (Resource Webgates), these parameters are configured through challenge parameters of the same name. <p>See Also: Table 22-32 "Configuring the PasswordPolicyValidationScheme "</p>
DCCctxCookieMaxLength= DCC Only	<p>Defines the maximum length of the DCC cookie. Default: 4096</p> <p>See Also: TempStateMode in this table for more information.</p>

Table 22-25 (Cont.) User-Defined Challenge Parameters for Authentication Schemes

Challenge Parameter	Definition
TempStateMode =	<p>Controls how the DCC stores the OAM Server state (cookie or form) as specified with the parameter's value:</p> <ul style="list-style-type: none"> form: This is the default, and is required for retaining authentication POST data. The OAM Server state stored and passed through the form parameter "OAM_REQ", to avoid the case when the OAM Server configuration <code>serverRequestCacheType=COOKIE</code>, bloated server state causes DCCtxCookie to explode beyond limit resulting in incorrect behavior. The cookie cache mode can be changed to <code>FORM</code> mode from default <code>COOKIE</code> mode. <code>FORM</code> mode works with long URLs. The only difference in behavior is for programmatic authentication, which requires a proper form Submit to pass the <code>OAM_REQ</code> parameter set to the form. Custom credential collection pages need to handle the <code>OAM_REQ</code> parameter that is submitted with the form. cookie: Adding this parameter and value stores the OAM Server state through part of the DCCtxCookie (<code>encdata=... svrctx=...</code>). However, when <code>serverRequestCacheType=COOKIE</code> or <code>=FORM</code>, this could cause incorrect behavior if the resulting cookie length is beyond browser limit. <p>Note:</p> <ul style="list-style-type: none"> When <code>serverRequestCacheType=COOKIE</code>, Oracle recommends <code>TempStateMode=form</code>. When <code>serverRequestCacheType=BASIC</code>, either mode is fine. <p>To update <code>serverRequestCacheType</code>, use the WLST command <code>configRequestCacheType</code> as described in the <i>WLST Command Reference for WebLogic Server</i>. Editing <code>serverRequestCacheType</code> is not supported using the Oracle Access Management Console.</p> <p>With ECC: The <code>serverRequestCacheType</code> dictates whether OAM Server stores its state in memory (BASIC) or not (FORM or COOKIE). <code>serverRequestCacheType = COOKIE</code> or <code>FORM</code> only makes difference when ECC is used. OAM Server stores its state in a request token, which ECC keeps in a cookie or hidden form field as specified with the parameter: <code>serverRequestCacheType=COOKIE</code>, for example.</p> <p>With DCC: There is no difference between <code>serverRequestCacheType=COOKIE</code> or <code>FORM</code>. <code>TempStateMode</code> controls how the DCC stores the OAM Server state (cookie or form) as specified with the parameter's value: <code>TempStateMode=cookie</code>, for example. With the DCC, POST data restoration with a Form-based Authentication Scheme requires the challenge parameter TempStateMode=form.</p> <p>See Also:</p> <ul style="list-style-type: none"> Configuring OAM WebGate and Authentication Policy for DCC Table 15-2 "Table 22-34" "Configuring the PasswordPolicyValidationScheme "
allowedAccessGateList=	<p>Authentication Scheme challenge parameter configured with SPACE separated list of WebGate IDs defining those WebGates that are allowed to enforce authentication by this scheme. For example:</p> <pre>allowedAccessGateList=WebgateID1 WebgateID2</pre>
TunneledUrls	<ul style="list-style-type: none"> For OAM : <code>TunneledUrls=/oam</code> For OIF : <code>TunneledUrls=/oamfed</code> For OIG : <code>TunneledUrls=/oam</code>

22.9.2 Understanding Multi-Level and Step-Up Authentication

This section provides the following topics:

- [About Multi-Level and Step-Up Authentication](#)
- [Changing Security Level of an Authentication Scheme during the Authentication Process](#)

22.9.2.1 About Multi-Level and Step-Up Authentication

Every authentication scheme requires a strength level. The higher the number, the more secure the authentication mechanism; the lower the number, the less stringent the scheme.

For example:

- LDAPScheme authLevel=1
- KerbScheme authLevel=3

**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).

Access Manager policies allow different resources of the same application to be protected with different authentication levels.

Agent types redirect the user to 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.

-

22.9.2.2 Changing Security Level of an Authentication Scheme during the Authentication Process

You can write a custom plug-in to change the security level of an authentication scheme during the authentication process.

In some cases, you may want to increase the security level of an authentication scheme during the authentication process depending on certain conditions. You may want the security level of an authentication scheme to depend on the application the user logged in from. For example, Active Directory and a reverse proxy are among the sources your users can log in from. During the authentication process, you may want to dynamically set one authentication security level to be used for users who log in from Active Directory and another security level to be used for users who log in from the reverse proxy.

Enable custom authentication plug-ins to set the authentication level as a plug-in response. Write a new custom authentication plug-in where a configurable authentication level is set as a plug-in step parameter in the plug-in response. For example, `PLUGIN_AUTHN_LEVEL` is configured in the plug-in response. It helps set the authentication level dynamically based on the authentication mechanism used.

To avoid any security implications resulting from the custom plug-in, use an optional authentication scheme challenge parameter, `MAX_AUTHN_LEVEL`. The value of `MAX_AUTHN_LEVEL` is the maximum authentication level that can be set by a custom authentication plug-in protecting resources. In the custom authentication scheme, this parameter is disabled by default. You need to set this mandatory parameter with a higher value than the `PLUGIN_AUTHN_LEVEL` for the plug-in to dynamically change the value of authentication level during the authentication process.

Write a custom authentication plug-in and configure `KEY_AUTHN_LEVEL` with a value lower than `MAX_AUTHN_LEVEL`. Create a custom authentication module that uses the new authentication plug-in and associate it with the Custom Authentication Scheme. Specify the Scheme in Authentication policy protecting the resource. When the user session is created, warning message about the plug-in overriding the maximum authentication value set by the administrator is logged by OAM server. When `MAX_AUTHN_LEVEL` is not configured or the plug-in tries to set value greater than `MAX_AUTHN_LEVEL`, the authentication succeeds with the authentication level set in the authentication scheme.

Following is the sample code to set authentication level as a plug-in response:

```
String stepName =
context.getStringAttribute(PluginConstants.KEY_STEP_NAME);
String pluginLevel = PlugInUtil.getFlowParam(stepName, "
PLUGIN_AUTH_LEVEL ", context);
PluginResponse rsp = new PluginResponse();
rsp.setName(PluginConstants.KEY_AUTHN_LEVEL);
rsp.setType(PluginAttributeContextType.LITERAL);
rsp.setValue(pluginLevel);
context.addResponse(rsp);
```

22.9.3 Creating an Authentication Scheme

Users with valid Administrator credentials can add a new authentication scheme for use in an Application Domain.

Prerequisites

The authentication module must be defined and ready to use as described in ["Deploying and Managing Individual Plug-ins for Authentication"](#).

 **See Also:**

- ["Authentication Schemes and Pages"](#)
- [Configuring OAM WebGate and Authentication Policy for DCC](#) if needed

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security Console, click **Authentication Schemes** in the **Access Manager** section.
3. Click the **Create Authentication Scheme**.
4. Fill in the fresh Authentication Scheme page ([Table 22-22](#)) by supplying information based on your deployment:
 - a. Name: *LDAPSimpleFormScheme*
 - b. Authentication Level
 - c. Challenge Method: FORM
 - d. Challenge Redirect URL: *http://CredentialCollectorhost:port*
 - e. Authentication Module: LDAP
 - f. Challenge URL: */CredentialCollector/loginform...*
 - g. Challenge Parameters: [Table 22-24](#), [Table 22-25](#), [Table 22-32](#)
 - h. Context Type
5. Click **Apply** to submit the new scheme (or close the page without applying changes).
6. Dismiss the Confirmation window.
7. Optional: Click the **Set as Default** button to automatically use this with new Application Domains, then close the Confirmation window.
8. Confirm the new scheme appears in the list of schemes (refresh if needed).
9. Proceed to ["Defining Authentication Policies for Specific Resources"](#).

22.9.4 Searching for an Authentication Scheme

Users with valid Administrator credentials can search for a specific authentication scheme.

 **See Also:**

- ["SSO Agent Search Page"](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security Console, click **Authentication Schemes** in the **Access Manager** section.

3. In the **Name** field, enter the target 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.

22.9.5 Viewing, Editing, or Deleting an Authentication Scheme

Users with valid Administrator credentials can view or modify an existing authentication scheme.



Note:

During a delete operation, if the Authentication Scheme is associated with any authentication policy, she is prompted with association details. Without policy associations, the scheme is deleted.



See Also:

- ["Authentication Schemes and Pages"](#)
- ["Configuring the PasswordPolicyValidationScheme "](#)

1. Search for the target scheme, as described in the previous section.
2. In the list of search results, select the target scheme and click **Edit**.
3. **Edit:**
 - a. On the Authentication Scheme page, modify values for your environment ([Table 22-22](#)).
 - 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 scheme when creating policies in fresh Application Domains, then close the Confirmation window.
5. **Delete:**
 - a. Review any Application Domain using this authentication scheme and assign a different scheme.

- b. Review the Authentication Scheme page to confirm this is the scheme to remove, then close the page.
- c. In the navigation tree, click the name of the scheme and then click the **Delete** button in the tool bar.
- d. Confirm removal (or dismiss the Confirmation window).

22.10 Extending Authentication Schemes with Advanced Rules

Advanced Rules have been added to allow for extending an existing authentication policy.

Both Pre-Authentication and Post-Authentication rules can be applied although the following configurations are not supported by Post-Authentication Rules.

- Two or more resources front ended by the same OHS/WebGate and protected by the same Authentication Scheme.
- A Post-Authentication rule configured for one of the resources defined in step up authentication.
- A user accesses a resource for which no Post-Authentication rule is configured followed by a resource for which a Post-Authentication rule is configured for step up authentication. In this case, the Post-Authentication rule configured for the resource is not effective.



Note:

Advanced Rules are part of the Adaptive Authentication Service for which a license is required. See [About Adaptive Authentication Service](#).

Advanced Rules contain Boolean expressions. If there is more than one triggered outcome to an Authentication Scheme, the lowest execution order outcome will be chosen as the final outcome. [Table 22-26](#) documents the attributes that need be defined when creating an Advanced Rule.

Table 22-26 Advanced Rules Attributes

Name	Description
Name	AuthnRule name. Name has to be unique within the checkpoint
Description	Description of the rule
Execution Order	Order in which the outcome will be executed in cases of more than 1 outcome
Condition	Script; the user can configure condition based on the HTTP request header's availability and set the desired outcome

Table 22-26 (Cont.) Advanced Rules Attributes

Name	Description
Outcome	ID of the Authentication Scheme to which the rule applies. Access / Deny.

**Note:**

If the **Deny Access** option is selected, an error message, The requested URL was not found, is displayed.

See the following sections for details.

- [Advanced Rules Use Cases](#)
- [Context Data for Advanced Rules](#)

22.10.1 Advanced Rules Use Cases

You can configure advance rules for certain scenarios.

For the following use cases, configure Advanced Rules:

- **Non Browser Client** - For user authentication, a form-based login page is presented through the browser for the user to complete. In some cases, a non-browser client (switches, routers and the like) might need to do basic authentication based on credentials passed via the request header. Non-browser client authentication can be configured as a pre-authentication Advanced Rule only. To support non-browser client authentication, configure the desired condition in an Authentication Rule (based on the HTTP request header's availability) and set the desired outcome.
- **Windows Native Authentication Option** - An Advanced Rule can be configured to allow for switching between Windows Native Authentication (WNA) and form-based user authentication depending on whether the user comes thru VPN or a corporate network.
- **User Authentication Scheme Option** - An Advanced Rule can be configured to allow administrator to choose the method of users authentication. The choice would be passed as stored in users attribute from custom LDAP attribute.
- **Second Factor Authentication** - An Advanced Rule can be configured to allow for Second Factor Authentication (SFA) based on defined user or request attributes. For details on SFA, see [Introducing the Adaptive Authentication Service](#) .

Table 22-27 contains examples of how the conditions might be configured in these Advanced Rules use cases.

Table 22-27 Sample Advanced Rules

Sample Rule	Sample Jython Script-based Condition	Notes
Switching authentication scheme based on private or public IP rule	<code>location.clientIP.startswith('10.') or location.clientIP.startswith('172.16') or location.clientIP.startswith('192.168')</code>	This rule can be used in Pre and Post authentication checkpoints
Black listed IP	<code>location.clientIP in ['130.35.50.115', '130.35.50.112', '130.35.50.113']</code>	This rule can be used in Pre and Post authentication checkpoints
Client Browser Type	<code>request.userAgent.lower().find('firefox') > 0</code>	This rule can be used in Pre and Post authentication checkpoints
Blocking access to user having user attribute 'description' equals 'test'	<code>user.userMap['description'] == 'test'</code>	This rule can be used only in Post authentication checkpoints
Non browser client	<code>request.authorization.lower().startswith('basic')</code>	This rule can be used only in Pre authentication checkpoints
Customer HTTP Header value	<code>request.requestMap['param'] == 'test'</code>	This rule can be used in Pre and Post authentication checkpoints
Switching authentication scheme based on IP address in range	<code>location.isIPinRange('192.35.50.180','192.35.50.188')</code>	This rule can be used in Pre and Post authentication checkpoints

22.10.2 Context Data for Advanced Rules

Before executing the Authentication Condition, the Access Manager server prepares a request context using the available data (to construct a Boolean expression based condition).

The following tables describe the various context data details.

- [Table 22-28](#)
- [Table 22-29](#)
- [Table 22-30](#)
- [Table 22-31](#)

Table 22-28 Request Context Data

Attribute Name	Description
<code>requestMap</code>	Map of all the request headers, parameters and post data values. This example can get the custom-header key from request header and compare it with value 'test'. <code>request.requestMap['custom-header'].lower().find('test') > 0</code>
<code>resourceMap</code>	Map of matched resource details
<code>accept</code>	Returns 'Accept' header value
<code>acceptCharset</code>	Returns 'Accept-Charset' header value
<code>acceptEncoding</code>	Returns 'Accept-Encoding' header value

Table 22-28 (Cont.) Request Context Data

Attribute Name	Description
acceptLanguage	Returns 'Accept-Language' header value
authorization	Returns 'Authorization' header value
connection	Returns 'Connection' header value
contentLength	Returns 'ContentLength' header value
cookie	Returns 'Cookie' header value
host	Returns 'Host' header value
ifModifiedSince	Returns 'ifModifiedSince' header value
pragma	Returns 'Pragma' header value
referer	Returns 'Referer' header value
userAgent	Returns 'UserAgent' header value
resourceHost	Returns matched Resource's Host value
resourcePort	Returns matched Resource's Port value
resourceOperation	Returns matched Resource's Operation value
resourceQueryString	Returns matched Resource's QueryString
resourceName	Returns matched Resource's name
resourceType	Returns matched Resource's Type
resourceURL	Returns matched Resource's URL; for example, if 'landingPage' is in request.resourceURL, condition will evaluate to true if resourceURL has landingPage in it.
isIPinRange('start IP' , 'end IP')	Evaluates to true if location.clientIP is in the specified range. Example: <code>location.isIPinRange('192.35.50.180', '192.35.50.188')</code>

Table 22-29 Location Context Data

Attribute Name	Description
locationMap	Map of all the location data values; for example: <code>location.locationMap['CLIENT_IP'] == '10.1.23.4'</code>
clientIP	Returns client IP address; for example: <code>location.clientIP.startswith('10.2')</code>
proxyIP	Returns Proxy IP address

Table 22-30 Session Context Data

Attribute Name	Description
sessionMap	Map of all the session data values; for example: <code>session.sessionMap['count'] > 2;</code>
count	Returns number of sessions for the current user; for example: <code>session.count > 2</code>

Table 22-31 User Context Data

Attribute Name	Description
userMap	Map of all the user profile data; for example: user.userMap['email'] == 'john.joe@example.com'

22.11 Configuring Challenge Parameters for Encrypted Cookies

OAM provides challenge parameters that you can use within any authentication scheme to control flags of the encrypted cookies.

- [Challenge Parameters for Encrypted Cookies](#)
- [Configuring Challenge Parameters for Security of Encrypted Cookies](#)
- [Setting Challenge Parameters for Persistence of Encrypted Cookies](#)

22.11.1 Challenge Parameters for Encrypted Cookies

In addition to the OAM Server cookie (OAM_ID), Access Manager implements single sign-on through an encrypted cookie

- **OAM 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.

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.

 **Note:**

The value of the challenge parameter is not case sensitive. Syntax is the same regardless of your Webgate release. A single value is specified after the equal sign (=):

```
ssoCookie=value
```

Multiple values must be separated by a semicolon (;). For example:

```
ssoCookie=value1;value2;...
```

- For detached credential collector-enabled Webgates, set these parameters directly in the agent registration page (Table 15-2).
- For non-DCC agents (Resource Webgates), these parameters are configured through Authentication Scheme challenge parameters (Table 22-32).

Table 22-32 describes specific challenge parameters that control how Webgates set encrypted cookie flags for single sign-on.

Table 22-32 Challenge Parameters for 11g Encrypted Cookies

OAM Webgate Challenge Parameter Syntax for Encrypted Cookies	Description
ssoCookie=	Parameter that controls flags for the SSO cookie OAMAuthnCookie.
miscCookies=	Parameter that controls flags for all other Access Manager encrypted cookies.
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=Secure miscCookies=Secure
disableSecure	Explicitly disables Secure cookies. ssoCookie=disableSecure miscCookies=disableSecure
httponly	Enabled by default with OAM Webgate SSO OAMAuthnCookie and miscellaneous cookies. ssoCookie=httponly miscCookies=httponly
disablehttponly	Explicitly disables httponly functionality, making the encrypted cookies accessible to client-side scripts. ssoCookie=disablehttponly miscCookies=disablehttponly

Table 22-32 (Cont.) Challenge Parameters for 11g Encrypted Cookies

OAM Webgate Challenge Parameter Syntax for Encrypted Cookies	Description
<code>ssoCookie=max-age=<i>time-in-seconds</i></code>	<p>Creates a persistent cookie in browsers, rather than one that lasts for a single session, and specifies the time interval <i>in-seconds</i> when the cookie expires.</p> <p>For example, to set the cookie to expire in 30 days (2592000 seconds):</p> <pre>max-age=2592000</pre>

22.11.2 Configuring Challenge Parameters for Security of Encrypted Cookies

The challenge parameter is not case sensitive.



See Also:

"Creating an Authentication Scheme"

1. Create an authentication scheme.
2. In the Challenge Parameter field, enter your specification for the desired encrypted cookies (Table 22-32).
3. Confirm that the OAM Servers and clients (OAM Agents) are communicating securely across the Oracle Access Protocol channel, as described in [Securing Communication](#).

22.11.3 Setting Challenge Parameters for Persistence of Encrypted Cookies

The challenge parameter is not case sensitive.



See Also:

"Creating an Authentication Scheme"

1. Define an authentication scheme.
2. In the challenge parameter for this scheme, add the following (Table 22-32):

```
WebGate ssoCookie=max-age=time-in-seconds
```

22.12 Configuring Authentication POST Data Handling

Post data preservation and restoration functions apply to both credential collectors (ECC or DCC).

This section provides the following topics:

- [Authentication POST Data Preservation and Restoration](#)
- [Authentication POST Data Handling](#)
- [Configuring Authentication POST Data Handling](#)
- [Testing POST Data Handling Configuration](#)

22.12.1 Authentication POST Data Preservation and Restoration

POST data preservation and restoration functions come into play when an application has a form wherein the user has entered a credential (or other data) but the session has expired, an idle session timeout has occurred, or the token validity period has ended by the time the user submits the form. If this scenario occurs, the user is presented with a fresh login form (depending on the authentication scheme) unless POST data is preserved and restored.

Administrators can configure the Resource Webgate to perform POST data preservation when the expired user and newly authenticated user are the same. [Table 22-33](#) describes Resource Webgate support and behavior for post data.

 **Note:**

Authentication POST data preservation and restoration is not supported when Access Manager performs authentication through custom agents.

Table 22-33 Resource Webgate Support of POST Data Preservation and Restoration

Resource Webgate	Description
Supports Authentication Schemes	LDAP, Basic, Sessionless Basic, X509, WNA
Supports form encoding	with text/html, text/plain, multipart/form-data, and application/x-www-form-urlencoded type data posted by the application form.
Preserves	The encoding type of the data posted by the original application form, except the input field of file type.
Ensures	The downstream application sees the same post data that was posted by the original application form.
Constrains	The overall size of the inbound request data or the inbound front channel message. There shall be a configuration parameter to override the code default value. This shall be per application.

Table 22-33 (Cont.) Resource Webgate Support of POST Data Preservation and Restoration

Resource Webgate	Description
Maintains application data confidentiality and integrity	Neither the Resource Webgate nor credential collector will interpret, nor log, application post data. If, after expiration and during re-authentication, the user authenticates with different credentials, then the post data of the previous user is cleared by the Resource Webgate and not restored. However, Webgate will post to the downstream application URL that was posted by the original application form.
Ignores Preservation if ... Logs a Message when ... Performs Standard Authentication if ... Shows an Error when ...	Post data is larger than the configured or hard-coded limit, preservation is ignored. Post data is skipped because it is bigger than the allowed limit, a message is logged. Post data size is larger than the hard-coded limit (or the configured value), the standard authentication flow is used. Together, if both front channel message data and application post data are large an error occurs.

Credential Collector Support for POST Data Handling

- ECC and DCC
- Compatible with earlier 11g Webgates
- Supports post data preservation for Form based authentication scheme with the default login form provided out of the box.
- Preserves application post data during authentication processing by:
 - Challenging the user
 - Re-challenging the user if invalid credentials are provided
 Does not interpret application post data.
- Constrains the overall size of inbound front-channel messages using a configuration parameter to override the default value, per application.
- Logs a warning when post data is skipped because it is larger than the allowed limit.
- Does not preserve application post data when:
 - Authentication policy is configured with Success or Failure authentication URLs
 - Password management (password expiration and so on) is involved
 - Access Manager is used for performing authentication through custom agents.
- ECC Only
 - The embedded credential collector does not support POST data handling with the external login page.
- DCC Only
 - POST data is preserved through the HTTP header, and the amount of POST data that can be handled to 8192 characters.
 - POST data restoration with a Form-based Authentication Scheme requires the challenge parameter **TempStateMode=form**.

- DCC does not support custom login pages.
- DCC does not support POST data restoration during password management operations (password expiration, for instance) when the URL_ACTION in the password policy plug-in is set to anything other than FORWARD.

22.12.2 Authentication POST Data Handling

Authentication Schemes such as WNA, Basic and X509 are supporting POST Data Handling.

The following are the Authentication Schemes Supporting POST Data Handling:

- FORM challenge method, supported with the out of the box login page.
- WNA
- Basic
- Basic+Sessionless
- X509
- OIF

[Table 22-34](#) summarizes complete configuration requirements for authentication POST data handling. All requirements described in [Table 22-34](#) are supported end to end with the specified authentication schemes.

Table 22-34 Parameters Required for Authentication POST Data Handling

Parameter	Description
MaxPostDataBytes	<p>Configure this Authentication Scheme challenge parameter for POST-data preservation used by the DCC only to limit the maximum size of the POST data that can be posted as on the login form. DCC compares the value of the content-length header with the limit set.</p> <p>Default: unlimited</p> <p>This Authentication Scheme challenge parameter requires a positive integer value that restricts the maximum number of bytes of POST data that is submitted as user credentials and sent to the OAM Server.</p>
MaxPreservedPostDataBytes	<p>Configure this Authentication Scheme challenge parameter (or user-defined Webgate parameter) for authentication POST-data preservation.</p> <p>Default: 8192 bytes</p> <p>Note: Preference is given first to the Authentication Scheme containing this parameter; second to the Webgate providing this user-defined parameter. Otherwise, default behavior is 8192 bytes.</p> <p>This parameter defines the maximum length of POST data that Webgate can preserve. If the size of inbound raw user POST data (or encrypted post data after processing), crosses this limit, POST data is dropped and the existing authentication flow continues. The event is logged as usual.</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>

Table 22-34 (Cont.) Parameters Required for Authentication POST Data Handling

Parameter	Description
TempStateMode=form <i>DCC Only</i>	<p>With the DCC, a Form-based Authentication Scheme requires the challenge parameter TempStateMode=form for POST data restoration. For Form authentication scheme, if this parameter is not defined, the value will be "form".</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>
ChallengeRedirectMaxMessageBytes	<p>Configure this user-defined Webgate parameter to limit the size of the message data received as obrareq.cgi and obrar.cgi. Message data is comprised of query string length (if present) or POST data length (if POST data is present). If message size exceeds this limit, the message is not processed and the existing message is shown in the browser. The event is logged as usual. Default: 8192 bytes</p> <p>Notes:</p> <p>obrareq.cgi is the authentication request in the form of a query string redirected from Webgate to the credential collector (OAM Server or DCC).</p> <p>obrar.cgi is the authentication response string redirected from the credential collector (OAM Server or DCC) to Webgate.</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>
PostDataRestoration	<p>Configure this user-defined Webgate parameter to initiate authentication POST-data preservation for the resource Webgate. This parameter requires a value of <code>true</code> or <code>false</code>. Default: <code>false</code></p> <p>When set to <code>true</code>, Webgate initiates POST data preservation.</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>
serverRequestCacheType <i>ECC Only</i>	<p>Configure this OAM parameter to define the mechanism used to remember the request context by the embedded credential collector (ECC).</p> <p>This OAM Server parameter in <code>\$DOMAIN_HOME/config/fmwconfig/oam-config.xml</code> indicates mechanism to be used to remember the request context. Possible values are FORM, COOKIE, or CACHE. Default: COOKIE</p> <p>FORM is the required value for POST data preservation, Long URL handling and Form-based authentication schemes.</p> <p>See Also: TempStateMode in this table. "Configuring Authentication POST Data Handling"</p>

22.12.3 Post Data Size Limits

Assuming the usual form data entered by users is about several kilobytes, putting a limit on data consumption from the incoming request is a general requirement. The data transferred

in the front channel protocol (either request or response) must also go through the size check.

Considering these situations:

- Limit the size of data passed to the OAM Server on the back channel using the `maxpostdatabytes` authentication challenge parameter
In cases where the DCC is used, the `maxpostdatabytes` authentication challenge parameter performs this check on the overall POST data.
- Limit the size of the POST data from the end user application using `MaxPreservedPostDataBytes` authentication scheme challenge parameter.
The `MaxPreservedPostDataBytes` authentication scheme challenge parameter handles this. Additionally, this can be set as a user-defined Webgate parameter.
- Limit size of the front channel payload on `obrar.cgi` or `obrareq.cgi` with a Webgate user-defined parameter `ChallengeRedirectMaxMessageBytes`.

22.12.4 Configuring Authentication POST Data Handling

Be sure to read all POST data topics in this section before attempting this procedure. There is no need to make any explicit change in your authentication scheme.

1. Configure the Authentication Scheme:
 - a. Use the Oracle Access Management Console to create or find the desired scheme ([Authentication POST Data Handling](#)).
 - b. On the Authentication Scheme page, modify values for POST data handling.
This example uses the embedded credential collector ([Table 22-22](#)) and values for POST data handling ([Table 22-25](#)):

```
Name: DesiredScheme
Authentication Level 2
Challenge Method: Form
Challenge Redirect URL: /oam/server/
Authentication Module: LDAP
Challenge URL: /pages/login.jsp
Context Type: External
Challenge Parameters
```

Authentication Scheme Challenge Parameters for Post Data with ECC	Authentication Scheme Challenge Parameters for Post Data with DCC
<code>MaxPreservedPostDataBytes=9000</code>	<code>MaxPreservedPostDataBytes=9000</code>
<code>0</code>	<code>TempStateMode=form</code>

- c. Click **Apply** to submit the changes.
2. ECC: Configure `serverRequestCacheType`, the OAM parameter in `oam-config.xml`, if using ECC.
 - a. Stop the managed server.
 - b. Stop the administration server.
 - c. Open `oam-config.xml` and modify the value of `serverRequestCacheType`.

See [Updating OAM Configuration](#).

- d. Restart the administration server.
 - e. Restart the managed server.
3. Configure Webgate Parameters for POST data handling:
 - a. From the System Configuration tab, Access Manager section, create or find the desired OAM Agent registration.
 - b. On the agent registration page, submit values for POST data handling ([Table 22-25](#)):

Name: *DesiredAgent*
User-Defined Parameters

User-Defined Webgate Post Data Parameters with ECC	User-Defined Webgate Post Data Parameters with DCC
PostDataRestoration=true	PostDataRestoration=true

- c. Click **Apply** to submit the changes.

22.12.5 Testing POST Data Handling Configuration

The following actions can be performed in sequence to test your POST data handling configuration.

1. Complete all configurations as documented.
2. Develop a simple script to print the POST data and the URL protected by Webgate.
3. Use a browser to access the protected resource.
4. Provide credentials and establish SSO. Wait for the idle session timeout period.
5. With the same browser, use the form to post data to the same Webgate using the URL which can print the POST data. You will be redirected to credential collector.
6. Enter the same credentials previously used.

From the HTTP headers you can see, after getting obrar.cgi from the credential collector, the protected resource Webgate will give a 200 response (previously it was 302) and the POST data can be printed by your script.

22.13 Long URL Handling During Authentication

Long URL handling applies to both credential collectors (ECC or DCC) and is a default operation.

22.13.1 About Long URLs and Authentication Handling

Authentication involves redirecting the user's request to a centralized component that performs authentication, known as a Credential Collector. The mechanism used to redirect user from the policy enforcement point (OAM Agent) to the Credential Collector, is a proprietary front channel protocol over HTTP. This protocol currently provides the context of the request and the authentication response on the query string. In situations where the URL of the requested page is larger, the overall context becomes larger and can go beyond the browser's permissible size. This is referred to as Long URL Handling.

By default, the Resource Webgate checks the payload size of the front channel protocol message to determine if it is larger than the coded limit. When long URL handling is explicitly enabled, the limit is ignored and has no impact.

The credential collector determines if the front channel response payload is to be sent as HTTP Post data when:

- The incoming request indicates that the agent is capable of handling HTTP POST or REDIRECT type of response
- The credential collector is configured to always send the payload as HTTP post data
- The credential collector is configured to always send the payload as a query string

If no explicit configuration is present, then if the payload size is greater than predefined limit, then it shall send payload as the HTTP post data. But if the payload size is lower than the predefined limit, then it shall send it on the query string.



Note:

If application post data is also preserved there is no impact.

Table 22-35 identifies Long URL handling functionality with both the ECC and DCC.

Table 22-35 ECC and DCC: Long URL Handling

ECC Long URL Handling	DCC Long URL Handling
ECC is compatible with all OAM Webgates.	Same as ECC.
N/A	<p>Long URL handling is limited to the maximum allowed size of the DCCContextCookie.</p> <p>The DCC does not perform explicit long URL handling.</p> <p>There is no support to preserve the front channel payload on the form.</p>

22.13.2 Configuration Requirements for Long URL Handling

The following are the Authentication Schemes Supporting Long URL Handling:

- FORM challenge method, supported with the out of the box login page.
- WNA
- Basic
- Basic+Sessionless
- X509
- OIF

Table 22-36 summarizes the parameters and complete configuration requirements for authentication Long URL handling. All requirements described in Table 22-36 are supported end to end with the specified authentication schemes.

Table 22-36 Parameters Required for Long URL Handling

Parameter	Description
ChallengeRedirectMethod	<p>Configure this as either as an Authentication Scheme challenge parameter (or as a user-defined Webgate parameter) for POST-data preservation for both the embedded credential collector (ECC) and the detached credential collector (DCC).</p> <p>Note: Preference is given first to the Authentication Scheme containing this parameter; second to the Webgate providing this user-defined parameter. Otherwise, default behavior is Dynamic.</p> <p>Value: GET POST DYNAMIC</p> <p>Behavior when value is:</p> <ul style="list-style-type: none"> • POST: Webgate sends encquery as POST data and credential collectors send encreply as POST data. • GET: Webgate sends encquery as query string and expects encreply as query string. • DYNAMIC: Default behavior, based on the length of the encquery/encreply. Webgate/credential collector sends data either as a query string or as POST data. Code default maximum length is 2000 characters. <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>
ChallengeRedirectMaxMessage Bytes	<p>Configure this user-defined Webgate parameter to limit the size of the message data received as obrareq.cgi and obrar.cgi. Message data is comprised of query string length (if present) or POST data length (if POST data is present). If message size exceeds this limit, the message is not processed and the existing message is shown in the browser. The event is logged as usual.</p> <p>Default: 8192 bytes</p> <p>Notes:</p> <p>obrareq.cgi is the authentication request in the form of a query string redirected from Webgate to the credential collector (OAM server or DCC).</p> <p>obrar.cgi is the authentication response string redirected from the credential collector (OAM server or DCC) to Webgate.</p> <p>See Also: "Configuring Authentication POST Data Handling" Table 15-2</p>
serverRequestCacheType <i>ECC Only</i>	<p>Configure this OAM parameter to define the mechanism used to remember the request context by the embedded credential collector (ECC).</p> <p>This OAM Server parameter indicates mechanism to be used to remember the request context. Possible values are FORM, COOKIE, or CACHE.</p> <p>Default: COOKIE</p> <p>FORM is the required value for POST data preservation, Long URL handling and Form-based authentication schemes.</p> <p>See Also: TempStateMode in this table. "Configuring Authentication POST Data Handling"</p>

Long URL handling is enabled by default. The Webgate/credential collector sends data either as a query string or a POST. The length of the querystring parameter sent with obrareq.cgi and obrar.cgi is 2000 characters maximum.

22.14 Using Application Initiated Authentication

Access Manager exposes a Reauthentication URL that applications may choose to invoke if the user is accessing a sensitive URL or operation. This re-authentication will be triggered irrespective of whether or not the user already has a valid session.

An application can trigger re-authentication by invoking the `/oamreauthenticate` URL at:

```
http://<ohs_host>:<ohs_port>/oamreauthenticate
```

Access Manager will expect the `/oamreauthenticate` to be registered and associated with an authentication policy. Re-authentication will be performed using the scheme associated with this policy. The re-authentication URL takes the redirection URL as a query parameter. After re-authentication is complete, Access Manager redirects the user to this URL. A request to re-authenticate the user might look like the following:

```
http://<host>:<port>/oamreauthenticate?  
  redirect_url=http://<host>:<port>/<redirection_resource_url>
```

If the redirection URL is not specified, a 404 error code is returned. If the incorrect credentials are specified during re-authentication, the user will remain on the login page and, after the maximum retry limit, the user will be redirected to an appropriate error page. The following process is how to configure for application initiated authentication.

1. Create an `http://<ohs_host>:<ohs_port>/oamreauthenticate` resource and assign the desired authentication scheme to it.
2. In the redirect URL, set the appropriate responses to verify that re-authentication has been successful and to communicate back to the application about the re-authentication responses.

Access Manager sets the last re-authentication time as a "OAM_LAST_REAUTHENTICATION_TIME" header and this value is updated every time the user is re-authenticated.

23

Understanding Credential Collection and Login

Access Manager credential collection is a functionality that can be executed either on the Access Manager server (ECC) or WebGates (DCC).

This chapter includes the following topics:

- [Overview of Access Manager Credential Collection](#)
- [Overview of the SSO Login Process with OAM Agents and ECC](#)
- [Overview of the SSO Login Process with OAM Agents and DCC](#)
- [Configuring OAM WebGate and Authentication Policy for DCC](#)
- [Tunneling from DCC to Access Manager Over Oracle Access Protocol](#)
- [Configuring a DCC WebGate for X509 Authentication](#)



Note:

Unless explicitly stated, information in this chapter is the same for all agent types and Access Manager credential collectors.

See [Introduction to Centralized Logout for Access Manager](#).

23.1 Overview of Access Manager Credential Collection

Access Manager provides two mechanisms for credential collection during authentication processing.

- The default Embedded Credential Collector (ECC) is installed with the Access Manager Server and can be used as-is with no additional installation or set up steps (except the global password policy configuration described in "[Managing Global Password Policy](#)").

The mechanism that redirects the user from the Policy Enforcement Point to the Credential Collector is a proprietary front channel protocol over HTTP. This protocol currently provides a context of the request and the authentication response on the query string.

- The OAM WebGate provides a single switch for the optional Detached Credential Collector (DCC). The DCC allows termination of end-user requests in the Web Tier as opposed to the Application Tier.

 **Note:**

- It is recommended that you use ECC for the new features introduced in OAM 12c. Some of the new features introduced in OAM 12c do not support DCC. For example, OpenIDConnect with DCC is not supported.
- ECC and DCC both allow for a secure deployment where a Defense in Depth approach requires that security vulnerabilities (DDOS, Throttling, Application Firewalling) be handled in the upstream application delivery tier. For typical Enterprise deployments, robust upstream capabilities exist, making any potential benefit of terminating unauthenticated connections on the Web Tier very minimal, given the limitations with DCC.

For a detailed comparison of the two mechanisms for credential collection, see [Understanding Authentication Methods and Credential Collectors](#).

Single Sign On login processing determines whether the user is a valid user and whether the session state is active or inactive (either a first time user or the user session has expired). Session management support locates, persists, and cleans up the session context and user token. Details are in the following sections.

- [Overview of the Login process with Self-Service Provisioning Applications](#)
- [Overview of the Login Process with Access Manager-Protected Resources](#)

23.1.1 Overview of the Login process with Self-Service Provisioning Applications

Provisioning does not create the session in Access Manager. When a new user uses a self-service provisioning application to create an account, he is prompted for his userID and password again when accessing an application.

The protected application is directed to Access Manager, which requests the user's credentials. For example, if Oracle Identity Governance is protected by Access Manager, the user request is redirected to Access Manager from which a request to enter credentials is made.

 **Note:**

Success and failure results are the same as described in "[Overview of the Login Process with Access Manager-Protected Resources](#)".

23.1.2 Overview of the Login Process with Access Manager-Protected Resources

The first time a user attempts to access a protected resource, she is prompted for her credentials based on the authentication scheme and authentication level for the resource. Typically a userID and password are needed.

Failure: Authentication fails if the wrong userID or password is entered. The user is not authenticated and another prompt for credentials appears.

With Oracle Access Manager, only the ECC in the OAM server was available. Access Manager supports the ECC by default. However, Access Manager also enables you to configure an OAM WebGate to use as a detached credential collector (DCC). A DCC-enabled WebGate can be separate from (or combined with) a Resource WebGates.

Both the ECC and DCC provide an authentication flow that includes form login, error, and login retries. They provide SecurID and server affinity as well as password policy enforcement and a dynamic, multi-step, iterative, and variable (multi-step authentication) where the credentials are not supplied all at one time. A customizable authentication flow can include authentication plug-ins with contracts between the plug-in, OAM Proxy, and Credential Collector; a contract between the plug-in and login application; and between the Credential Collector and login application.

When deciding whether to use one credential collector or both, consider:

- **Co-existence:** Allowing both the ECC and DCC to co-exist enables you to use authentication schemes and policies configured for either the ECC or the DCC. This enables a fallback mechanism for resources that rely on the ECC (Oracle Access Management Console, for instance).
- **Disabling ECC:** Disabling the ECC entirely prohibits access to resources that rely on the ECC mechanism (Oracle Access Management Console, for instance).

Table 23-1 provides links to more information.

Table 23-1 Login Processing with Access Manager-Protected Resources

Login Processing Topic	See
With OAM Agents and ECC	"Overview of the SSO Login Process with OAM Agents and ECC "
With OAM Agents and DCC	"Overview of the SSO Login Process with OAM Agents and DCC "
With Other Agents or Mixed Agent Types	Mixed agent types are supported. Processing is the same for each agent type.
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 Access Manager SSO, and uses OPSS SSO for user authentication. For more information, see Integrating Oracle ADF Applications with Access Manager SSO .

23.2 Overview of the SSO Login Process with OAM Agents and ECC

Access Manager authenticates each user with a customer-specified authentication method to determine the identity and leverages information stored in the user identity store.

This topic is based on using the default Embedded Credential Collector with OAM Agents (Resource WebGates) protecting resources.

Access Manager authentication supports several authentication methods and a number of 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 Access Manager which checks for the existence of the SSO cookie.

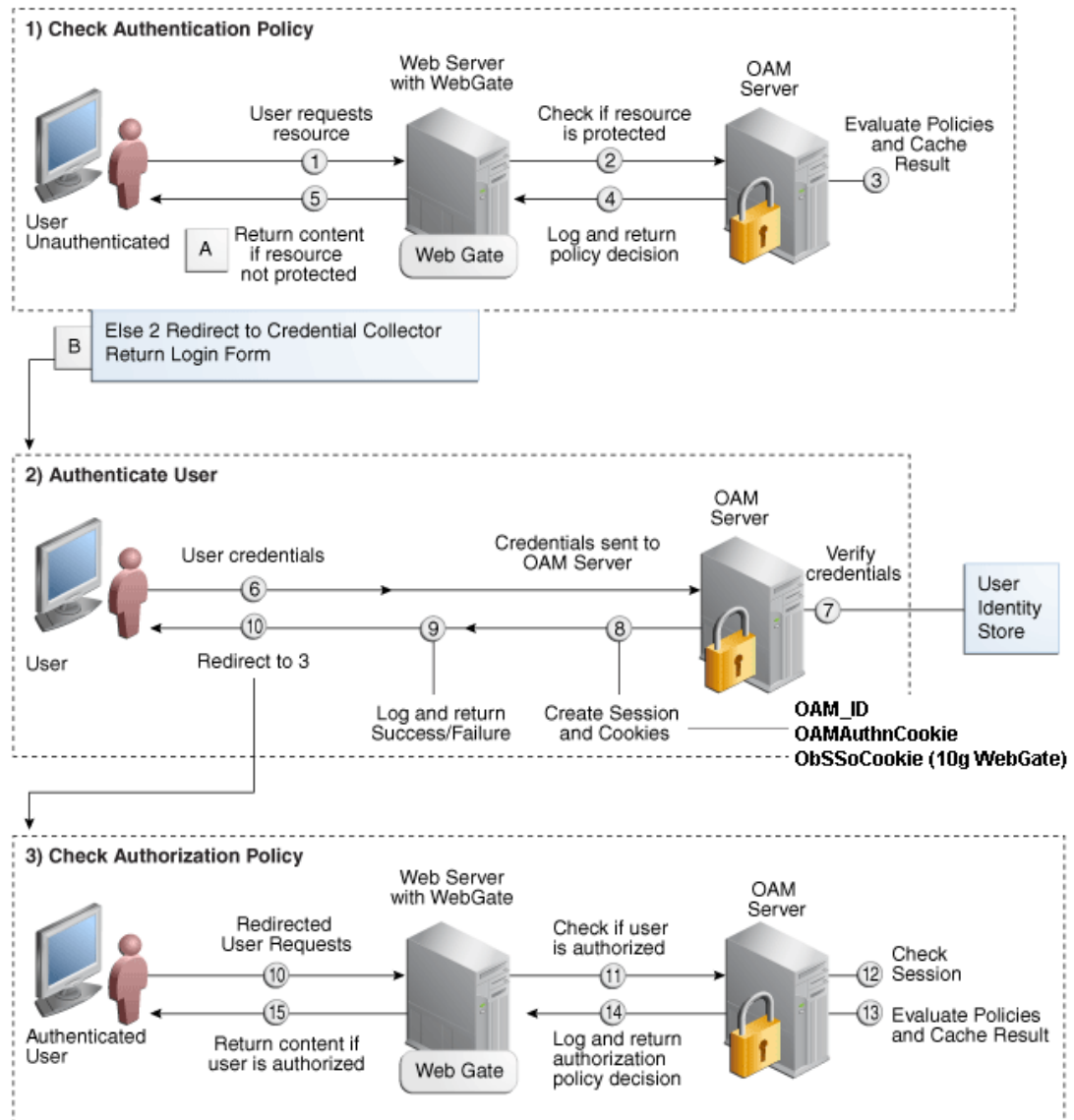
After authenticating the user and setting up the user context and token, Access Manager sets the SSO cookie and encrypts the cookie with the SSO Server key (which can be decrypted only by the SSO Engine).

Depending on the actions (responses in Access Manager 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 23-1](#) 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. There are slight variations with 12c WebGates/Access Clients.

Figure 23-1 SSO Log-in with Embedded Credential Collector and OAM Agents



Process overview: SSO Login Processing with Embedded Credential Collector and OAM Agents

1. The user requests a resource.
2. WebGate forwards the request to Access Manager for policy evaluation.
3. Access Manager:
 - Checks for the existence of an SSO cookie.
 - Checks policies to determine if the resource protected and if so, how?
4. Access Manager 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. Access Manager verifies credentials.
- 8. Access Manager starts the session and creates the following host-based cookies:
 - **One per Agent:** OAMAuthnCookie set by OAM WebGates 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. Access Manager logs Success or Failure.
- 10. Credential collector redirects to WebGate and authorization processing begins.
- 11. Webgate prompts Access Manager to look up policies, compare the user's identity, and determine the user's level of authorization.
- 12. Access Manager 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.

23.3 Overview of the SSO Login Process with OAM Agents and DCC

The detached credential collector is simply a WebGate configured to use the additional Credential Collection capability in your deployment.

There are two deployment types depending on whether the DCC WebGate is also protecting the applications or not.

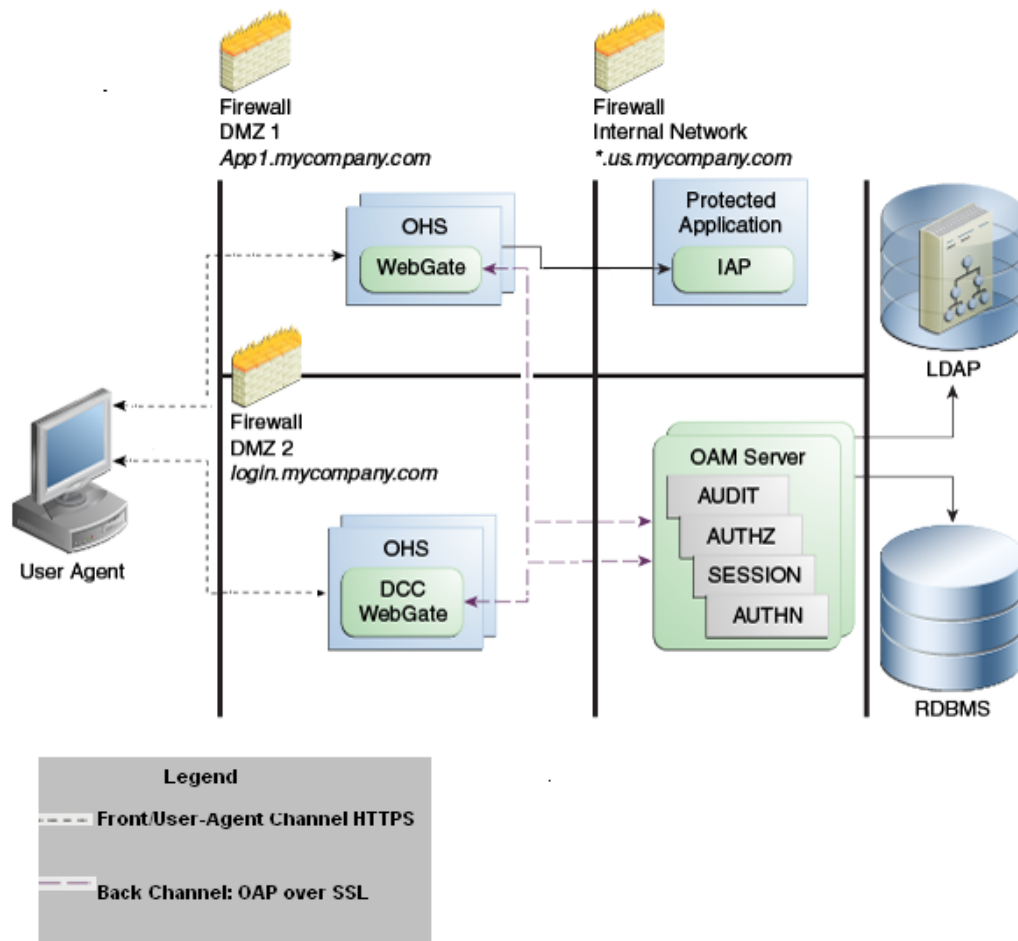
[Table 23-2](#) identifies the DCC-supported deployments.

Table 23-2 DCC Deployment Support

Deployment Type	Description
Separate DCC and Resource Webgate	<p>A distributed deployment where WebGates protecting applications are managed independently from the centralized DCC. You can have:</p> <ul style="list-style-type: none"> • Two or more Resource Webgates that redirect to the DCC-enabled WebGate for authentication • <p>Enable HTTPS between the user-agent and the DCC (but not with some or all Resource WebGates).</p> <p>When credential collection is externalized and centralized in the DCC, the user-agent connections with other WebGates never carry user credentials, nor session tokens that could be used to obtain access to resources protected by any other WebGate. This significantly reduces exposure caused by lack of SSL on these links and may be an acceptable tradeoff in some deployments.</p> <ul style="list-style-type: none"> • Separate OHS Instances: Install the DCC on a different OHS instance (on the same or different host) as the Resource WebGate. • Define the Resource WebGate Authentication Scheme Challenge Redirect URL to point to the DCC. • Define the Resource WebGate logoutRedirectUrl to point to the DCC logout script/page (logout callbacks to Resource WebGate is invoked during logout). <p>See Also: Figure 23-2</p>
Combined DCC and Resource Webgate	<p>A streamlined deployment minimizing configuration and processing overhead.</p> <p>A DCC WebGate can function as both a resource WebGate (Policy Enforcement Point) that protects application resources and a DCC. In this case, there is no front-channel redirection or processing:</p> <ul style="list-style-type: none"> • Install the DCC on a the same OHS instance (on the same host) as the Resource WebGate. • Simplified configuration: The Challenge Redirect URL can be empty. • No logoutRedirectUrl is needed, no logout callback is needed. <p>See Also: Figure 23-3</p>

Separate DCC and Resource Webgates: A sample deployment with segregated DCC is shown in [Figure 23-2](#).

Figure 23-2 Example: Separate Resource WebGate and DCC WebGate Deployment



This topology (Figure 23-2) showcases choices appropriate for scenarios with maximum security sensitivity. Both centralized and externalized credential collection are used: Resource WebGates protecting applications are segregated from the DCC WebGate performing credential collection.

The user accesses the Access Manager-protected resource from the public network. A WebGate protecting the application is deployed within a DMZ. The DCC WebGate is also deployed within a DMZ. The protected application and OAM Server instances are located within the private network and not directly accessible from the public network.

Using the DCC in the DMZ, only authenticated network connections are allowed to reach the server itself. The DCC inherits all back-channel communication characteristics available to OAM WebGates (network connection using the Oracle Access Protocol). The OAP offers:

- SSL between the client and the server, optionally using 3rd party signed certificates
- mutual authentication at the application level using client id and password
- request multiplexing and full-duplex communication at the application level

- built-in connection load balancing and failover capability

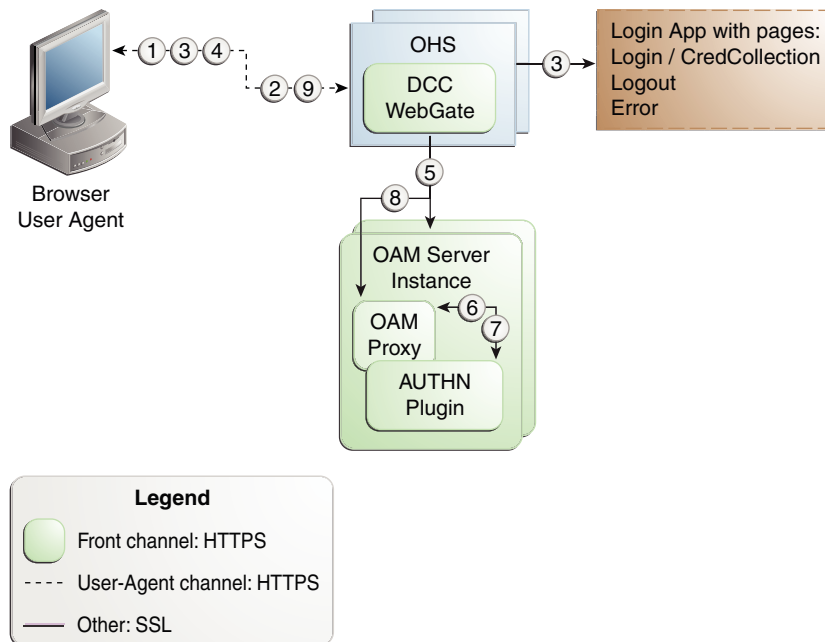
The DCC receives an authentication request from the Agent and checks for the presence of the DCC cookie. If the cookie does not exist, credential collection is initiated; checks are made, and user-supplied credentials are passed for validation.



Note:

Encryption occurs only from resource WebGate to the DCC. The channel is not encrypted for communication between resource WebGate and DCC; this is in clear text.

Figure 23-3 Combined DCC and WebGate Configuration



Process overview: Authentication with the combined DCC and Resource Webgate

1. The user requests access to a resource which initiates the authentication process.
2. The DCC redirects through the front channel to the login page.
3. The login page is returned to the user.
4. User enters credentials, which are posted to the action URL (a user-defined parameter in an authentication scheme, [Table 22-25](#)).
5. Authentication occurs using the back channel (OAP) and OAM Proxy.
6. The Authentication Plug-in is activated.
7. The Plug-in requests redirect to a URL to collect additional credentials.
8. The Plug-in request is returned to the DCC.
9. The DCC redirects to the URL and expects specified credentials.

10. The Browser follows the redirect.
11. Credentials are posted to the Action URL.



See Also:

["Configuring Logout When Using Detached Credential Collector-Enabled WebGate"](#)

23.4 Configuring OAM WebGate and Authentication Policy for DCC

Administrators can enable DCC credential operations, update DCC forms for password policy, add PasswordPolicyValidationScheme to Authentication Policy, and use DCC for converged Federation flows.

The following steps describe configuring a WebGate and Authentication Policy for use with the DCC. The appropriate sub sections are linked within each step.

1. [Enabling DCC Credential Operations](#) provides steps for either configuration:
 - DCC Combined with Resource Webgate:** Enable Allow Credential Collector Operations in the DCC's OAM Agent registration page.
 - Separate DCC and Resource Webgate:** Enable Allow Credential Collector Operations in the DCC's OAM Agent registration page and edit the Resource Webgate registration page to set the `Logout Redirect URL` to the DCC's `logout.pl`.
2. [Locating and Updating DCC Forms for Password Policy](#)
3. [Adding PasswordPolicyValidationScheme to Authentication Policy for DCC](#) provides steps for either configuration:
 - DCC Combined with Resource Webgate:** In the combined DCC/Resource Webgate Application Domain, update the Protected Resources Authentication Policy to use your DCC Authentication Scheme.
 - Separate DCC and Resource Webgate:** In the separate Resource Webgate Application Domain, update the Protected Resources Authentication Policy to use your DCC Authentication Scheme.
4. [Supporting Federation Flows With DCC](#) provides steps to incorporate the DCC into Federation flows.



Note:

If your environment uses the ECC, See [Completing Password Policy Configuration](#).

23.4.1 Enabling DCC Credential Operations

Whether you are using a separate DCC or combined DCC and Resource WebGate, you must enable Allow Credential Collector Operations in the DCC's OAM Agent registration page.

With a separate DCC and Resource WebGate, you must also edit the Resource WebGate registration page to set the `Logout Redirect URL` to the DCC's `logout.pl`, as described in Step 3.

The following procedure presumes your deployment uses Open mode communication. If your deployment uses Simple or Cert mode communication, be sure to copy the appropriate artifacts when you perform Step 4.

Prerequisites

- [Configuring and Managing Registered OAM Agents Using the Console](#)
 - [Managing Global Password Policy](#)
 - [Configuring Password Policy Authentication](#) using DCC-specific details
1. In the Access Manager section of the Oracle Access Management Console, click **SSO Agents** to find and open the registration page for the OAM Webgate that will function as the DCC.
 2. **DCC WebGate Registration:** Check **Allow Credential Collector Operations**, change **Token Validity Period** to match the globally configured Session Lifetime value. Click **Apply** and then perform Steps 4 and 5.

 **Note:**

If the DCC is combined with a Resource WebGate, skip Step 3.

3. **Separate Resource WebGate:** Edit the Resource WebGate registration to set the `Logout Redirect URL` to the DCC's `logout.pl` ([Table 24-3](#)), click **Apply**, then perform Steps 4 and 5.
4. Copy Agent configuration file (including Simple or Cert mode files) from the AdminServer (Console) host to the Agent host. For example:

Agent & Artifacts	Artifacts
OAM WebGate/Access Client	From the AdminServer (Console) host:
ObAccessClient.xml and cwallet.sso	<code>\$DOMAIN_HOME/output/\$Agent_Name/</code> To the Agent host: <code>\$12cWG_install_dir/webgate/config</code>
Simple or Cert Mode	Copy to the Agent host: <code>\$12cWG_install_dir/webgate/config</code> <ul style="list-style-type: none"> • <code>aaa_key.pem</code> • <code>aaa_cert.pem</code> • <code>aaa_chain.pem</code> • <code>password.xml</code> See Also: Securing Communication

5. Restart the OHS Web server.
6. Proceed to "[Locating and Updating DCC Forms for Password Policy](#)".

23.4.2 Locating and Updating DCC Forms for Password Policy

Access Manager provides several dynamic pages for user interactions with the DCC.



See Also:

Developing Applications with Oracle Access Management

Prerequisites

Enabling DCC Credential Operations

1. Locate the DCC forms in the WebGate host (Table 24-3): `$WEBGATE_HOME/webgate/ohs/oamssso/*`, `$WEBGATE_HOME/webgate/ohs/oamssso-bin/*pl`, and `$WEBGATE_HOME/webgate/ohs/oamssso-bin/templates/*`.
2. Customize their location, depending on the desired topology of the authentication scheme being developed.
3. **Update Perl Location:** Update the Perl location to be consistent with the actual location, in the first line of the login, logout, and securid scripts on Webgate host in `$WEBGATE_HOME/webgate/ohs/oamssso-bin/*pl` (Table 24-3).
4. Customize the default pages for your enterprise, or replace them entirely with custom pages. For example, you can design, implement, and deploy a custom page that displays a different version of the login form for a mobile browser than is used for a desktop browser.
5. Proceed to "Adding PasswordPolicyValidationScheme to Authentication Policy for DCC".

23.4.3 Adding PasswordPolicyValidationScheme to Authentication Policy for DCC

You can use your DCC Authentication Scheme in a Protected Resources Authentication Policy.

The steps you perform depend on the type of deployment you have:

- **Combined DCC/Resource WebGate:** Perform Step 1 to add your DCC Authentication Scheme to the Protected Resources Authentication Policy of the combined DCC/Resource WebGate Application Domain.
- **Separate Resource WebGate:** Perform Step 3 to add your DCC Authentication Scheme to the Protected Resources Authentication Policy of the separate Resource WebGate Application Domain.

Perform Step 2 regardless of your DCC deployment type. By default, login and logout forms are excluded through OHS /httpd.conf/webgate.conf so that you do not need to exclude them through policies. However, with the Chrome browser, you must explicitly exclude the async favicon.ico request (which overrides the DCCctxCookie).



Note:

This example refers to the PasswordPolicyValidationScheme set for the DCC in [Configuring Password Policy Authentication](#).

Prerequisites

[Locating and Updating DCC Forms for Password Policy](#)

1. **Combined DCC/Resource WebGate:** Open the DCC application domain:

Policy Configuration
Application Domains
DCCDomain

- a. Locate and open the **Authentication Policy, Protected Resource Policy** (see ["Searching for an Authentication Policy"](#)).
- b. Add your **DCC Authentication Scheme** to this policy (see ["Defining Authentication Policies for Specific Resources"](#)).

PasswordPolicyValidationScheme (DCC Authentication Scheme)

- c. Perform Step 2 if you have the Chrome Browser. Otherwise, go to Step 4.

2. **Chrome Browser:** Add and exclude resource */favicon.ico* in the *DCCDomain*, as follows.

- a. From *DCCDomain*, click the **Resources** tab.
- b. Find and open the HTTP resource */favicon.ico* (or click the New Resource button and then add this resource).
- c. Confirm or edit the Resource URL to:
/favicon.ico
- d. In the **Protection** section, **Protection Level** list, select **Excluded**, then click **Apply**.
- e. Proceed to Step 4.

3. **Separate Resource Webgate:** Open the Resource Webgate application domain.

Policy Configuration
Application Domains
ResourceWGDomain

- a. Locate and open the **Authentication Policy, Protected Resource Policy** (see ["Searching for an Authentication Policy"](#)).
- b. Add your **DCC Authentication Scheme** and an optional Failure URL (when not specified, Failure URL displays the default error page) to this policy (see ["Defining Authentication Policies for Specific Resources"](#)):

DCC Authentication Scheme
Failure URL (optional)

- c. Perform Step 2 if you have the Chrome Browser. Otherwise, go to Step 4.
4. Restart your Web server and proceed to "[Completing Password Policy Configuration](#)".

 **See Also:**

["Configuring Logout When Using Detached Credential Collector-Enabled WebGate"](#)

23.4.4 Supporting Federation Flows With DCC

The DCC is enhanced to work as a public end-point to the Access Manager server. HTTP requests to the DCC are tunneled through NAP to the proxy module of the Access Manager server. Only requests defined in the TunneledUrls parameter of the DCC Profile will be tunneled. The JSP pages and servlets are executed in the Access Manager server and the response is tunneled back to the DCC. The end user effectively communicates only to the DCC.

 **Note:**

If a WebGate is configured as a DCC and federated flows are in use, the DCC WebGate cannot be used to protect the resource. A separate WebGate must be configured and used to protect the resource. Authentication and authorization requests will be tunneled to the OAM Server, and the ECC login form will be tunneled and displayed in the user's browser.

To use DCC for converged Federation flows, perform the following manual steps.

1. Configure the following internal resources as Public instead of Excluded.

```
/oamfed/.../*  
/oam/.../*  
/.../*
```
2. In the DCC WebGate, set the logout value to a valid DCC WebGate logout URL; for example, `/oamsso-bin/logout.pl`

3. Update the DCC Agent entry by adding the following entry to the User Defined Parameters list using the Access Manager Administration Console.

```
TunneledUrls=/oam,/oamfed
```

See [Configuring OAP Tunneling](#).

4. Update the OAM public endpoint entry so that it points to the DCC WebGate.
Under Access Manager Settings, set the OAM Server Host, OAM Server Port and OAM Server Protocol to the values pertinent to the OHS/DCC and click Apply.

 **Note:**

Alternately you can update a single Authentication Scheme to point to the DCC WebGate by altering the challenge redirect URL leaving the REST parameters unchanged.

5. Update the ProviderID value under Federation Settings (if applicable) and redistribute the new metadata to all Federation partners due to the endpoint change.
6. Set the contextType to 'External'.
See [Authentication Schemes and Pages](#) for details on this setting.

23.5 Tunneling from DCC to Access Manager Over Oracle Access Protocol

Access Manager supports HTTP communication over the Oracle Access Protocol (OAP). In this case, a WebGate configured as a DCC uses the ECC servlets for credential collection during Access Manager authentication.

The following sections contain more details.

- [How DCC Tunneling with OAP Works](#)
- [Configuring OAP Tunneling](#)

 **Note:**

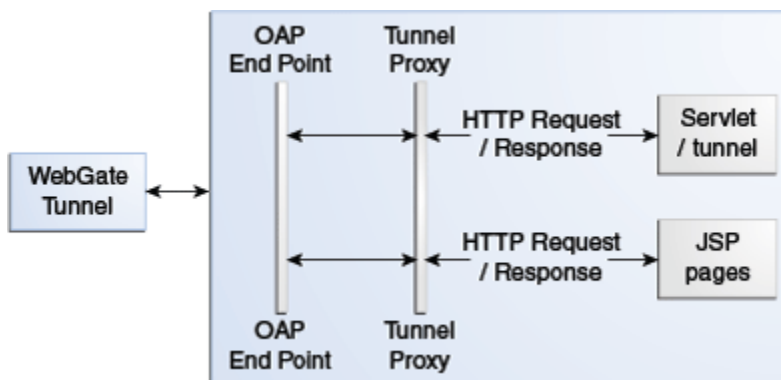
For details on the DCC, see [Embedded Credential Collector Versus Detached Credential Collector](#).

23.5.1 How DCC Tunneling with OAP Works

The tunneling process works for DCC WebGates only.

[Figure 23-4](#) illustrates how the tunneling process works.

Figure 23-4 OAP Tunneling with DCC



The following steps provide more details in regards to the OAP Tunneling process.

1. The URL to be tunneled is configured in the DCC WebGate profile.
2. This same URL is mapped to a servlet or JSP page in the Access Manager server.
3. On accessing the tunneled URL, the WebGate intercepts the HTTP request and converts it to an OAP request.
4. The OAP request is forwarded to the Access Manager server.
5. The Access Manager server (OAM proxy) receives the OAP request and passes it to the tunnel proxy.
6. The tunnel proxy will convert the OAP request to an `HttpServletRequest` and invoke the corresponding servlet (or compiled servlet in the case of a JSP).
7. The response is converted back to an OAP message and passed to the OAP end point.
8. The OAM end point responds to the WebGate with the converted OAP message.
9. The WebGate converts the OAP message back to an HTTP response.
10. The WebGate provides the HTTP response to the caller (browser).

23.5.2 Configuring OAP Tunneling

To configure OAP Tunneling, a WebGate must be installed and configured to work with the Access Manager server as a DCC.

The Access Manager endpoint must be deployed on the Access Manager Server. After ensuring these prerequisites have been met, add a user-defined parameter to the WebGate profile that defines all URLs to be tunneled using the form `TunneledUrls=<URL>,<URL1>`. For example:

```
TunneledUrls=/oam,/sampleapp
```

Lastly, protect the Tunneled URLs with an Authentication Policy. For details, see [Managing Authentication and Shared Policy Components](#).

23.6 Configuring a DCC WebGate for X509 Authentication

Configure a WebGate for DCC and convert it to SSL for using it with X509 authentication.

1. [Configuring the WebLogic Server](#)
2. [Configuring a WebGate For DCC](#)
3. [Converting the DCC WebGate to SSL](#)

23.6.1 Configuring the WebLogic Server

Use the procedures in the following sections to configure a WebLogic Server for X509 authentication.

1. [Creating the Server and Trust Store](#)
2. [Configuring the WebLogic Server Instance](#)
3. [Creating the User Certificate](#)

4. Adding the Root CA Certificate

23.6.1.1 Creating the Server and Trust Store

These are common procedures for WebLogic Server.

1. Create Server certificate

Create a Server Certificate and key for the WLS domain on which Oracle Access Management 11g is deployed. This entails requesting a certificate (in which the Common Name is the OAM server machine name), having the certificate signed and converting it to the P12 format. The Server certificate can be created and signed using any Certificate utility.

2. Create the server store and the trust store using keytool.

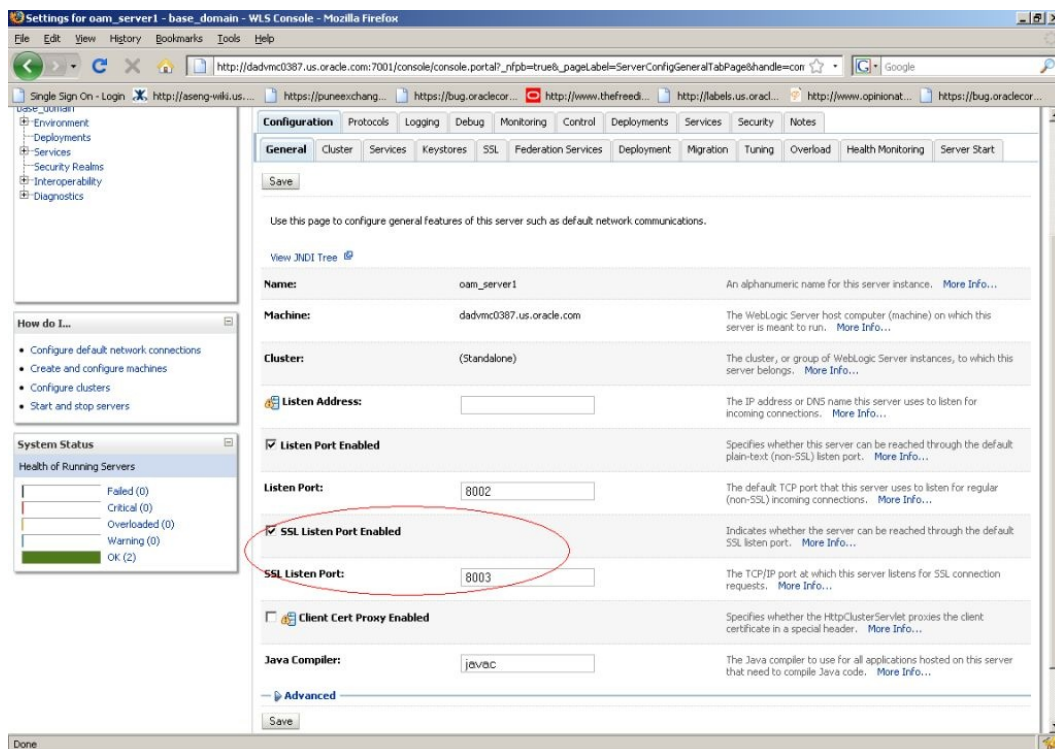
See [Securing Communication Between OAM Servers and WebGates](#) for details.

23.6.1.2 Configuring the WebLogic Server Instance

Use the WebLogic console to configure the instance of the WebLogic Server to be SSL and client certificate enabled.

1. Navigate to the server instance which is to be SSL and Client Cert enabled.

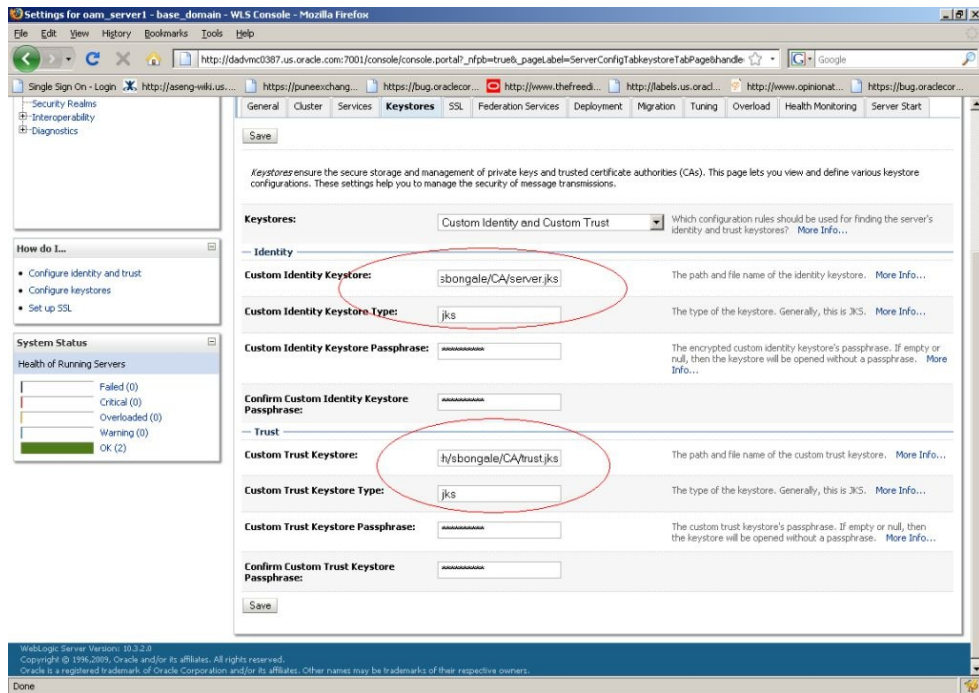
Figure 23-5 Enable SSL



2. Check the SSL Listen Port Enabled check box and provide the port number as in [Figure 23-5](#).

3. Provide the server and trust keystore path under the "Keystore" tab.

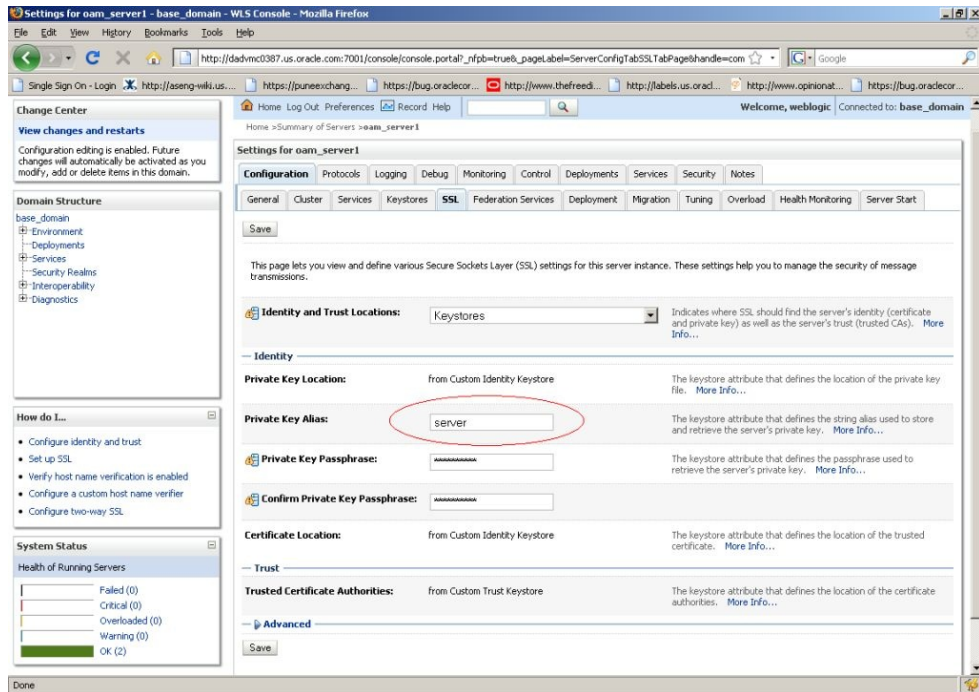
Figure 23-6 Keystore Configuration



4. Add the private key alias details under the SSL tab.

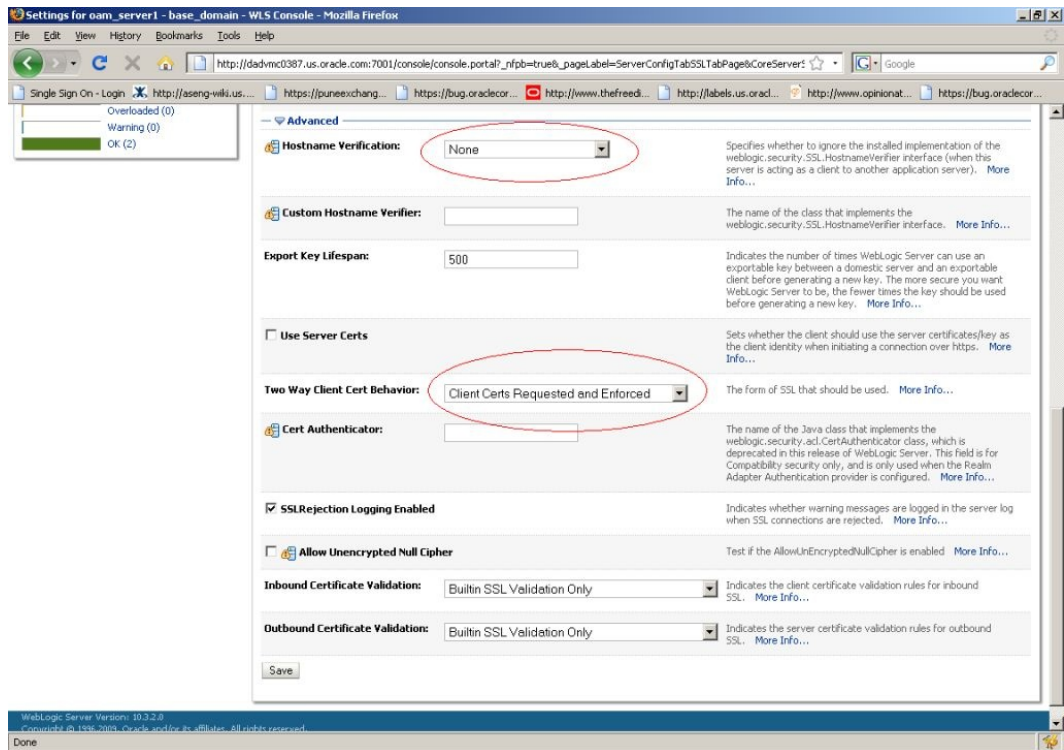
The alias name is same name specified as the server store name in [Creating the Server and Trust Store](#).

Figure 23-7 Add Private Key Alias



5. Display the Advanced options under the SSL tab and make the configurations illustrated in [Figure 23-8](#).

Figure 23-8 SSL Advanced Options



23.6.1.3 Creating the User Certificate

You can create a user certificate in the .p12 format and install it in your browser.

Run the following OpenSSL commands:

1. `openssl req -config openssl.cnf -new -out weblogic.csr`
Provide the certificate details. The Common Name is the name of the user for whom the certificate is requested.
2. `openssl x509 -req -md5 -CAcreateserial -in weblogic.csr -days 180 -CA F:\openssl\simpleCA\ca.pem -CAkey F:\openssl\simpleCA\ca-key.pem -extfile F:\openssl\openssl.cnf -out weblogic.pem`
3. `openssl rsa -in privkey.pem -out weblogic.key`
4. `openssl pkcs12 -export -in weblogic.pem -inkey weblogic.key -out user1k1.p12`
5. Install the .p12 formatted certificate output in your browser.

23.6.1.4 Adding the Root CA Certificate

You can add the Root CA certificate of the certificate utility used to SSL enable the WebLogic server.

(In this example, the OpenSSL certificate utility is used.) The Root CA certificate must be added to the `.oamkeystore` and `amtruststore` files located in the following WebLogic directory:

`$DOMAIN_HOME/base_domain/config/fmwconfig`

1. Retrieve the password for the `.oamkeystore` and `amtruststore` files in WebLogic.
 - a. Navigate to `$MIDDLEWARE_HOME/Oracle_IDM1/common/bin/`.
 - b. Run `wlst.sh`.
 - c. Run `connect()` in the WLST shell.
 - d. Run `domainRuntime()` in the WLST shell.
 - e. Run `listCred(map="OAM_STORE",key="jks")` in the WLST shell to display the password.
2. Add the Root CA certificate to the `.oamkeystore` and `amtruststore` files using the `keytool` command.

The value of `-storepass` is the password retrieved in the previous step. For example:

```
./keytool -importcert -alias ROOT_CA -file /scratch/CA/ca.pem -keystore /
scratch/Oracle/Middleware/user_projects/domains/base_domain/config/
fmwconfig/.oamkeystore -storepass oru8nd3hhd4t4nrmh6unhv825b -storetype jceks

./keytool -importcert -alias ROOT_CA -file /scratch/CA/ca.pem -keystore /
scratch/Oracle/Middleware/user_projects/domains/base_domain/config/fmwconfig/
amtruststore -storepass oru8nd3hhd4t4nrmh6unhv825b -storetype jks
```

23.6.2 Configuring a WebGate For DCC

You can configure a WebGate for DCC. As part of this procedure you will also create the `LDAPScheme_DCC` Authentication Scheme.

You will use the Oracle Access Management Console for the configuration steps. This procedure assumes you have already installed the WebGates for which you will be creating profiles.

1. Configure an OAM WebGate profile named, for example, `ABC_WG1` on `http://<host>:7778/index.html`.
2. Configure an OAM WebGate profile named, for example, `XYZ_WG1_DCC` on `http://<host>:7779/index.html`.

This WebGate will act as the authentication WebGate.

3. Navigate to the `XYZ_WG1_DCC` WebGate profile and Select Allow Credential Collector Operations Option.

This configures the WebGate for use as a DCC.

4. Create a new Authentication Scheme by making a copy of the `LDAPScheme` Authentication Scheme and modifying the following values.

Only modify the following values; leave the other parameters untouched.

- a. Name as `LDAPScheme_DCC`
 - b. Challenge redirect URL is `http://<host>:<port>/` (`http://<host>:7779/`)
 - c. Challenge URL : `/oamso-bin/login.pl`
5. Navigate to the `ABC_WG1` Application Domain and do the following.
 - a. Go to Authentication Policy.

- b. Point to the server.pem certificate generated in step 3.
 - c. Import the server certificate and save the wallet.
6. Edit the Oracle HTTP Server (OHS) ssl.conf file to point to this wallet as follows.

```
#Path to the wallet
SSLWallet "/<path to wallet>/wallet"
SSLVerifyClient require
```

ssl.conf is located at <webtier>/<instance_home>/config/ohs/ssl.conf.

- 7. Restart the OHS instance.

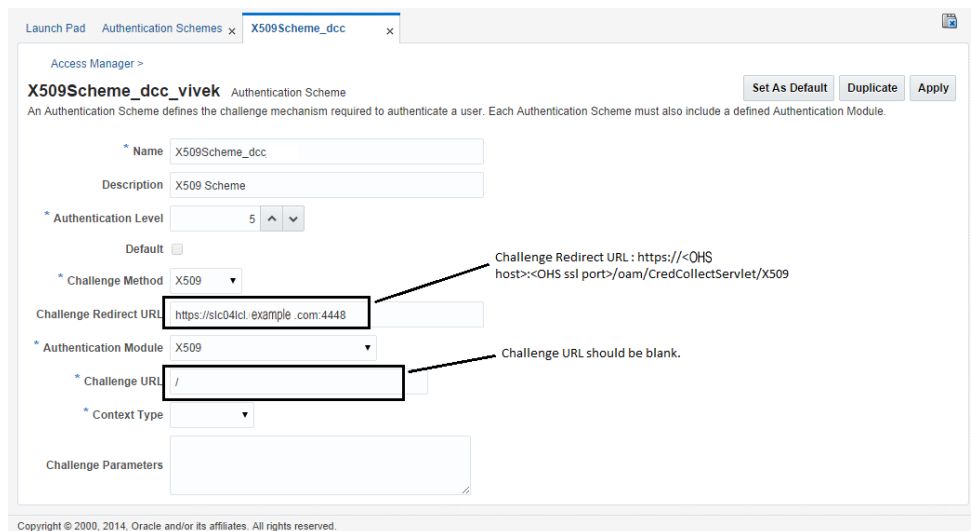
23.6.3.2 Generating and Importing Client Certificates

You can generate and import client certificates and create a new X509 authentication scheme.

- 1. Create a user certificate by following the steps documented in [Creating the User Certificate](#).
- 2. Create a new Authentication Scheme named X509_DCC as illustrated in [Figure 23-9](#).

Add a Challenge Redirect URL. The Challenge URL should be blank.

Figure 23-9 New X509 Scheme



- 3. Import <user_cert>.p12 into your browser.
- 4. Access the protected resource via its SSL port. For example:

```
https://<ohs_host>:<ohs_port>/index.html
```

A popup is displayed asking which certificate to use. Select the appropriate certificate and the requested resource is accessed.

24

Using Password Policy

Access Manager provides several pages for user interactions during credential collection. This includes login, error and password forms. This chapter contains details on these forms and how to configure a password policy.

- [Understanding Password Management](#)
- [Enabling Password Management](#)
- [Accessing Password Policy Configuration Page](#)
- [Specifying Credential Collector URLs with Password Policy](#)
- [Oracle-Provided Password Forms](#)
- [Managing Global Password Policy](#)
- [Configuring Password Policy Authentication](#)
- [Completing Password Policy Configuration](#)
- [Configuring the PasswordManagementPlugin](#)

24.1 Understanding Password Management

The Password Management feature is only supported when the identity store used is an LDAP directory.

When enabled, Password Management can be used for the following scenarios.

- When a user account is disabled by an administrator, the user is not allowed to enter the system. An appropriate error message is displayed if the user requests access.
- When a user account is locked by an administrator (whether permanently or temporarily due to incorrect passwords or challenges), the user is not allowed to enter the system. An appropriate error message is displayed if the user requests access.
- A user can be forced by the administrator to change a password if specific values are set in the user's LDAP entry.
- When a user has to change a soon-to-be expiring password, a screen is displayed from which the user can choose one of the following options: change the password now or continue to the requested page without changing the password.
- If a user submits a password with invalid characters during a password reset operation, an error message can be displayed with the password creation rules that the user must follow.

[Oracle-Provided Password Forms](#) has screenshots of the error messages discussed.

Caveats for Integrated Deployments

When you are using Oracle Identity Management and Oracle Access Management with Oracle Internet Directory, there are two sets of password policy definitions and enforcement. Password Policy Definition can be configured in both Oracle Identity Management and in Oracle Internet Directory. Password Policy Enforcement occurs according to the following:

- Oracle Access Management enforces state policies (incorrect password, for example) during Web access; Oracle Internet Directory enforces its own state policies as well as LDAP operations (bind and compare, for example).
- Oracle Identity Management enforces value policies (characteristics of the password) during user creation of the password update; Oracle Internet Directory enforces its own value policies as well for policies for LDAP operations (add, modify for example).

Password Policy is only certified when the configured Identity Store is an LDAP directory. It is not certified with a virtualized LDAP directory (for example, Oracle Virtual Directory fronting another data repository) or a non LDAP directory

Any LDAP directory (such as Oracle Internet Directory) has a way to configure password policies that define lexical constraints to which the user password must conform (minimum characters, maximum length of time the password is valid, use of special characters, etc.) This password policy gets applied when the user's password is changed in the LDAP directory. To make sure that this LDAP directory password policy does not conflict with the password policy configured in OAM, the administrator has to manually study the LDAP password policy and do one of the following.

1. Make the backend LDAP identity store policies weaker or the same strength as the Oracle Identity Management and Oracle Access Management policies. However, this leads to a double enforcement.
2. Disable native LDAP password policy validation, which unfortunately leaves no enforcement for direct LDAP operations.

24.2 Enabling Password Management

Use the Oracle Access Management Console to enable the Password Management service. This is done as a configuration of the defined user identity store.

The Password Management feature is only supported when the identity store used is an LDAP directory.

1. Log in to the Oracle Access Management Console as Administrator.
2. Click Configuration at the top right of the Oracle Access Management Console.
3. Click User Identity Stores in the Configuration console.
4. Select the appropriate LDAP directory to enable Password Management.

Alternately, click Create to register a user identity store. See [Managing Data Sources](#) for details.

5. Under Password Management, check Enable Password Management.
6. Define the Password Management parameters and click Apply to save.

[Table 24-1](#) documents the parameters used for configuration.

Table 24-1 Password Policy Configuration Parameters

Parameter	Description
Enable Password Management	Enables password management for this identity store. If password management is not enabled, the password plugin returns right away and the status is not captured.
Use Oblix Schema	If checked, the Oblix schema is used. If not, the Oracle Schema is used.
Global Common ID Attribute	This is the userid attribute used for password policy verification to make sure the password doesn't contain the user id attribute value.
First Name Attribute	This is the first name attribute used for password policy verification to make sure the password doesn't contain the first name attribute value.
Last Name Attribute	This is the last name attribute used for password policy verification to make sure the password doesn't contain the last name attribute value.
Email Address Attribute	This is the email attribute of the users in this identity store. It is used for password policy verification.

Note:

Ensure **Password Management** is enabled before you use the Multifactor authentication OTP REST APIs. If **Password Management** is not enabled, the null pointer exception occurs: Exception occurred while resetting password using OTP for user with exception `java.lang.NullPointerException`

Note: Password Management Module replaces the deprecated Password Policy Validation Module.

24.3 Accessing Password Policy Configuration Page

Once Password Management is enabled, you can configure the Password Policy. Administrators define password policy based on enterprise requirements. When configured, the Password Options and Challenge Options are used by both the Embedded Credential Collector (ECC) and Detached Credential Collector (DCC).

See [Understanding Credential Collection and Login](#) for information on the Credential Collection options.

Follow this procedure to access the Password Policy configuration page.

1. Log in to the Oracle Access Management Console as Administrator.
2. Click Application Security at the top right of the Oracle Access Management Console.
3. Click Password Policy in the Application Security console. For detailed information about the options see, [Password Policy Configuration Page](#).

24.3.1 Password Policy Configuration Page

Various options are available on the password policy configuration page.

Figure 24-1 Password Policy Configuration Page

Password Policy

This password policy will be applied to all resources protected by Oracle Access Management. Specify the password policy's details.

Password Options

Minimum Uppercase Characters <input type="text" value=""/> ^ v	Maximum Special Characters <input type="text" value=""/> ^ v
Minimum Lowercase Characters <input type="text" value=""/> ^ v	Maximum Unicode Characters <input type="text" value=""/> ^ v
Minimum Alphabetic Characters <input type="text" value=""/> ^ v	Maximum Password Length <input type="text" value=""/> ^ v
Minimum Numeric Characters <input type="text" value=""/> ^ v	Maximum Repeated Characters <input type="text" value=""/> ^ v
Minimum Alphanumeric Characters <input type="text" value=""/> ^ v	
Minimum Special Characters <input type="text" value="1"/> ^ v	
Minimum Unicode Characters <input type="text" value=""/> ^ v	
Minimum Password Length <input type="text" value="1"/> ^ v	
Minimum Unique Characters <input type="text" value=""/> ^ v	
Minimum Password Age (days) <input type="text" value=""/> ^ v	

Characters Required <input type="text"/>	
Characters Not Allowed <input type="text"/>	
Characters Allowed <input type="text"/>	
Substrings Not Allowed <input type="text"/>	

Alphabetic Character Must Start Password <input checked="" type="checkbox"/>	Can Include User's Last Name <input type="checkbox"/>
Can Include User's First Name <input checked="" type="checkbox"/>	Can Include User ID <input type="checkbox"/>
Warn After (days) <input type="text" value=""/> ^ v	* Maximum Attempts <input type="text" value="3"/> ^ v
Expire After (days) <input type="text" value=""/> ^ v	Permanent Lockout <input checked="" type="checkbox"/>
Disallow Previous Passwords <input type="text" value=""/> ^ v	Lockout Duration (minutes) <input type="text" value=""/> ^ v

Password Dictionary File <input type="text"/>	
Password File Delimiter <input type="text"/>	
Password Service URL <input type="text" value="/oam/pages/pswd.jsp"/>	

Note: Use Password Policy Configuration page to configure only Global Password Policies. Multiple Password Policies can be configured using REST API's .

[Table 24-2](#) describes the configurable Password Policy options (as read from left to right in the console). These elements are used by both the ECC and DCC.

Table 24-2 Password Policy Elements

Element	Description
Minimum Uppercase Characters	Defines the minimum number of uppercase characters required in a password.

Table 24-2 (Cont.) Password Policy Elements

Element	Description
Minimum Lowercase Characters	Sets the minimum number of lowercase characters required in a password.
Minimum Alphabetic Characters	Defines the minimum number of special characters allowed in the password.
Minimum Numeric Characters	Sets the minimum number of numeric characters required in a password.
Minimum Alphanumeric Characters	Defines the minimum number of alphanumeric characters required in a password.
Minimum Special Characters	Sets the minimum number of special characters required in a password.
Maximum Special Characters	Defines the maximum number of special characters allowed in a password.
Minimum Unicode Characters	Defines the minimum number of unicode characters required in a password.
Maximum Unicode Characters	Sets the maximum number of unicode characters allowed in a password.
Minimum Password Length	Sets the total minimum number of characters required in a password.
Maximum Password Length	Defines the total maximum number of characters allowed in a password.
Characters Required	Defines the specific characters that are required in a password. No delimiter is needed or allowed in this definition.
Characters Not Allowed	Sets the specific characters that cannot be used in a password. No delimiter is needed or allowed in this definition.
Characters Allowed	Defines all allowed characters in a password. No delimiter is needed or allowed in this definition.
Substrings Not Allowed	Specific character strings that are not allowed in a password. Use a comma as the delimiter in this definition.
Alphabetic Character Must Start Password	Specifies that the first character in a password must be alphabetic, when checked.
Can Include User's Last Name	Specifies that the user's last name is allowed in the password, when checked.
Can Include User's First Name	Specifies that the user's first name is allowed in the password, when checked.
Can Include User ID	Specifies that the user's userID is allowed in the password, when checked.
Warn after (days)	Defines the number of days before a designated date in which a user will be warned about password expiration. For example, you enter 30 in the Expires After (Days) field, and 20 in the Warn After (Days) field, and the password is created on November 1. On November 21, the user will be informed that the password will expire on December 1. This field accepts values from 0 to 999.
Maximum Attempts	Identifies the maximum number of login attempts a user can make before a lockout.
Expire after (days)	Defines the period of time (in days) that the password is valid.

Table 24-2 (Cont.) Password Policy Elements

Element	Description
Lockout Duration (minutes)	Identifies the period of time the user is locked out (in minutes) after the designated number of failed login attempts. After this period, the user can attempt a fresh login.
Permanent Lockout	specifies permanent lockout after the designated number of failed login attempts.
Disallow Last	Defines the number of previous passwords that cannot be used when the user changes her password.
Password Dictionary File	Identifies the physical file on OAM Servers that contain the list of restricted words that can not be specified in a password.
Password File Delimiter	Defines the delimiter used in the Password Dictionary file to separate various words. For example, if the file contains <code>abc, def, welcome</code> and the dictionary delimiter is comma (<code>,</code>), the words that are restricted and cannot be used in a user password are <code>abc</code> , <code>def</code> and <code>welcome</code> .
Password Service URL	Note: The Password Service URL is deprecated and works only when used with Password Policy Validation authentication module which is deprecated. To configure different pages for ECC/DCC as well as custom pages for Password Management Module, please use the <code>URL_REDIRECT</code> field to set appropriate page

24.4 Specifying Credential Collector URLs with Password Policy

Regardless of the credential collection method, you can configure one global password policy that applies to all Access Manager-protected resources (using the Password Policy Validation Module in the authentication scheme).

The relevant URLs for the credential collector and related forms must be specified as outlined in [Table 24-3](#).

Table 24-3 Specifying Credential Collectors and Related Forms for Authentication

In the . . .	For the ECC . . .	For the DCC . . .
OAM Agent Registration <i>DCC Only</i>	N/A.	Check the box beside Allow Management Operations in the OAM Agent registration page. See Also: " Enabling DCC Credential Operations "

Table 24-3 (Cont.) Specifying Credential Collectors and Related Forms for Authentication

In the . . .	For the ECC . . .	For the DCC . . .
login, error, and password pages	<p>Pages where the user enters credentials arrive out of the box on the OAM Server and require no additional settings or changes.</p> <ul style="list-style-type: none"> • Login page: /pages/login.jsp • Logout page: /pages/logout.jsp • Error page: /pages/servererror.jsp • Multi-step authentication: /pages/mfa_login.jsp 	<p>Dynamic pages for general login/logout and password policy with the DCC are excluded automatically through the OHS <code>httpd.conf/webgate.conf</code> file--you do not need to configure a policy to exclude these.</p> <p>See WebGate host directories <code>\$WEBGATE_HOME/webgate/ohs/oamsso/*</code>, <code>\$WEBGATE_HOME/webgate/ohs/oamsso-bin/*.pl</code>, and <code>\$WEBGATE_HOME/webgate/ohs/oamsso-bin/templates/*</code> for:</p> <ul style="list-style-type: none"> • Login page: /oamsso-bin/login.pl • Logout: /oamsso-bin/logout.pl • RSA SecurID login pages: /oamsso-bin/securid.pl <p>Perl Scripts for DCC-based Login and Logout</p> <p>The path name of the Perl executable must be updated in Oracle-provided Perl scripts on the WebGate host <code>\$WEBGATE_HOME/webgate/ohs/oamsso-bin/*.pl</code> to be consistent with the actual location.</p> <p>See Also: Table 22-4</p>
Password Policy, Password Service URL	<p>The Default/ECC password page is used automatically:</p> <p>Password Service URL for ECC: /oam/pages/pswd.jsp</p> <p>See Also: "Defining Your Global Password Policy"</p>	<p>Enter the DCC password page:</p> <p>Password Service URL for DCC: /oamsso-bin/login.pl</p> <p>See Also: "Locating and Updating DCC Forms for Password Policy"</p>
User Identity Store	<p>The user data object definition in the Access Manager schema is extended with attributes that enable password user status and password history maintenance. This definition is provided in an LDIF file, and must be added to each user identity store using the <code>ldapadd</code> tool. Oracle-provided LDIFs are identified in Table 24-6.</p>	<p>Same for both DCC and ECC:</p> <p>See Also:</p> <ul style="list-style-type: none"> • "Adding Key Password Attributes to the Default Store" • "Adding an Administrator to Change User Attributes After a Password Change"
Password Management Module	<p>Enter the Default Store as the <code>KEY_IDSTORE_REF</code> for each of the three plug-ins / steps (with an Error redirect on Failure):</p> <p>See Also:</p> <ul style="list-style-type: none"> • Table 22-12 • "Password Policy Validation Module" 	<p>Same for both DCC and ECC:</p>

Table 24-3 (Cont.) Specifying Credential Collectors and Related Forms for Authentication

In the . . .	For the ECC . . .	For the DCC . . .
Authentication Scheme, Challenge Redirect URL	Enter the Credential Collector host: <ul style="list-style-type: none"> • For ECC, relative URI format: /oam/server (server prepends the <i>host:port</i>) See Also: " Configuring the PasswordPolicyValidationScheme "	Enter the Credential Collector host: <ul style="list-style-type: none"> • For DCC, full URL: <code>http://dcchost:port</code> • For DCC combined with Resource Webgate: Leave empty See Also: " Configuring the PasswordPolicyValidationScheme "
Authentication Scheme, Challenge URL	Enter the Credential Collector login form relative URI: <ul style="list-style-type: none"> • For ECC: /pages/login.jsp See Also: " Configuring the PasswordPolicyValidationScheme "	Enter the Credential Collector login form relative URI: <ul style="list-style-type: none"> • For DCC: /oamssso-bin/login.pl See Also: " Configuring the PasswordPolicyValidationScheme "
Authentication Scheme, Challenge Parameters	ECC: User-defined Challenge Parameters: <ul style="list-style-type: none"> OverrideRetryLimit=0 initial_command=NONE See Also: <ul style="list-style-type: none"> • Table 22-25 • "Configuring the PasswordPolicyValidationScheme" 	DCC: User-defined Challenge Parameters: <ul style="list-style-type: none"> • creds • extracreds • MaxPostDataBytes • DCCctxCookieMaxLength • TempStateMode See Also: <ul style="list-style-type: none"> • Table 22-25 • "Configuring the PasswordPolicyValidationScheme"
Server Error Mode	Same for both DCC and ECC. See: " Setting the Error Message Mode for Password Policy Messages "	Same for both DCC and ECC. See: " Setting the Error Message Mode for Password Policy Messages "

Table 24-3 (Cont.) Specifying Credential Collectors and Related Forms for Authentication

In the . . .	For the ECC . . .	For the DCC . . .
Authentication Policy	Credential collectors in authentication policies: <ul style="list-style-type: none"> • ECC: Use any authentication scheme configured for the ECC in the application domain for the protecting Webgate (Resource Webgate) <p>See Also: "Adding Your PasswordPolicyValidationScheme to ECC Authentication Policy"</p>	Credential collectors in Authentication Policies: <p>DCC Separate from Resource Webgate:</p> <p>***Protecting (Resource) Webgate Application Domain, (Authentication Policy protecting resources), use the DCC-related Authentication Scheme.</p> <p>***DCC Webgate Application Domain, Authentication Policy protecting resources, use the DCC-related Authentication Scheme. Consider:</p> <p>--With No Action URL: DCC uses the default /oam/server/auth_cred_submit, which is automatically protected with the DCC-related authentication scheme.</p> <p>--With an Action URL: Explicitly protect the specified Action URL with the DCC Scheme.</p> <p>See Also: "Adding PasswordPolicyValidationScheme to Authentication Policy for DCC"</p>

Table 24-3 (Cont.) Specifying Credential Collectors and Related Forms for Authentication

In the . . .	For the ECC . . .	For the DCC . . .
Logout Configuration	<p>ECC:</p> <p>In the protecting (Resource) Webgate Agent registration, configure the Logout URL as shown in Table 15-3</p> <p>See "Configuring Centralized Logout for OAM WebGates"</p>	<p>DCC:</p> <ul style="list-style-type: none"> In the DCC Agent registration page the Logout Redirect URL is ignored. In the protecting (Resource) Webgate registration, define the: <pre>Logout Redirect URL: http//dcchost:port/oamssso-bin/ logout.pl</pre> <p>Note: If the Resource Webgate's Logout Redirect URL is anything other than <code>logout.*</code>, then that URL must be defined in the Logout URL parameter of the DCC Webgate registration. For example:</p> <p>If Resource Webgate registration has: Logout Redirect URL <code>http//dcchost:port/ someurl.html</code></p> <p>then DCC Webgate registration must have: Logout URL: <code>someurl.html</code></p> <ul style="list-style-type: none"> DCC: Perl path must be updated in Oracle-provided scripts. <p>See "Configuring Logout When Using Detached Credential Collector-Enabled WebGate"</p>

24.5 Oracle-Provided Password Forms

Access Manager provides several pages for user interactions during credential collection. The location can be customized, depending on the desired topology of the authentication scheme being developed.

The Credential Collectors password pages are described in [Table 24-4](#).

Table 24-4 Credential Collector Password Pages

Credential Collector	Description
ECC pages	<p>The default embedded credential collector jsp forms, by default, reside on the OAM Servers.</p> <ul style="list-style-type: none"> • Login page: /pages/login.jsp • Logout page: /pages/logout.jsp • Error page: /pages/servererror.jsp • Multi-step authentication page: /pages/mfa.jsp
DCC pages	<p>Dynamic pages general login/logout and password policy with the DCC are excluded automatically through the OHS httpd.conf/webgate.conf file--you do not need to configure a policy to exclude these. See the Webgate host:</p> <ul style="list-style-type: none"> • <code>\$WEBGATE_HOME/webgate/ohs/oamssso/*</code> • <code>\$WEBGATE_HOME/webgate/ohs/oamssso-bin/*pl</code> (update the Perl location in the first line of the login, logout, and securid scripts) • <code>\$WEBGATE_HOME/webgate/ohs/oamssso-bin/templates/*</code> <p>See Also: For details about customizing pages and messages, see the <i>Developing Applications with Oracle Access Management</i>.</p>

[Table 24-5](#) shows the password forms provided. The default pages can be customized for your enterprise, or replaced entirely with custom pages. For example, you can design, implement, and deploy a custom page that displays a different version of the login form for a mobile browser than is used for a desktop browser.

Table 24-5 Password Management Forms and Functions

Form	Function
Sign In Form	<p>The standard login form provides fields for userID and password. Clicking the Login button initiates authentication processing governed by the configured authentication module.</p> <p>See: <i>Developing Applications with Oracle Access Management</i> for details about customizing login forms.</p>

Table 24-5 (Cont.) Password Management Forms and Functions


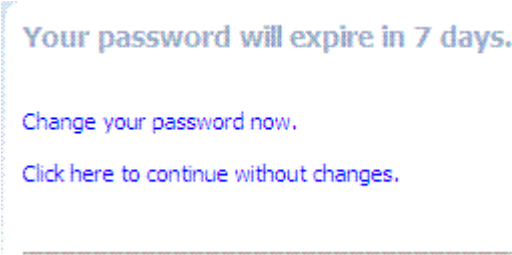
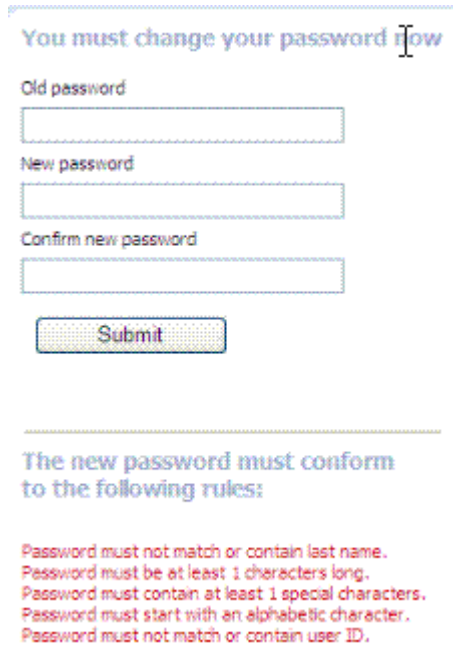
Form	Function
Sign In Error	This standard login form appears when an error occurs. The text in red identifies the errors, which can be suppressed or displayed.
	
<p>See: <i>Developing Applications with Oracle Access Management</i> for details about suppressing or displaying.</p>	
Password Expiry Notification	The following message appears to inform the user that her password will expire, based on the notification policy.
	

Table 24-5 (Cont.) Password Management Forms and Functions

Form	Function
Change Password Form	Based on password expiration policy configuration, the following window appears to enforce the policy and require user to change his password.



You must change your password now

Old password

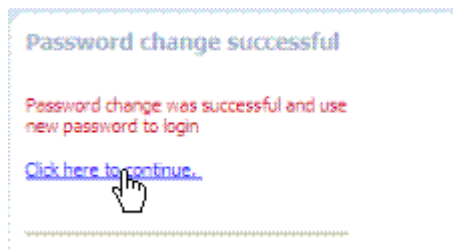
New password

Confirm new password

The new password must conform to the following rules:

- Password must not match or contain last name.
- Password must be at least 1 characters long.
- Password must contain at least 1 special characters.
- Password must start with an alphabetic character.
- Password must not match or contain user ID.

Password Change Success	The following message appears to confirm the password change was successful.
-------------------------	--



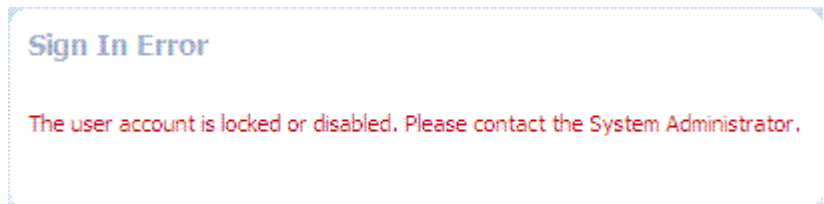
Password change successful

Password change was successful and use new password to login

[Click here to continue.](#)

Table 24-5 (Cont.) Password Management Forms and Functions

Form	Function
Locked or Disabled User Account	Based on the password policy, user account lockout occurs when supplied credentials fail during the maximum allowed login attempts.



24.6 Managing Global Password Policy

Regardless of whether you choose the ECC or DCC, you can configure a global password policy that applies to all Access Manager-protected resources. In addition, multiple password policies are also supported with configuration changes.

Authentication involves determining which credentials a user must supply when requesting access to a resource, gathering credentials, and returning a response that is based on the results of credential validation. Access Manager authentication processing relies on an authentication module (or plug-in) to define the rules governing requirements and transmission of information to the back-end authentication scheme. By default, Access Manager supports using the OAM Server Embedded Credential Collector (ECC) for authentication processing. However, you can also configure an OAM WebGate to use as an detached credential collector (DCC) instead.

Note:

Both the ECC and DCC facilitate multi-step authentication flows where credentials are not provided all at once. This increases the flexibility of interaction with users or programmatic entities for the purpose of collecting authentication-related information. For more information, see [Orchestrating Multi-Step Authentication with Plug-in Based Modules](#).

The following overview provides links to topics that describe how to configure and use the password policy. Unless explicitly stated, all tasks apply equally to the ECC and DCC. Skip any tasks that do not apply to your deployment.

Password policy management includes

1. [Defining Your Global Password Policy](#)
2. [Adding Key Password Attributes to the Default Store](#)
3. [Adding an Administrator to Change User Attributes After a Password Change](#)

4. [Configuring Password Policy Authentication](#)
5. **DCC:** [Configuring OAM WebGate and Authentication Policy for DCC](#)
6. [Completing Password Policy Configuration](#)
7. [Testing Your Multi-Step Authentication](#)

24.6.1 Defining Your Global Password Policy

Users with Oracle Access Management Administrator credentials can define a common password policy based on enterprise-defined requirements.



Note:

The only difference between a global password policy for the ECC versus the DCC is `Password Service URL`, which is credential collector-specific and defaults to ECC pages as shown in Step 2.

The specifications in this example are for illustration only. Your environment will be different.

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Password Policy**.
3. On the Password Policy page, enter the Password Service URL for the desired credential collector login page (ECC or DCC, [Table 24-3](#)).

ECC Password Service URL	DCC Password Service URL
<code>/oam/pages/pswd.jsp</code>	<code>/oamssso-bin/login.pl</code>

4. On the Password Policy page, enter values ([Table 24-2](#)) based on requirements for your enterprise. For example:
 - Warn After 3
 - Expire After 20
 - Permanent Lockout (Disable)
 - Lockout duration 1
 - Minimum Special Characters 1
5. Click **Apply** to submit the policy.
6. Proceed as needed for your environment; skip any tasks that have been completed already:
 - [Adding Key Password Attributes to the Default Store](#)
 - [Adding an Administrator to Change User Attributes After a Password Change](#)

24.6.2 Designating the Default Store for Your Password Policy

The Password Policy operates only with the designated Default Store. Administrator roles and credentials must reside in the System Store.



See Also:

["About using the System Store for User Identities"](#)

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. In the Configuration console, click **User Identity Stores**.
3. **Set the System Store:** Administrator roles and credentials must reside in this store.
 - a. Open the page of the store to designate as the System Store.
 - b. Check **Set as system store** (for domain wide authentication and authorization operations).
 - c. Click **Apply**.
 - d. **Add Administrators:** See ["Managing Administrator Roles"](#).
 - e. **Authentication Module:** Set the LDAP Authentication Module used by the `OAMAdminConsoleScheme` (authentication scheme) to use this System Store.
 - f. Configure one or more authentication plug-ins to use this store, as described in ["Orchestrating Multi-Step Authentication with Plug-in Based Modules"](#).
4. **Set Default Store:** This store is required for Password Policy and migration when patching.
 - a. Open the page of the store to designate as the Default Store.
 - b. Check the box beside Set as default store.
 - c. **Authentication Module:** Locate `OAMAdminConsoleScheme` and confirm that the LDAP module does not refer to this store. See ["Managing Native Authentication Modules"](#).
 - d. **Authorization Policy Conditions:** Choose the desired user identity store when setting Identity Conditions in Authorization Policies. See ["Defining Authorization Policy Conditions"](#).
5. Close the registration page.

24.6.3 Adding Key Password Attributes to the Default Store

The Password Policy operates only with the designated Default Store.

This section provides steps for extending the default store schema for Oracle Access Management password policy operations.

- [LDIF Files and Key Password Attributes for Password Policy](#)
- [Extending the Default Store Schema with Password Policy Attributes](#)

24.6.3.1 LDIF Files and Key Password Attributes for Password Policy

The LDIF (Lightweight Directory Interchange Format) files distributed as part of Access Manager are meant to extend the schema with required object classes. Generally, these are applied using the Access Manager and Oracle Identity Management wiring has been performed manually. The user data object definition in the Access Manager schema is extended with attributes that enable password user status and password history maintenance. This definition is provided in an LDIF file, and must be added to each user identity store using the `ldapadd` tool.

Oracle-provided LDIFs are identified in [Table 24-6](#).



Note:

OAM_HOME contains installed files necessary to host Oracle Access Management. OAM_HOME resides within the directory structure of the Middleware home (\$MW_HOME).

Table 24-6 Location of Oracle-provided LDIFs for LDAP Providers

LDAP Provider	LDIF Location
OID: Oracle Internet Directory	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ OID_PWDPersonSchema.ldif
OVD: Oracle Virtual Directory	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ OVD_PWDPersonSchema.ldif
AD: Microsoft Active Directory	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ AD_PWDPersonSchema.ldif
SJS: sun Java System Directory	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ IPLANET_PWDPersonSchema.ldif
eDirectory: Novell eDirectory	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ EDIR_PWDPersonSchema.ldif
ODSEE: Oracle Directory Server Enterprise Edition	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ IPLANET_PWDPersonSchema.ldif
OOD: Oracle Unified Directory	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ OOD_PWDPersonSchema.ldif
SLAPD: OpenLDAP Directory	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ OLDAP_PWDPersonSchema.ldif
IBM: OBM Tivoli Directory	\$OAM_ORACLE_HOME/server/pswdservice/ldif/ TIVOLI_PWDPersonSchema.ldif

 **Note:**

The above ldif files extension for userid stores are required only when using Password Policy Validation Module. Password Policy Validation Module is deprecated and may not be available in future releases. Instead Password Management Module which handles same functionality with new features such as multiple password policies, password reset with OTP and various modes are available. Please refer, [Table 24-9](#) to extend the directory for Password Management Service.

The attributes that enable password user status and password history maintenance are shown in [Table 24-7](#). The user data object of each user identity store must include the attributes shown in [Table 24-7](#). These can be added with the `ldapadd` tool, LDIF (Lightweight Directory Interchange Format) file.

Table 24-7 Key Password Attributes in a Password Policy

Attribute	Description	Format and Values
obPasswordCreationDate	The date and time used to calculate (at the time of user login) whether the password has expired and whether a warning needs to be issued.	YYYY-MM-DDThh:mm:ssZ
obPasswordHistory	Used to track the number of last passwords used. Access Manager understands 12c oblixPersonPwdPolicy format and changes it to new format.	New format: password1###password2### Previous format: passwordX = SHA256 (password+canonical userid)
obPasswordChangeFlag	Used during forced password change for first time user login (or forced password change initiated by the Administrator). Note: Forced password change is administered using REST API's. Administrator can invoke these Rest API's. See, Rest API for Password Management in Oracle Access Manager	Boolean string value. true false Empty string represents false.
obuseraccountcontrol	Used to represent a disabled user.	Non-encrypted string value. activated deactivated Empty string represents "activated".
obpasswordexpirydate	The time after which the user password is considered to be expired.	YYYY-MM-DDThh:mm:ssZ Empty value represents not expired.

Table 24-7 (Cont.) Key Password Attributes in a Password Policy

Attribute	Description	Format and Values
obLockoutTime	The time up to which the user is considered to be locked out due to too many login attempts.	Epoch value (in seconds) representing time in the future. Seconds (since 01 January, 1970)
obLoginTrvCount	The number of consecutive login failures by the user. This counter is reset on the first correct password entry.	Non-encrypted integer value. 1, 2, 3, and so on.
oblastsuccessfullogin	The time of the last successful login.	YYYY-MM-DDThh:mm:ssZ
oblastfailedlogin	The time of the last failed login.	YYYY-MM-DDThh:mm:ssZ

24.6.3.2 Extending the Default Store Schema with Password Policy Attributes

You can skip this task if the environment has been configured using `idmConfigTool -prepareIDStore`. If your user identity store has not been extended with the `oblis` schema, you must update the schema to include the object classes required by the password service. LDAP tools should be run from the `/bin` directory beneath `$OAM_HOME`.

The following procedure illustrates extending the Oracle Internet Directory schema. Your environment might be different.

1. Use the following command to update the Oracle Internet Directory object classes of the designated Default Store required by the password service:

```
ldapadd -D "cn=orcladmin" -w <password> -h <hostname> -p 3060 -x -f
$OAM_ORACLE_HOME/server/pswdservice/ldif/OID_PWDPersonSchema.ldif
```

2. Proceed to ["Adding an Administrator to Change User Attributes After a Password Change"](#).

24.6.4 Adding an Administrator to Change User Attributes After a Password Change

You can modify the Default Store (Oracle Internet Directory in this example) to use a different privileged account as the Bind DN. This enables sufficient privileges to change user attributes after a password change.

Prerequisites

Register a supported LDAP store and designate it as the Default Store. Ensure that the user you add is defined within the Default Store.



See Also:

["Managing Administrator Roles"](#)

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. In the Configuration console, click **Administration**.
3. **Add a New Administrator:**
 - a. In the **Administration** page, click **Grant**.
 - b. In the dialog that appears, click **Search**.
 - c. Select the desired role from the **Roles** drop-down list and click **Add Selected** to grant it to the selected user.
 - d. Click **Apply** to submit the changes.
4. Proceed with "[Configuring Password Policy Authentication](#)".

24.7 Configuring Password Policy Authentication

After preparing your password policy, Default Store, and Administrator, you can develop your authentication module and scheme.

- [Password Policy Validation Module](#)
- [Configuring the PasswordPolicyValidationScheme](#)
- [Adding Your PasswordPolicyValidationScheme to ECC Authentication Policy](#)—If you are using the DCC, skip this topic and go to "[Configuring OAM WebGate and Authentication Policy for DCC](#)"
- [Supporting DCC Authentication Schemes with Pre-Authentication Rules](#)

24.7.1 Password Policy Validation Module

You must also configure the Password Policy Validation Authentication Module to use the Default Store.



Note:

There are no credential collector dependencies when defining the Password Policy Validation Module for authentication.

Password Policy Validation Module is deprecated and is replaced using Password Policy Management Module. See [Configuring Password Policy Management Module](#).

A sample module is shown in [Figure 24-2](#). The User Password Status Step is the unique step that relies on the `UserPasswordPolicyPlugin`.



Note:

UserPasswordPolicyPlugin is supported only when using LDAP based authentication modules. It does not work with non LDAP authentication modules.

Figure 24-2 Password Policy Validation Authentication Module with Orchestrated Plug-ins

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View ▾ + × Detach

Step Name	Description	Plug-in Name
User Identification Step		UserIdentificationPlugin
User Authentication Step		UserAuthenticationPlugin
User Password Status Step		UserPasswordPolicyPlugin

Step Details Save Cancel

Step Name User Identification Step

Description

Plug-in Name UserIdentificationPlugin

Plug-in Parameters

KEY_LDAP_FILTER

KEY_IDENTITY_STORE_REF

KEY_SEARCH_BASE_URL

Each step identifies the action provided by a specific named plug-in.



See Also:

"Orchestrating Multi-Step Authentication with Plug-in Based Modules".

Figure 24-3 shows the orchestration of steps within the authentication module. For more information on modules and steps, see "Pre-populated Plug-ins for Configuring Access Manager with Multi-Step Authentication".

Figure 24-3 Step Orchestration for Password Policy Validation Module

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General Steps **Steps Orchestration**

You can specify the initial step

* Initial Step

Name	Description	On Success	On Failure	On Error
User Identification Step		<input type="text" value="User Authentication Step"/>	<input type="text" value="failure"/>	<input type="text" value="failure"/>
User Authentication Step		<input type="text" value="User Password Status Step"/>	<input type="text" value="User Password Status Step"/>	<input type="text" value="failure"/>
User Password Status Step		<input type="text" value="success"/>	<input type="text" value="failure"/>	<input type="text" value="failure"/>

Table 24-8 describes the Password Policy Validation module step details that you specify.

Table 24-8 User Password Step Details

Step Name	Step Details	Description
User Identification Step	KEY_LDAP_FILTER	Add the LDAP filter to the KEY_LDAP_FILTER attribute. Only standard LDAP attributes can be used when defining an LDAP search filter. For example: <code>(uid={KEY_USERNAME})</code> See Also: Table 25-15 and your vendor documentation for the exact syntax for your identity store
User Identification Step	KEY_IDENTITY_STORE_REF	The name of the registered Identity Store containing the module users. Default: The registered Default Store.
User Identification Step	KEY_SEARCH_BASE_URL	Base URL for user searches. For example: <code>dc=us,dc=example,dc=com</code>
User Authentication Step	KEY_IDENTITY_STORE_REF	The name of the registered Identity Store containing the module users. Default: The registered Default Store.
User Authentication Step	KEY_PROP_AUTHN_EXCEPTION	Enable or disable the propagation of LDAP errors. "KEY_PROP_AUTHN_EXCEPTION" needs to be set to TRUE when the Authentication module has "Password Policy Plugin" as the next step of plugin execution; for example, when the module has Authentication Plugin ->Password Plugin, change this parameter to TRUE.

Table 24-8 (Cont.) User Password Step Details

Step Name	Step Details	Description
User Password Status Step	PLUGIN_EXECUTION_MODE	<p>The execution mode of plug-in. Depending upon the configuration, this plug-in can operate either alone or with other default plug-ins. Values are one of the following:</p> <ul style="list-style-type: none"> PSWDONLY: The most preferred configuration where only the password status is determined. The ID and authentication must be performed using the UserIdentification and UserAuthentication Plugins. AUTHWITHPSWD: Both authentication and password are performed using this plug-in. AUTHONLY: Only the user identification and authentication is performed using this plug-in <p>Default: PSWDONLY</p>
User Password Status Step	OBJECTCLASS_EXTENSION_SUPPORTED	<p>The object classes "oblixpersonpwdpolicy" and "oblixorgperson" are required to be present in the OAM user's entry for successful execution of this plugin. If this parameter is FALSE, the plugin will not add these object classes. If this parameter is TRUE, the plugin will try to add these object classes to the user's entry if the current user's entry does not already have them present.</p> <p>Default: FALSE</p>
User Password Status Step	KEY_IDENTITY_STORE_REF	<p>The name of the registered Identity Store containing the module users.</p> <p>Default: The registered Default Store.</p>
User Password Status Step	NEW_USERPSWD_BEHAVIOR	<p>Configures retroactive behavior of the new-user password-policy. Values are either:</p> <ul style="list-style-type: none"> FORCEPASSWORDCHANGE: Forces a password change. NOFORCEPASSWORDCHANGE: The password policy change does not affect user passwords that are already set. <p>Default: NOFORCEPASSWORDCHANGE</p>
User Password Status Step	URL_ACTION	<p>The type of servlet action needed for redirecting the user to the specific password page for expiry and warning pages. Values can be either:</p> <ul style="list-style-type: none"> REDIRECT_POST REDIRECT_GET FORWARD <p>Default: REDIRECT_POST</p>
User Password Status Step	DISABLED_STATUS_SUPPORT	<p>Specifies whether the disabled status is to be supported and acted upon in this password service. Valid values are either True or False.</p> <p>Default: TRUE</p>

Prerequisites

[Defining Your Global Password Policy](#)



Note:

There are no credential collector dependencies when defining the Password Policy Validation Module. Enter the Default Store as the KEY_IDSTORE_REF for each of the three plug-ins (with an Error redirect on Failure).

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Authentication Modules**. in the **Plug-ins** section.
3. In the Authentication Modules page, click **Search**, then click **Password Policy Authentication Module**.
4. Select the **Steps** tab; for each of the three steps add the **Default Store** name in the field beside KEY_IDSTORE_REF (Save after each change). For example:
 - a. **User Identification Step**
KEY_IDSTORE_REF: *OID*
Save.
 - b. **User Authentication Step**
KEY_IDSTORE_REF: *OID*
Save.
 - c. **User Password Status Step**
KEY_IDSTORE_REF: *OID*
Save.
5. Click **Apply**.
6. Proceed to "[Configuring the PasswordPolicyValidationScheme](#)".

24.7.2 Configuring the PasswordPolicyValidationScheme

Users with Administrator credentials can configure the PasswordPolicyValidationScheme.

You can have multiple authentication schemes for use with the global password policy.

 **Note:**

In case of an upgraded environment, the PasswordPolicyValidationScheme will be using the original Password Policy Validation Module. Customers who want to use the following features:

- Multiple password policy feature
- Forgot password using OTP
- Changing user status using REST API

need to manually change the module that PasswordPolicyValidationScheme is using to PasswordPolicyManagementModule.

Differences between values for the ECC versus the DCC include (Table 24-3):

- Challenge Redirect URL: *Credential Collector host and port*
- Challenge URL: *Credential Collector Pages*
- Challenge Parameters: [Table 22-24](#)

Prerequisites

[Password Policy Validation Module](#)

 **See Also:**

["Managing Authentication Schemes"](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Authentication Schemes** in the **Access Manager** section.
3. In the Search Authentication Schemes page, click **Search**, then click **PasswordPolicyValidationScheme**.
4. Set up the scheme for your environment. For example:
 - Authentication Level **2**
 - Default (*blank*)
 - Challenge Method: **Form**
 - Challenge Redirect URL: `http://CredCollector_host:port/`
 - Authentication Module: **Password Management Module**
 - Challenge URL: `/CredCollector_pages/`
 - Context Type: **External**
 - Challenge Parameters:

ECC Challenge Parameters	DCC Challenge Parameters
<pre>OverrideRetryLimit=0 initial_command=NONE</pre>	<pre>OverrideRetryLimit=0 creds=userid password</pre> <p>See Also: Table 22-25</p> <p>action If not specified, the default for both ECC and DCC is <code>/oam/server/auth_cred_submit</code>.</p> <p>DCCctxCookieMaxLength (default is 4096)</p> <p>TempStateMode controls how the DCC stores the OAM Server state: cookie or form (the default) as specified with the parameter's value.</p> <p>MaxPostDataBytes Restricts the maximum number of bytes of POST data submitted as user credentials.</p> <p>creds Whatever is passed must be specified in the <code>obMap</code> credentials parameter of the <code>ObUserSession</code> object, as described in the <i>Developing Applications with Oracle Access Management</i></p>

5. Click **Apply**.
6. Proceed to "[Adding Your PasswordPolicyValidationScheme to ECC Authentication Policy](#)".

24.7.3 Adding Your PasswordPolicyValidationScheme to ECC Authentication Policy

A user with Administrative privileges can use the PasswordPolicyValidationScheme configured for the ECC in the application domain of the protecting Webgate (Resource Webgate).

Prerequisites

[Configuring the PasswordPolicyValidationScheme](#)

1. **ECC:** In the console, search for and open the appropriate Application Domain. (See "[Searching for an Existing Application Domain](#)").



See Also:

["Adding Your PasswordPolicyValidationScheme to ECC Authentication Policy"](#)

2. **ECC:** Protect Resources using the **PasswordPolicyValidationScheme**:
 - a. Find and open your **Protected Resource Policy** on the Authentication Policies tab (see "[Viewing or Editing an Authentication Policy](#)"):
 - Authentication Policies
 - Protected Resource Policy
 - b. Select **PasswordPolicyValidationScheme** for the **Protected Resource Policy** (Authentication Scheme) and click **Apply**.

- c. Finish updating your Authentication and Authorization policies, as desired ([Managing Policies to Protect Resources and Enable SSO](#)).
3. Proceed as needed for your environment:
 - **ECC:** [Completing Password Policy Configuration](#)
 - **DCC:** [Configuring OAM WebGate and Authentication Policy for DCC](#)

24.7.4 Supporting DCC Authentication Schemes with Pre-Authentication Rules

When DCC authentication schemes are used, pre-authentication rules are unable to distinguish between internal and external URLs from different proxies. You have to create a new pre-authentication rule using *returnHost* parameter to support DCC authentication schemes. Restart the server to use the newly added DCC scheme.

Pre-authentication rules allow you to define a policy that can either block access to the user or allow OAM to use a different authentication scheme based on certain conditions.

The *host* parameter in the request data allows pre-authentication rules to be executed against the host name of a protected resource. When the request is originating from a DCC WebGate, the *host* parameter is unable to distinguish between internal and external URLs from different proxies. If you want the DCC WebGate to work with the proxy, you have to create a new pre-authentication rule as follows:

```
request.returnHost.lower().find('<proxy_host_name>')>0
```

The *returnHost* parameter has the proxy host name for internal and external URLs irrespective of whether the request is originated from a ECC or DCC WebGate. When you access the resource through the specified proxy, the authentication scheme is switched as specified in the new pre-authentication rule. In case of other configured proxy, the original authentication scheme specified in the **Authentication Policy** tab is retained.

24.8 Completing Password Policy Configuration

Administrators can set error mode for password policy messages, override native LDAP password policy validation, and perform evaluations to confirm the deployment is working as required.

These tasks are the same regardless of the credential collector you have configured. Perform the following tasks to complete your password policy configuration:

- [Setting the Error Message Mode for Password Policy Messages](#)
- [Overriding Native LDAP Password Policy Validation](#)
- [Disabling ECC Operation and Using DCC Exclusively](#)
- [Testing Your Multi-Step Authentication](#)

24.8.1 Setting the Error Message Mode for Password Policy Messages

Users with administrative privileges can set the Server Error Mode for password policy messages.

[Figure 24-4](#) shows the Access Manager settings.

Figure 24-4 Server Error Mode for Password Management

Access Manager Settings Apply Revert
 The following settings apply to the Access Manager service.

▲ **Load Balancing**

* OAM Server Host * OAM Server Protocol

* OAM Server Port * Server Error Mode

Prerequisites

- [Managing Global Password Policy](#)
 - [Configuring Password Policy Authentication](#)
 - Optional: [Configuring OAM WebGate and Authentication Policy for DCC](#)
1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
 2. In the Configuration console, select **Access Manager** from the **Settings** drop-down list.
 3. In the **Load Balancing** section, set the **Server Error Mode** to **Internal**.
 4. Click **Apply**.
 5. Proceed with "[Overriding Native LDAP Password Policy Validation](#)".

24.8.2 Overriding Native LDAP Password Policy Validation

You need to disable native LDAP password policy validation before the non-native password policy can be used.

For example, with Oracle Internet Directory registered for Oracle Access Management, native password policy is generally located as follows:

```
dn:
cn=default,cn=pwdPolicies,cn=Common,cn=Products,cn=OracleContext,<DOMAIN_CONTAINER>
```

▲ Caution:

Disabling the native LDAP password policy validation leaves no enforcement for direct LDAP operations. There are various password policies in Oracle Internet Directory, including one in the following:

```
dn: cn=default,cn=pwdPolicies,cn=Common,cn=Products,cn=OracleContext
```

However, this might not apply to your domain.

You can disable the Oracle Internet Directory password policy by setting the `orclpwdpolicyenable` parameter to zero (0).

**See Also:**

The various attributes described in *Administering Oracle Internet Directory*

The following procedure is only an example. Your environment will be different.

Prerequisites**Setting the Error Message Mode for Password Policy Messages**

1. Refer to the manual from your LDAP directory vendor.
2. **Oracle Internet Directory:** Disable native policy by setting `orclpwdpolicyenable` to zero (0).
 - Confirm the location of the password policy for your domain.
 - When you are sure you have the proper native LDAP policy, disable the policy. For example:

```
orclpwdpolicyenable = 0
```
3. Proceed as follows, depending on your deployment:
 - ["Disabling ECC Operation and Using DCC Exclusively"](#)
 - ["Testing Your Multi-Step Authentication"](#)
 - [Configuring Centralized Logout for Sessions Involving OAM WebGates: "Configuring Logout When Using Detached Credential Collector-Enabled WebGate"](#)

24.8.3 Disabling ECC Operation and Using DCC Exclusively

You can skip this task to allow the DCC and ECC to co-exist, and maintain authentication schemes and policies for both credential collectors. To disable ECC, you must edit the `oam-config.xml` file. Generally, Oracle recommends not editing `oam-config.xml`. Changes to this file could result in lost data or overwriting of the file during data sync operations. However, there is no other way to disable the ECC completely in favor of the DCC.

**Note:**

After disabling the ECC, access to resources protected by schemes and policies that rely on the ECC will be prohibited, including access to the Oracle Access Management Console.

Prerequisites**Configuring OAM WebGate and Authentication Policy for DCC**

1. Make your changes on the node running the AdminServer to minimize possible conflicts that another AdminConsole user might make.
2. Locate the `ECCEnabled` parameter in the `OAMServicesDescriptor` section and make the changes shown here in bold:

```
<Setting Name="OAMServicesDescriptor" Type="htf:map">
  ... ..
  <Setting Name="ECCEnabled" Type="htf:map">
    <Setting Name="ServiceStatus" Type="xsd:boolean">false</Setting>
  </Setting>
```

See [Updating OAM Configuration](#).

3. Proceed to "[Testing Your Multi-Step Authentication](#)".

24.8.4 Testing Your Multi-Step Authentication

You can perform a number of evaluations to confirm that your deployment is working properly.



See Also:

[Configuring Centralized Logout for Sessions Involving OAM WebGates](#)

1. Confirm access after login:
 - a. Open a new browser and request a resource.
 - b. Log in with your user credentials.
 - c. Confirm that you have access to the resource.
2. Confirm no access on incorrect login:
 - a. Open a new browser and request a resource.
 - b. Log in with incorrect user credentials.
 - c. Confirm that you must re-authenticate.
3. Confirm lockout after exceeding maximum incorrect login attempts:
 - a. Open a new browser and request a resource.
 - b. Log in with incorrect user credentials repeatedly.
 - c. Confirm that the user account is locked.
4. Modify and evaluate your password expiry policy:
 - a. Log in to the Oracle Access Management Console.
 - b. In your password policy, reset the expiry and lockout periods ([Table 24-2](#)) so that you will see warnings on your next login.
 - c. Save the policy updates.
 - d. Open a new browser and request a resource.
 - e. Verify the warning page appears advising that the password will expire.
 - f. Click the link to continue without password change.
5. Change your password:
 - a. Open a new browser and request a resource.
 - b. On the password expiry warning page, click the link to change your password.

- c. On the password change page, enter your correct old password.
- d. In the new password field, enter a different new password that does not follow the password policy and confirm the password validation error.
- e. Enter a new password that meets requirements and confirm success and access to the resource.

24.9 Configuring the PasswordManagementPlugin

The Password Management policy plugin handles the password related flows during login. Configuring the Password Management policy plugin is the most critical step in making sure that OAM and OIG LDAP applications can work in tandem.

Using the Password Management plugin in OAM makes sure that password features act across both OAM and OIG in similar ways. This section contains the following information:

- [Configuring Password Policy for PasswordManagement Service](#)
- [Extending the LDAP Definitions](#)
- [Configuring Password Policy Management Module](#)
- [Configuring Password Policy Management Module](#)
- [Setting up the Forgot Password Module](#)
- [Configuring Forgot Password using OTP](#)

24.9.1 Configuring Password Policy for PasswordManagement Service

Note that the password policy in OAM should be in sync with that of OAM LDAP to work consistently between both products.

See [Accessing Password Policy Configuration Page](#) for details. It is up to the administrator to ensure that the policies are indeed the same and consistent.

24.9.2 Extending the LDAP Definitions

Depending on the type of the directory, add the required objectclass schema definitions so that the LDAP directory can use these to extend the user objectclass. The appropriate schema files are located in \$IDM_HOME/modules/oracle.idm.ipf_11.1.2/scripts/ldap.

[Table 24-9](#) documents the LDIF file to use with supported LDAP directories.

Table 24-9 Included LDIF Schema Files


LDAP Directory	LDIF Schema File
OID	OID_OblixSchema.ldif, OID_OracleSchema.ldif
AD	AD_OblixSchema.ldif, AD_OracleSchema.ldif
OUD	OUD_OblixSchema.ldif, OUD_OracleSchema.ldif
ODSEE	IPLANET_OblixSchema.ldif, IPLANET_OracleSchema.ldif
OPENLDAP	OLDAP_OblixSchema.schema, OLDAP_OracleSchema.schema
OVD	OVD_OblixSchema.ldif, OVD_OracleSchema.ldif
Tivoli	TIVOLI_OblixSchema.ldif, TIVOLI_OracleSchema.ldif

Table 24-9 (Cont.) Included LDIF Schema Files

LDAP Directory	LDIF Schema File
EDIR	EDIR_OblixSchema.ldif, EDIR_OracleSchema.ldif

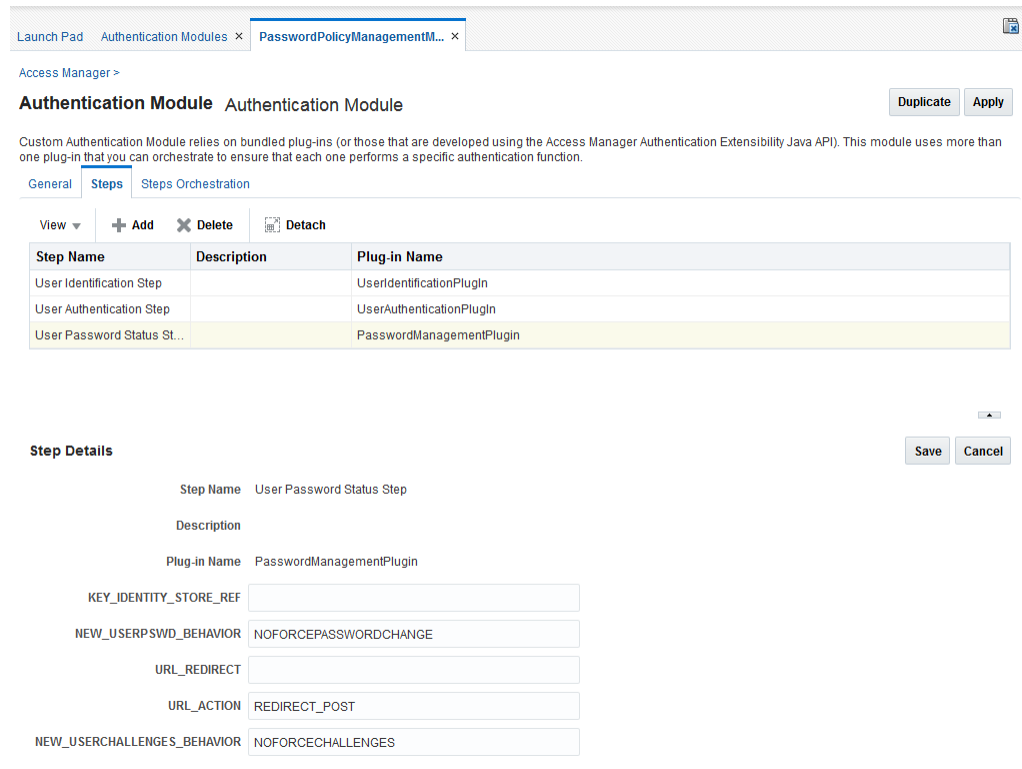
24.9.3 Configuring Password Policy Management Module

You must configure the Password Policy Management Module to configure the store that you want to be protected by Password Policy Management Module.

 **Note:**

There is no credential collector dependencies when defining the Password Policy Management Module for authentication.

Sample module is as shown in the Fig. below. The User Password Status Step is the unique step that relies on the PasswordManagementPlugin.



Access Manager >

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View

Step Name	Description	Plug-in Name
User Identification Step		UserIdentificationPlugin
User Authentication Step		UserAuthenticationPlugin
User Password Status St...		PasswordManagementPlugin

Step Details Save Cancel

Step Name User Password Status Step

Description

Plug-in Name PasswordManagementPlugin

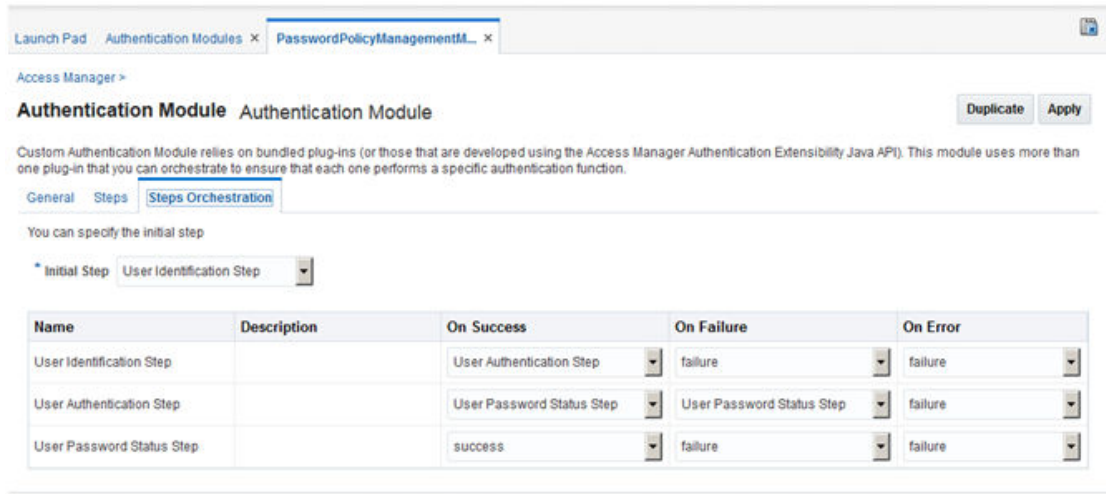
KEY_IDENTITY_STORE_REF

NEW_USERPSWD_BEHAVIOR NOFORCEPASSWORDCHANGE

URL_REDIRECT

URL_ACTION REDIRECT_POST

NEW_USERCHALLENGES_BEHAVIOR NOFORCECHALLENGES



The following table describes the Password Policy Validation module step details that you should specify:

Step Name	Step Details	Description
User Identification Step	KEY_LDAP_FILTER	Add the LDAP filter to the KEY_LDAP_FILTER attribute. Only standard LDAP attributes can be used when defining an LDAP search filter. For example: (uid={KEY_USERNAME})
User Identification Step	KEY_IDENTITY_STORE_REF	The name of the registered Identity Store containing the module users. Default: The registered Default Store.
User Identification Step	KEY_SEARCH_BASE_URL	Base URL for user searches. For example: dc=us,dc=example,dc=com
User Authentication Step	KEY_IDENTITY_STORE_REF	The name of the registered Identity Store containing the module users. Default: The registered Default Store.
User Authentication Step	KEY_PROP_AUTHN_EXCEPTION	Enable or disable the propagation of LDAP errors. KEY_PROP_AUTHN_EXCEPTION needs to be set to TRUE when the Authentication module has "Password Policy Plugin" as the next step of plugin execution. For example, when the module has Authentication Plugin ->Password Plugin, change this parameter to TRUE.

Step Name	Step Details	Description
User Password Status Step	KEY_IDENTITY_STORE_REF	The name of the registered Identity Store containing the module users. Default: The registered Default Store.
User Password Status Step	NEW_USERPSWD_BEHAVIOR	Configures retroactive behavior of the new-user password-policy. Values are either: <ul style="list-style-type: none"> FORCEPASSWORDCHANGE: Forces a password change. NOFORCEPASSWORDCHANGE: The password policy change does not affect user passwords that are already set. Default: NOFORCEPASSWORDCHANGE
User Password Status Step	URL_REDIRECT	The URL to Redirect the password pages. In case of DCC, the page needs to be specified as /oamssso-bin/login.pl
User Password Status Step	URL_ACTION	The type of servlet action needed for redirecting the user to the specific password page for expiry and warning pages. Values can be either: <ul style="list-style-type: none"> REDIRECT_POST REDIRECT_GET FORWARD Default: REDIRECT_POST
User Password Status Step	NEW_USERCHALLENGES_BEHAVIOR	Not supported.



Note:

There is no credential collector dependencies when defining the Password Policy Management Module. Enter the Default Store as the `KEY_IDSTORE_REF` for each of the three plug-ins (with an Error redirect on Failure).

Password Policy Validation Module is deprecated and is replaced using Password Policy Management Module.

See [Configuring the PasswordPolicyValidationScheme](#)

24.9.4 Setting up the Forgot Password Module

The forgot password feature in OAM can be accomplished using One Time Pin (OTP) generation and ChangePassword using OTP REST APIs.

The administrator can setup forgot password URL by following the procedure documented in [Administering the Forgot Password URL](#).

OTP (One Time Pin) can be generated for a OAM user by using Adaptive Authentication Plugin. Once enabled, users can change their passwords using OTP and password change REST APIs through application which can orchestrate these REST calls.

The forgot password feature in OAM can be accomplished using One Time Pin (OTP) generation and ChangePassword using OTP REST APIs.

See also,

[REST API for Password Management in Oracle Access Manager](#)

[REST API for OTP Forgot Password in Oracle Access Manager](#)

The sample forgot password application can be downloaded from the OTN location. Please create a support request to get the forgot password sample application.

24.9.5 Configuring Forgot Password using OTP

The forgot password feature in OAM can be accomplished using One Time Pin (OTP) generation and ChangePassword using OTP Rest APIs. The following sections provide the setup steps required for enabling forgot password flow using OTP in OAM

Directory Setup

1. Create an OID profile in OAM, add the required `objectclasses()` and add it as the default idstore in OAM. See [Creating an Identity Directory Service Profile](#) for more information on creating an ID profile.
2. Run the following command to add ldif files. See [Extending the LDAP Definitions](#) for more information.

```
ldapadd -D <DIRECTORY_USERNAME> -w <DIRECTORY_PASSWORD> -h  
<DIRECTORY_HOST_NAME> -p <DIRECTORY_PORT> -f $MW_HOME/idm/modules/  
oracle.idm.ipf_12.2.2/scripts/ldap/OID_OblixSchema.ldif  
ldapadd -D <DIRECTORY_USERNAME> -w <DIRECTORY_PASSWORD> -h  
<DIRECTORY_HOST_NAME> -p <DIRECTORY_PORT> -f $MW_HOME/idm/modules/  
oracle.idm.ipf_12.2.2/scripts/ldap/OID_OracleSchema.ldif
```

Note:

Ensure that the LDAP directory has password management enabled.

Ensure that the LDAP profile is configured as the default store in the User Identity Stores

Setup Related to OTP Rest API

These steps are required to be able to use the OTP Rest service for creating and validating the OTP for users successfully.

1. Add `otprestusergroup` to OAM admin user in the LDAP directory.
In case of admin user being `weblogic`, add the privilege in embedded ldap in `weblogic` console. Add a group called `otprestusergroup` to groups and add `weblogic` user to this group.
2. Enable UMS in Adaptive Auth Plugin. See [Configuring the Adaptive Authentication Plug-in in the Oracle Access Management Console](#)

 **Note:**

Ensure the settings `UmsAvailable = true` and `UmsClientUrl = < has the relevant client url>`

Use the following WLST command line script to set the credentials for the Oracle User Messaging Service

```
cd <MW_HOME>/oracle_common/common/bin
./wlst.sh
connect()
createCred(map="OAM_CONFIG", key="umsKey", user="weblogic",
password="welcome1")
```

 **Note:**

For further information on OTP Rest APIs in OAM refer REST API for Multifactor Authentication One Time PIN in Oracle Access Manager

Setup for forgot password link on default login page

On the OAM console, *Enable* the Adaptive Authentication Service. This is a prerequisite for enabling the OTP forgotpassword link on OAM login pages. For information on available services , See [Available Services of the Common Configuration Section](#).

Use the following Rest API command with the relevant `hostname:port` to enable the OTP forgot password link on the default login page in OAM

```
curl -X PUT \

  http://hostname:port/oam/services/rest/access/api/v1/config/
otpforgotpassword/ \
-H 'authorization: Basic d2VibG9naWM6d2VsY29tZTE=' \
-H 'content-type: application/json' \
-d
'{"displayOTPForgotPassworLink":"true","defaultOTPForgotPasswordLink":"
```

```
false", "localToOAMServer": "false", "forgotPasswordURL": "http://hostname:port/otpfp/pages/fp.jsp", "mode": "userselectchallenge"}'
```

The following table describes the values for the `mode` parameter in the Curl command.

Value	Description
email	OTP will be sent to the email configured in the mail field
sms	OTP will be sent to the mobile number configured in the mobile field
userchoose	OTP will be sent by letting the user choose either mail or phone option, without exact values
userselectchallenge	User can see masked values either as mail or mobile and select one of the options

For further information on OTP Forgot Password Rest APIs in OAM refer REST API for OTP Forgot Password in Oracle Access Manager

Adding Rest admin credentials to CSF

Use the following WLST command to add the Rest admin credentials to the Credential Store Framework (CSF). The credential is required for accessing OAM Rest services.

```
cd <MW_HOME>/oracle_common/common/bin
./wlst.sh
connect ()
createCred(map="OAM_CONFIG", key="umsKey", user="Adminusername",
password="password")
```



Note:

The `user` and `password` are the Rest *adminusername* and *password* to connect to the Rest service.

See also,
About Credential Store Framework Keys

Once all the above steps are completed, the default login page also shows the OTP forgot password link that the user can click to change the forgotten password based on a one-time-pin and login to access the protected page thereafter.

24.10 Multiple Password Policies

Multiple Password policies facilitate user logons belonging to different groups in an organization ensuring the organization's security.

Multiple password policies is useful for setting up varied levels of password based complexity protections to users belonging to different groups.

The Multiple Password policies cannot be configured using OAM console in this release. It can be configured using REST API's only. The policy Admin users can invoke these REST API's. See Rest API for Password Policy Management in Oracle Access Manager

Policies can be defined at different granularities and the priority of the password policy determines which password policy gets applied to the user. The different granularities being:

- IDSTORE level
- GROUP level

For example, while resolving the password policy for a user:

- The list of password policies for the idstore that the user belongs to is retrieved.
- The password policies are sorted by priority.
- The applicable password policy with the highest priority is chosen for the user.
- If the user has no specific password policy defined, then the default password policy is applied to that user.

Note: After any configuration change, you should wait for 60 seconds for the changes to reflect.

Resolving which password policy can be applied for the login user is done as part of the **PasswordManagementPlugin** via the **PasswordPolicyManagementModule**.

25

Managing Policies to Protect Resources and Enable SSO

Access Manager Application Domains and policies can be accessed and managed through the Oracle Access Management Console.

The following topics describe how to create and manage policies and identify the resources to be governed by these policies:

- [Prerequisites to Managing Policies and Protecting Resources](#)
- [Introduction to Application Domain and Policy Creation](#)
- [Understanding Application Domain and Policy Management](#)
- [Managing Application Domains Using the Console](#)
- [Adding and Managing Policy Resource Definitions](#)
- [Defining Authentication Policies for Specific Resources](#)
- [Defining Authorization Policies for Specific Resources](#)
- [Configuring Success and Failure URLs for Authorization Policies](#)
- [Introduction to Authorization Policy Rules and Conditions](#)
- [Defining Authorization Policy Conditions](#)
- [Defining Authorization Policy Rules](#)
- [Configuring Policy Ordering](#)
- [Introduction to Policy Responses for SSO](#)
- [Adding and Managing Policy Responses for SSO](#)
- [Validating Authentication and Authorization in an Application Domain](#)
- [Understanding Remote Policy and Application Domain Management](#)
- [Managing Policies and Application Domains Remotely](#)
- [Application and Application-types](#)

25.1 Prerequisites to Managing Policies and Protecting Resources

Before you proceed with policy management and resource protection ensure the system level requirements are met.

Following are the requirements to perform tasks in this chapter:

- OAM Server should be running.
- Users and groups who can access a protected resource should already be created in the User Identity Store associated with Oracle Access Management.

- Policy-enforcement Agents should be registered.
- Shared components for use in any Application Domain should be defined.

 **See Also:**

- [Understanding Application Domain and Policy Management](#)
- [Introduction to Agents and Registration](#)
- [Managing Authentication and Shared Policy Components](#)

25.2 Introduction to Application Domain and Policy Creation

Application domains are the top-level constructs of the 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 protected resources.

Certain shared components are used within each Application Domain. Each Application Domain represents a singular application on a particular host or Administrators can define different Application Domains for resources that 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. Configurable policies allow or deny access to the resources.

 **Note:**

To enhance security, Access Manager, by default, will deny access when a resource is not protected by a policy that explicitly allows access.

Each Access Manager Application Domain contains information regarding:

- **Resource Definitions**
Each resource definition in an Application Domain requires a Resource Type, Host Identifier (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 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, Conditions, Rules, 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 conditions that must be fulfilled for a successful authorization and define responses to be applied after successful authorization.
- **Policy Ordering**
Policy ordering is a new feature in which the administrator manually designates the order in which policies within an application domain will be matched to incoming requests for access to protected resources. Previous versions of Access Manager used the best match algorithm for this purpose.

When a new application is placed behind an existing agent, the Administrator must decide if the application should be protected by a separate (new) Application Domain and policies or an existing Application Domain and policies. This section provides information in the following sections to inform your choice.

- [About Generating Application Domains and Policies Automatically](#)
- [About Managing Application Domains and Policies Remotely](#)
- [Creating or Managing an Application Domain and Policies](#)

25.2.1 About Generating Application Domains and Policies Automatically

When you register a policy-enforcement Agent with Access Manager, you can choose to have the domain and policies generated automatically or decline the automatic generation.

An automatically generated Application Domain is named for the Agent and seeded with default resources and basic policies (authentication and authorization). No Token Issuance Policy is defined, though an empty container is provided.

During Agent registration, it is presumed that the Agent resides 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. Default resources are protected by basic policies until an Administrator adds more resources or modifies or adds policies.

25.2.2 About Managing Application Domains and Policies Remotely

Access Manager provides two modes to manage Application Domains and their policies without registering or modifying the companion agent.

Remote policy and Application Domain management supports only create and update functions. Remote management does not support removing Application Domains or policies. For more information, see "[Understanding Remote Policy and Application Domain Management](#)".

25.2.3 Creating or Managing an Application Domain and Policies

Here is an overview that outlines the procedures that must be performed to manually create or manage an Application Domain and policies.

To create or manage an Application Domain:

1. Get acquainted with the following details:
 - [Understanding Single Sign-On with Access Manager](#)
 - [Managing Authentication and Shared Policy Components](#)
 - [Understanding Application Domain and Policy Management](#)
 - [Understanding Remote Policy and Application Domain Management](#)
2. Perform all prerequisite tasks for this chapter, as described in:
 - [Prerequisites to Managing Policies and Protecting Resources](#)
3. Start a fresh Application Domain (or view an existing one), as described in:
 - [Creating a New Application Domain](#)
 - [Viewing or Editing an Application Domain](#)
 - [Managing Policies and Application Domains Remotely](#)
4. Add resource definitions to your Application Domain as described in:
 - [Adding and Managing Policy Resource Definitions](#)
5. Define your Authentication Policy, as described in:
 - [Creating an Authentication Policy for Specific Resources](#)
 - [Adding and Managing Policy Responses for SSO](#)
6. Define your Authorization Policy, as described in:
 - [Creating an Authorization Policy and Specific Resources](#)
 - [Adding and Managing Policy Responses for SSO](#)
 - [Defining Authorization Policy Conditions](#)
 - [Defining Authorization Policy Rules](#)
7. Define your Token Issuance Policy, as described in:
 - [Adding and Managing Policy Responses for SSO](#)
 - [Defining Authorization Policy Conditions](#)
 - [Defining Authorization Policy Rules](#)
8. Configure SSO settings and policy evaluation caches, as described in:
 - [Configuring Access Manager Settings : "Managing SSO Tokens and IP Validation"](#)
 - [Configuring Access Manager Settings : Managing Run Time Policy Evaluation Caches](#)
9. Validate your policies and configuration, as described in:

- [Configuring Centralized Logout for Sessions Involving OAM WebGates: Validating Global Sign-On and Centralized Logout](#)

25.3 Understanding Application Domain and Policy Management

Whether you create an Application Domain manually or you accept automatic policy generation when registering an Agent, the elements of an Application Domain are the same. All policies and Application Domains are managed using the Oracle Access Management Console.

For details, see the following topics:

- [Application Domain Pages](#)
- [Application Domain Summary Page](#)
- [Application and Application-types](#)
- [Resource Container in an Application Domain](#)
- [Authentication Policy Pages](#)
- [Authorization Policy Pages](#)
- [Token Issuance Policy Pages](#)

25.3.1 Application Domain Pages

Regardless of the method you choose to create an Application Domain, a unique name is required to be used as an identifier. When you click Application Domains, a Search page is displayed. The Create Application Domain button in the upper-right corner enables you to start a fresh domain definition. Otherwise, enter a name (or leave the Name field blank) and click the Search button to list existing Application Domains.

[Figure 25-1](#) is the Application Domains Search page, controls, and the Search Results table with its own tool bar.

Figure 25-1 Application Domains Search Page

Search Application Domains + Create Application Domain

Use the search tool to find an existing Application Domain or click the Create Application Domain button to create a new one.

▲ Search

Name

Search Results

Actions ▼ View ▼

Row	Name	Description
1	Fusion Apps Integration	Policy objects enabling integration with Oracle Fusion Applications
2	IAM Suite	Policy objects enabling OAM Agent to protect deployed IAM Suite applications

25.3.2 Application Domain Summary Page

When you click the name of an Application Domain in the Search results, the Name, an optional description and Policy Ordering configuration are displayed on the Summary tab.

Other information is organized in the following tabs.

- Resources
- Authentication Policies
- Authorization Policies
- Token Issuance Policies
- Administration

Figure 25-2 is a screenshot of a typical Application Domain page. In a generated Application Domain, the Name and Description are populated as shown. When you create an Application Domain manually, the Description is entered by the Administrator.

Figure 25-2 Example Application Domain Summary Page

The screenshot shows the 'IAM Suite' Application Domain page. At the top, there is a breadcrumb 'Access Manager >' and the title 'IAM Suite Application Domain'. Below the title is a description: 'Application Domain provides a logical container for resources or sets of resources, and the associated policies that dictate who can access specific protected resources.' A navigation bar contains tabs for 'Summary', 'Resources', 'Authentication Policies', 'Authorization Policies', 'Token Issuance Policies', and 'Administration'. The 'Summary' tab is active. On the right side, there is an 'Apply' button. The main content area includes a form with the following fields and controls:

- Name:** A text input field containing 'IAM Suite'.
- Description:** A text area containing 'Policy objects enabling OAM Agent to protect deployed IAM Suite applications'.
- Session Idle Timeout (minutes):** A numeric input field with '0' and up/down arrows.
- Allow OAuth Token:** A checkbox that is unchecked.
- Allow Session Impersonation:** A checkbox that is unchecked.
- Enable Policy Ordering:** A checkbox that is unchecked.

25.3.3 Resource Container in an Application Domain

The Resources tab in the Application Domain represents the container for all resource definitions in that domain. When the Resources tab is clicked and displayed, the Search controls are available to help you find specific definitions quickly.

Figure 25-3 illustrates Search controls that you can use to refine your resource definition search. There is also a New Resource button in the upper-right corner. The Search Results table provides key information about each definition found.

Figure 25-3 Search Results for Resources in an Application Domain

IAM Suite Application Domain

Application Domain provides a logical container for resources or sets of resources, and the associated policies that dictate who can access specific protected resources.

Summary **Resources** Authentication Policies Authorization Policies Token Issuance Policies Administration

Use the search tool to find an existing Resource or click the New Resource button to create a new one.

Search

Resource Type: HTTP
Host Identifier:
Resource URL:
Query String:
Authentication Policy:
Authorization Policy:

Search Reset

Search Results

Actions View Create Duplicate Edit Delete Detach

Row	Resource Type	Host Identifier	Resource URL	Query String	Authentication Policy	Authorization Policy
1	HTTP	IAMSuiteAgent	/ucs/**			
2	HTTP	IAMSuiteAgent	/reqsvc/**			
3	HTTP	IAMSuiteAgent	/sts/**			
4	HTTP	IAMSuiteAgent	/oamfed/**			
5	HTTP	IAMSuiteAgent	/oam/serverfed/authn/sc...		LocalAuthnFederationFA...	Protected Resource Policy
6	HTTP	IAMSuiteAgent	/oam/serverfed/authn/sc...		LocalAuthnFederationLD...	Protected Resource Policy
7	HTTP	IAMSuiteAgent	/oam/serverfed/authn/sc...		LocalAuthnFederationBa...	Protected Resource Policy
8	HTTP	IAMSuiteAgent	/oam/serverfed/authn/sc...		LocalAuthnFederationBa...	Protected Resource Policy

The default Resource Type is HTTP; default Resource URL is /**. With HTTP resource definitions you can also search on a query string defined for that resource. The query string can be only the Base URL and can include optional pattern-matching special characters to represent a set of URLs. In this generated domain, the Host Identifier matches the name of the HTTP agent that was registered. Basic information about the policies is also provided.

See Also:

- "Adding and Managing Policy Resource Definitions"

25.3.4 Authentication Policy Pages

The Authentication Policies tab provides access to defined or generated policies with no search controls needed.

When an Administrator creates an Application Domain manually she must also manually create all policies. In a generated Application Domain, two Authentication policies are created automatically, as shown in Figure 25-5:

- Authentication Policy: Protected Resource Policy
- Authentication Policy: Public Resource Policy

Figure 25-4 Authentication Policies Tab

IAM Suite Application Domain

Application Domain provides a logical container for resources or sets of resources, and the associated policies that dictate who can access specific protected resources.

Summary Resources **Authentication Policies** Authorization Policies Token Issuance Policies Administration

Select an existing Authentication Policy from the list or click the Create Authentication Policy button to create a new one.

Actions **View** **Create** Duplicate Edit Delete Detach

Row	Name	Description
1	LocalAuthnFederationFAAuthScheme	Policy Component created for Local Authentication in IdP mode - Do not modify
2	LocalAuthnFederationLDAPScheme	Policy Component created for Local Authentication in IdP mode - Do not modify
3	LocalAuthnFederationBasicFAScheme	Policy Component created for Local Authentication in IdP mode - Do not modify
4	LocalAuthnFederationBasicScheme	Policy Component created for Local Authentication in IdP mode - Do not modify
5	ESSOAuthnPolicy	Authentication policy for ESSO resources

Authentication policies are local, which means that each policy applies only to the resources specified for the policy. Each resource can be protected by only a single authentication policy.

Figure 25-5 shows the Protected Resource Policy and the columns of information displayed automatically on the policy's Resources tab. The Responses tab is available.

Figure 25-5 Authentication Policy Page: Resources and Responses

ESSOAuthnPolicy Authentication Policy Duplicate Apply

Authentication Policy defines the type of verification that must be performed to provide a sufficient level of trust for Access Manager to grant access to the user making the request. A single policy can be defined to protect one or more resources in the Application Domain.

* Name: ESSOAuthnPolicy Success URL:

Description: Authentication policy for ESSO resources Failure URL:

* Authentication Scheme: OAMLDAPPluginAuthnScheme

Resources Responses Advanced Rules

Resources Add Delete			
Resource Type	Host Identifier	Resource URL	Query String
HTTP	IAMSuiteAgent	/oamreauthenticate/**	
HTTP	IAMSuiteAgent	/logonmanager/**	
HTTP	IAMSuiteAgent	/idaas/iam/esso/**	
HTTP	IAMSuiteAgent	/wlm/**	



Note:

Initially, all resources are protected. Success and Failure URLs and Responses must be added manually; no default values are supplied.

A description is provided during automatic generation:

"Policy set during domain creation. Add resources to this policy to protect them."

This generated policy uses the LDAPScheme as the authentication scheme. However, the optional elements of the policy are not yet defined.

Protected Resources are identified on the Resources tab as *HostIdentifier/***.



Note:

Administrators can change the authentication scheme, specify Success and Failure URLs, add other resources, and define SSO Responses.

Public Resource Policy: A second authentication policy is also generated automatically. This policy uses AnonymousScheme as the default scheme for authentication, which allows anyone access.

Initially, this Public Resource Policy does not include or serve any Resources. The Description tells Administrators what is needed:

Policy set during domain creation. Add resources to this policy to allow anyone access.



See Also:

["Introduction to Policy Responses for SSO"](#)

25.3.5 Authorization Policy Pages

The Authorization Policies tab provides access to defined or generated policies with no search controls needed.

In a generated Application Domain, two Authorization policies are created automatically; however, each resource can be protected by only a single authorization policy:

- Protected Resource Policy
- Public Resource Policy

The Authorization Policy tab is shown in [Figure 25-7](#). From this tab, you can select a policy to edit or create a new policy.

Figure 25-6 Authorization Policies Page

IAM Suite Application Domain

Application Domain provides a logical container for resources or sets of resources, and the associated policies that dictate who can access specific protected resources.

Summary Resources Authentication Policies **Authorization Policies** Token Issuance Policies Administration

Select an existing Authorization Policy from the list or click the Create Authorization Policy button to create a new one.

Actions View **+** Create **↺** Duplicate **✎** Edit **✕** Delete **📄** Detach

Row	Name	Description
1	ESSOReauthAuthzpolicy	Protected Authorization Policy for oamreauth
2	ESSOAuthzPolicy	Protected Authorization Policy for ESSO Resources
3	Protected Resource Policy	Protected Authorization Policy for OAMAgent
4	OICAuthorizationPolicy	Protected Authorization Policy for OIC Resources

The Authorization Policy page is shown [Figure 25-7](#). It provides several tabs where you can define the various components of this Authorization policy. Initially, all resources are protected and access is denied. Success and Failure URLs Conditions, Rules, and Responses must be added manually (no default are supplied).

Figure 25-7 Individual Authorization Policy Page

ESSOAuthzPolicy Authorization Policy Duplicate Apply

Authorization policy contains a set of conditions that define whether a user should be permitted or denied access to the resources protected by the policy. Authorization rules and conditions apply to all resources within a specific Authorization policy.

Summary Resources Conditions Rules Responses

* Name: ESSOAuthzPolicy

Description: Protected Authorization Policy for ESSO Resources

Success URL:

Failure URL:

The Authorization Policy Resources tab is shown in [Figure 25-8](#). You use this page to add (or remove) resources for this policy.

Figure 25-8 Individual Authorization Policy Resources tab

ESSOAuthzPolicy Authorization Policy Duplicate Apply

Authorization policy contains a set of conditions that define whether a user should be permitted or denied access to the resources protected by the policy. Authorization rules and conditions apply to all resources within a specific Authorization policy.

Summary Resources Conditions Rules Responses

Resources + Add X Delete					
Resource Type	Host Identifier	Resource URL	Query String	Name Value list	Operations
HTTP	IAMSuiteAgent	/loginmanager/**			All
HTTP	IAMSuiteAgent	/idaas/fam/esso/**			All
HTTP	IAMSuiteAgent	/wlm/**			All

Administrators can also define Conditions, Rules, and Responses for this policy. None are generated automatically.



See Also:

- ["Introduction to Policy Responses for SSO"](#)
- ["Introduction to Authorization Policy Rules and Conditions"](#)

25.3.6 Token Issuance Policy Pages

By default, only a container for Token Issuance Policies is provided in a generated Application Domain. Any Resources, Conditions, Rules, and Responses must be added manually.

Figure 25-9 Token Issuance Policies Page

IAM Suite Application Domain

Application Domain provides a logical container for resources or sets of resources, and the associated policies that dictate who can access specific protected resources.

Summary Resources Authentication Policies Authorization Policies **Token Issuance Policies** Administration

Select a Token Issuance Policy from the list or click the Create Token Issuance Policy button to create a new one.

Actions View Create Duplicate Edit Delete Detach

Row	Name	Description
No data to display.		

25.4 Managing Application Domains Using the Console

Managing an Application Domain involves adding, modifying, or deleting general and resource-related settings and policies.

Each Application Domain must have a unique name that matches the agent name. After entering a name and optional description for the new Application Domain, click Apply to create it. This manual creation makes available the complete series of tabs: Summary, Resources, Authentication Policies, Authorization Policies, Token Issuance Policies.

Note:

If the Application Domain was created using remote registration or while registering an agent, basic policy information is generated with it. For details, see [Understanding Remote Policy and Application Domain Management](#) and [Managing Policies and Application Domains Remotely](#).

This section describes how to create and manage an Application Domain using the Oracle Access Management Console. It includes the following topics:

- [Creating a New Application Domain](#)
- [Searching for an Existing Application Domain](#)
- [Viewing or Editing an Application Domain](#)
- [Deleting an Application Domain and Its Contents](#)

25.4.1 Creating a New Application Domain

Users with valid Administrator credentials can manually create an Application Domain using the Oracle Access Management Console. Alternatively, Application Domains can be

generated automatically during agent registration, as described in [Introduction to Agents and Registration](#) and [Registering and Managing OAM Agents](#). Decide whether you need a new Application Domain or if you can add resources to an existing Application Domain. You can protect multiple applications using the same Agent by manually creating one Application Domain and manually adding resources and policies.

Prerequisites

See [Prerequisites to Managing Policies and Protecting Resources](#) at the beginning of this chapter.

To create a new Application Domain

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, select **Create Application Domain** from the **Create (+)** menu in the Access Manager section.
3. On the Create Application Domains page, add a unique name, an optional description and other details, then click **Apply** and close the Confirmation window.

See [Configuring Policy Ordering](#).

4. View and manage the following containers (tabs) within the Application Domain container:
 - **Resources:** See "[Adding and Managing Policy Resource Definitions](#)".
 - **Authentication Policies:** See "[Defining Authentication Policies for Specific Resources](#)".
 - **Authorization Policies:** See "[Defining Authorization Policies for Specific Resources](#)".

25.4.2 Searching for an Existing Application Domain

Users with valid Administrator credentials can to search for a specific Application Domain.



Note:

This Search operation is case sensitive.

To search for an Application Domain

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Application Domains**.
3. In the page that appears, enter the name of the Application Domain you want to find (or partial name and wild card, *, or leave the field blank to retrieve all domains). For example:

DesiredDomain

4. Click the **Search** button to initiate the search.
5. Choose a name in the Search Results table to perform the desired task. For instance:
 - **Edit:** Click the **Edit** button in the tool bar to display the configuration page and go to "[Viewing or Editing an Application Domain](#)".
 - **Delete:** See "[Deleting an Application Domain and Its Contents](#)" before you perform this task.
 - **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.

25.4.3 Viewing or Editing an Application Domain

Users with valid Administrator credentials can view or modify an Application Domain (including its resources, policies, conditions, and responses) using the Oracle Access Management Console.

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.



See Also:

["Managing Policies and Application Domains Remotely"](#)

To view or modify an Application Domain and its content

1. Locate the desired Application Domain as described in "[Searching for an Existing Application Domain](#)".
2. Click to open each of the following tabs to add, view, modify, or delete specific details:
 - **Resources:** See "[Adding and Managing Policy Resource Definitions](#)".
 - **Authentication Policies:** See "[Defining Authentication Policies for Specific Resources](#)".
 - **Authorization Policies:** See "[Defining Authorization Policies for Specific Resources](#)".

25.4.4 Deleting an Application Domain and Its Contents

Users with valid Administrator credentials can delete an Application Domain (including its resources, policies, conditions, and responses) using the Oracle Access Management Console.

Deleting the Application Domain and its content removes all referenced objects, including the Agent registration. Using this method, if you later need to re-register the same Agent, you can because there are no remaining references to the previous Application Domain and its content.



Note:

During a Delete operation, if the Application Domain contains any policy elements, you are alerted.

Prerequisites

Ensure that resources in the domain to be deleted are placed in another Application Domain for protection.

To delete an Application Domain

1. Locate the desired Application Domain as described in "[Searching for an Existing Application Domain](#)".
2. Ensure that resources in the domain to be deleted are placed in another Application Domain for protection.
3. In the Search Results table, click the Serial Number beside the desired name, and then click the **Delete (x)** button in the tool bar.
4. In the Warning window, click **Delete** (or click **Cancel** to dismiss the window).
5. Check the results table to confirm the Application Domain has been removed.

25.5 Adding and Managing Policy Resource Definitions

Protecting resources requires an Application Domain containing definitions of the specific resources.

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

Once you have defined a resource in this container, you can add it to a policy in the Application Domain. This section provides the following topics:

- [Resources in an Application Domain](#)
- [Searching for a Resource Definition](#)
- [Defining Resources in an Application Domain](#)
- [Viewing, Editing, or Deleting a Resource Definition](#)

25.5.1 Resources in an Application Domain

Each resource must be defined separately in an Application Domain. Within an Application Domain, resource definitions exist as a flat collection of objects. Each resource is defined as a specific type, and the URL prefix that identifies the resource (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.



Note:

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.

Resource Definition Guidelines

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, with definition of specific operations. Non-HTTP resource types are never associated with a host identifier.
7. Resources can be designated as either Protected, Unprotected, or Excluded.
8. Custom resource types are allowed.

Figure 25-10 illustrates the Create Resource page.

Figure 25-10 Create Resource Page in the Application Domain

Create Resource Resource Apply

Use the following screen to define a Resource and the URL prefix that identifies the resource (document or entity) stored on a server. Individual resource URLs need not be unique across domains, but the combination of a resource URL, Query String, and a host identifier must be unique across domains.

* Type HTTP

Description

* Host Identifier

Uri

* Resource URL

Query Name Value list String

Name	Value
No Data to Display	

Operations

* Operations Available

- All
- CONNECT
- OPTIONS
- POST
- PUT

Protection

* Protection Level

Authentication Policy

Authorization Policy

Table 25-1 describes elements that comprise a resource definition.

Table 25-1 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 defined for the resource. The w1_authen resource type is used for Fusion Middleware application scenarios, as described in the <i>Securing Applications with Oracle Platform Security Services</i> . Any custom resource type that has been defined is listed with default resource types when you add a resource definition (or search for resources). See Also: " Resource Type in a Resource Definition ".
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 ".
URI Section	Information will differ depending on the selected Resource Type.
Query Name-Value list	For HTTP resource types only. You can provide a list of Name and Value pairs for use in access policies. See Also: " Query String Name and Value Parameters for Resource Definitions ".

Table 25-1 (Cont.) Resource Definition Elements

Elements	Description
Query String	For HTTP resources, you can provide a query string for literal full query string matching within access policies. See Also: " Literal Query Strings in Resource Definitions ".
Resource URL	The 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: <ul style="list-style-type: none"> • 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 25-2 .
Operations Section	You can define specific allowed operations to customize you own resource definitions. Note: Oracle-provided Resource Types are read-only. Operations associated with Oracle-provided Resource Types need not be defined and cannot be modified. Policies developed and applied to resources of Oracle-provided types apply to all operations.
Operations Available	Identify all HTTP operations that are allowed for this resource definition. Policies developed and applied to this customized resource apply to only the operations you identify. Unless explicitly noted, all of the following possible operations are for HTTP resource types: <ul style="list-style-type: none"> • Connect • Options • Put • Post • Trace • Head • Delete • Connect • Login (wl_authen resource type only) • Issue (TokenServiceRP resource type only) Note: During Agent registration, if no operation is specified for the resource definition itself, then All operations for that resource type are supported. See Also: " Resource Types and Their Use ".
Protection	Using the controls in this section of the Resource Definition, you can identify the desired level of protection for this resource and name the policies to be used.

Table 25-1 (Cont.) Resource Definition Elements

Elements	Description
Protection Level	<p>Choose the appropriate protection level from the following:</p> <ul style="list-style-type: none"> 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, conditions, 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 conditions and responses is irrelevant. Responses, conditions, 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. While allowing access to excluded resources, Webgate does not contact the OAM Server. 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.
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 the one to protect this resource.

After adding the resource, it is grouped under the Resources node of the named Application Domain. When you create policies all defined resources for the domain are listed and you can choose one or more for inclusion in the policy.

For more information about different specifications within a resource definition, see the following topics:

- [Resource Type in a Resource Definition](#)
- [Host Identifier in a Resource Definition](#)
- [Resource URL, Prefixes, and Patterns](#)
- [Query String Name and Value Parameters for Resource Definitions](#)
- [Literal Query Strings in Resource Definitions](#)
- [Run Time Resource Evaluation](#)

25.5.1.1 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. Native Resource Types are read-only cannot be modified or deleted; these include HTTP, TokenserviceRP, and wl_authen.



See Also:

["Resource Types and Their Use"](#)

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.

Table 25-2 shows sample URL values for resources. For more information, see ["Resource URL, Prefixes, and Patterns"](#).

Table 25-2 HTTP Resources Sample URL Values

Resource	Sample URL Values
Directories	<ul style="list-style-type: none"> • /mydirectory • /mydirectory/**
Pages	<ul style="list-style-type: none"> • /mydirectory/projects/index.html • /mydirectory/projects/*.html • /mydirectory/.../*.html
Web applications	<ul style="list-style-type: none"> • /mydirectory/projects/example.exe • /mydirectory/projects/*.exe • /mydirectory/**

25.5.1.2 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, administrators must choose one host identifier for the computer hosting the resource.

To ensure that Access Manager recognizes the URL for a resource, Access Manager must know the various ways used to refer to that resource's host computer.

25.5.1.3 Resource URL, Prefixes, and Patterns

During automated Application Domain generation, a URL prefix is defined under which all resources are protected. The Access Manager policy model does not support a resource prefix. Administrators can create granular URL patterns.

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 Access Manager policy model does not support a resource prefix. In other words, there is no policy inheritance.

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).

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/.../*`.



Note:

There is no policy inheritance in the Access Manager policy model.

URL Patterns, Matching, and Precedence

The granular URL patterns specify the fine-grained portion of a resource's namespace. All matching is case insensitive.

- Supported wildcard matching is provided for the patterns in [Table 25-3](#)

- Sample Resource URLs and their correctness are shown in [Table 25-4](#)

Table 25-3 Supported Wildcards in Resource URL Patterns (Precedence Order)

Pattern	Description	Example
<code>/**</code>	<p>The default. Matches any sequence of zero or more characters that starts with the forward slash character (/).</p> <p>You can use this pattern to protect a path under a specific, named directory.</p> <p>Note: This is not an existing 12c wildcard. In 12c, the <code>/.../*</code> pattern yielded an exclusive match that did not include the root of the level at which the pattern was defined. For example, <code>/foo/.../*</code> matched <code>/foo/bar</code> and the root directory <code>/foo/</code> itself, but it wouldn't match <code>foo/</code> or <code>/foo</code>. 12c had the notion of a prefix (the "root"), and most evaluation occurred after striping off the prefix.</p>	<p><code>./**</code></p> <p>Matches</p> <p><code>/foo/bar</code></p> <p><code>/foo/</code></p>
Literals	The resource's pattern contains no special characters.	
<code>{pattern1,pattern2,...}</code>	<p>Matches one from a set of patterns.</p> <p>The patterns inside the braces can themselves include any other special characters (except braces; sets of patterns cannot be nested).</p>	<ul style="list-style-type: none"> • <code>a{ab,bc}b</code> matches <code>aabb</code> and <code>abcb</code>. • <code>a{x*y,y?x}b</code> matches <code>axyb</code>, <code>axabayb</code>, <code>ayaxb</code>, and so on.
<code>[range or set]</code>	<p>Matches one from a set of characters.</p> <p>A set can be specified as a series of literal characters or as a range of characters. A range of characters is any two characters (including -) with a hyphen (-) between.</p> <p>A range of characters is any two characters (including -) with a hyphen (-) between them.</p> <p>The forward slash character (/) is not a valid character to include in a set.</p> <p>A set of characters will not match / even if a range that includes / is specified.</p>	<ul style="list-style-type: none"> • <code>[nd]</code> matches only <code>n</code> or <code>d</code>. • <code>[m-x]</code> matches any character between <code>m</code> and <code>x</code>, inclusive. • <code>[-b]</code> matches any character between - and <code>b</code> inclusive (except for /; see <code>/usr/pub/ascii</code> for order of punctuation characters). • <code>[abf-n]</code> matches <code>a</code>, <code>b</code>, and any character between <code>f</code> and <code>n</code>, inclusive. • <code>[a-f-n]</code> matches any character between <code>a</code> and <code>f</code> inclusive, -, or <code>n</code>. (The second - is interpreted literally because the <code>f</code> preceding it is already part of a range.)
Single Character Wildcard <code>?</code>	The <code>?</code> (question mark) matches any one character other than <code>/</code> . This is not treated as a query string delimiter.	<code>a?b</code> matches <code>aab</code> and <code>azb</code> but not <code>a/b</code> .
Wildcard <code>*</code>	The <code>*</code> (asterisk) wildcard matches any sequence of zero or more characters. However, the <code>*</code> (asterisk) does not match the forward slash character (/).	<code>a*b</code> matches <code>ab</code> , <code>azb</code> , and <code>azzzzzzb</code> but not <code>a/b</code> .
<code>*</code>	<p>The <code>*</code> (asterisk) can be used only at the lowest, terminating level of the path. It matches zero or more characters.</p> <p>Every character in a URL pattern must match the corresponding character in the URL path exactly.</p> <p>Exceptions:</p> <ul style="list-style-type: none"> • At the end of a pattern, <code>/*</code> matches any sequence of characters from that point forward. • The pattern <code>*.extension</code> matches any file name ending with the named extension. • Does not match <code>/</code>. 	<p>The following URL pattern:</p> <p><code>/.../update.html</code></p> <p>Matches:</p> <p><code>/humanresources/benefits/update.html</code></p> <p><code>/corporate/news/update.html</code></p> <p><code>update.html</code></p> <p>See Also: Table 25-4</p>

Table 25-3 (Cont.) Supported Wildcards in Resource URL Patterns (Precedence Order)

Pattern	Description	Example
/.../ Hierarchy	Matches any sequence of one or more characters that starts and ends with the forward slash character (/).	<ul style="list-style-type: none"> The pattern /.../index.html matches: /index.html/oracle/index.html/oracle/sales/index.htmlindex.html It does not match xyzindex.html or xyz/index.html.
/.../* Host wide	<p>Evaluation descends from the root /. At each directory level, resources matching the highest precedence level are selected as candidates for continued evaluation and then descend to the next level. This continues until resources representing the best match possible, based solely on the path information, is obtained.</p> <p>Host wide; the entirety of the pattern.</p>	<ul style="list-style-type: none"> /oracle/.../*.html matches: /oracle/index.html/oracle/sales/order.html, and so on.
\	<p>The backslash character is used to escape special characters. Any character preceded by a backslash matches itself.</p> <ul style="list-style-type: none"> Escaped special characters need to be ignored if putting the pattern in wildcard buckets Escaped special characters are matched literally to the special characters, if any, within the incoming URL 	<ul style="list-style-type: none"> abc\d only matches abc*d abc\\d only matches abc\d

Table 25-4 illustrates a number of resource definitions within an Application Domain, organized alphabetically according to the Host Identifier and Resource URL. The right-hand column in Table 25-4 declares whether the form is correct or not.

Table 25-4 Sample Resource URLs

Resource URL	Correct Form
/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

25.5.1.4 Query String Name and Value Parameters for Resource Definitions

The Policy Model supports Query String Name and Value parameters in a Resource pattern definition

- Name: A string literal that can contain any characters, including symbols; all characters are treated as literal.
- Value: Can be a string literal with any characters and can contain a wildcard (*) only) to match a sequence of 0 or more characters. Asterisk (*) is treated as a wildcard.

- Amount: There is no limit to the number of name and value pairs in a query string. However, for a single resource there will be only a few pairs.
- Order: Any order can be used for name and value pairs because at run time these might come in any order as part of the query string.

Resource Matching and Precedence: Query String Name and Value Parameters

Access Manager uses an algorithm that locates the least specific match and continues to the most specific possible resource. When you have candidates defined with both a single-query string and query parameters, those with the single string take precedence.

For resources containing parameter lists, the best match is determined as follows:

1. **Path Matching:** Access Manager attempts to match the path of the requested resource. There may be multiple candidates matched, differing by query component and/or operations declared.
2. **Query String Matching:** For matches obtained, Access Manager attempts to match the query string (if present in the requested URL). If candidates are defined with both single query string and query parameters, those with the literal string take precedence. There may be multiple candidates remaining, differing by operation.
3. **Operation Matching:** For matches obtained, attempt to match the requested operation. If there is no exact match present, then check for resources for which no specific operation(s) have been defined. In other words, they apply to any operation defined as part of the resource's type. In either case, this yields a single, best match.

Path Matching: Defined resources are evaluated for potential match, against the requested URL's path component, in the following precedence order:

- Literals (as in, the resource's pattern contains no special characters)
- Choice: $\{pattern1,pattern2,\dots\}$, each of which may itself contain the below special characters and is evaluated, in turn, using this same precedence order
- Range: []
- Single-char wildcard: ?
- Wildcard: *
- Hierarchy: /.../
- Hostwide: /.../* is the entirety of the pattern

Evaluation descends from the root '/'. At each directory level, resource(s) matching with the highest precedence level are selected as candidates for continued evaluation and then descent to the next level occurs. This continues until resource(s) representing the best match possible, based solely on the path information, is obtained.



Note:

All matching in 11g has been, and remains, case insensitive.

Table 25-5 illustrates the matching pattern for each of several requested URLs.

Table 25-5 Pattern Matching for Requested URLs

Requested URL	Matching Pattern
/oam/sales/oam/page8.html	/oam/.../*.html
/oam/Dept1/page8.html	/oam/Dept?/page8.html
/oam/DeptQ/page8.html	/oam/Dept[A-Z]/page[1-8].html
/oam/DeptQ/page8.html	/oam/Dept[A-Z]/page?.html
/oam/saals/foo/aba/zzz/indexp.html	/oam/sa{*,le,l?,a[k-m],[a-f-m]}s/.../{*b,?a}{a,../ii}/.../{index,test}[pa].?tml

Query String Matching:

When you have candidates defined with both single query string and query parameters, those with the single string take precedence. Single query strings are scored using the algorithm already-mentioned.

For resources containing parameter lists, the best match is determined as follows:

- Resources with parameter values without wildcards are given higher order of precedence; the combined length of the parameter names and values is used to determine the best match among the set of such resources.
- As for query string literals, if there are two or more matches with the same combined length, then matching will fail.
- Resources with parameter values containing wildcards are considered next. The total number of wildcards within each resource is used to determine the best match among such resources. If there are two or more matches having same number of wildcards, then the combined length of the parameter names and values determines the best match
- Matching fails if multiple resources contain the same combined length.

Query String Matching Patterns: Second and subsequent patterns use parameter lists:

```
/oam/index.html::a=*d (a single query string)
/oam/index.html::a:b
/oam/index.html::a:b,c:d
/oam/index.html::a:b*
/oam/index.html::a:b*,c:d
/oam/index.html::a:b*,c:d*
/oam/index.html::a:b*,c:*d*
```

Table 25-6 illustrates the matching pattern for each of several requested URLs.

Table 25-6 Query String Matching: Examples

Requested URL	Matching URL Pattern	Matching Query String Pattern
/oam/index.html?a=b&c=d	/oam/index.html	a=*d
/oam/index.html?a=b1&c=d	/oam/index.html	a:b*,c:d

Table 25-6 (Cont.) Query String Matching: Examples

Requested URL	Matching URL Pattern	Matching Query String Pattern
/oam/index.html?a=b1,c=d1	/oam/index.html	a:b*,c:d*
/oam/index.html?a=b1,c=1d1	/oam/index.html	a:b*,c:*d*

Operation Matching Examples: At this point in request processing, there are one or more candidate resources, all of which match the requested URL path and query string components equally. Access Manager now matches the requested operation to one of those candidates: a resource defined to protect that operation specifically (as well as other, specific operations). As only a single resource can be defined to protect any given operation, this will give the single best match.



Note:

If this does not give a match, verify that the candidate exists protecting all operations (meaning it has been defined using "All", and protects no specific operations).

Run Time Evaluation: Name value pairs are evaluated at run time as follows:

NAME VALUE

a ab
a a*

Same Resource URL Specified Differently: Resources with the same URL and with the same characters in the query string (although specified differently in the console (one as a key and value and the other as a single string)) are considered different and are allowed. For example, the two following resource patterns are considered as different:

Resource URL: /test.html
Query string: area=*&dept=*

Resource URL: /test.html
Query NAME VALUE
area *
dept *

Resource Matching During Policy Evaluation: The order in which name and value pairs arrive at run time does not matter. As long as all the names and values match the query string, the match is successful. The incoming request can have more name and value parameters than defined and still have a successful match.

Example 1: The following pattern matches the incoming URL if no other pattern is defined with the same URL, query string variables, and the extra query string variable (revenue=1000):

Incoming URL => /test.html?area=emea&dept=engg&revenue=1000


```
resource pattern => /test.html
```

Query String NAME VALUE

```
area emea  
dept engg
```

Example 1: For a resource with the same resource URL and the same query string, with one defined as a single string and the other defined as name and value pairs, the policy evaluation preference matches the literal query string before considering name and value pairs. For instance, in the following example, a) is matched:

```
Runtime Request: URL => /test.html?area=emea&dept=engg
```

Resource Patterns:

a)

```
Resource:/test.html
```

```
Query string: area=emea&dept=engg
```

b)

```
Resource:/test.html
```

Query String NAME VALUE

```
area emea  
dept engg
```

Best Match, Multiple Resources: When you have multiple resources with query string name and value pairs defined, the best resource match is the pattern that matches the most number of query string parameters. When wildcard values are used, this is followed by how closely each parameter value matches.

For example: With the following two query string patterns defined:

Query String NAME VALUE

```
area e*
```

```
dept e*
```

and

```
area em*
```

```
dept en*
```

Run Time Query String Parameters:

```
area emea
```

```
dept engg
```

Result: The second name and value pattern match order is higher.

[Figure 25-11](#) shows the resource definition page. Here, "rev*" is a valid name (the asterisk character is allowed and treated as a literal character, which is equivalent to 12c behavior). The Oracle Access Management Console enables you to add query strings as name and value pairs. You can also add the query string as a literal string. If

If you select a literal query string, then the name and value option is disabled (and vice versa).

Figure 25-11 HTTP Resources, Query String Resource URL Controls

Uri

Resource URL

Query Name Value list String

Name	Value
region	
dept	rev*

25.5.1.5 Literal Query Strings in Resource Definitions

The Policy Model supports resource protection based on matching literal, full query-string-based HTTP resource definitions within Access Policies. A single Query String Pattern that would be matched against the entire input Query string (as opposed to matching only portions—selected name and 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, only the 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 the 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.

Remote Registration: 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. Instead, a single authentication scheme is applied to all resources of an application.

25.5.1.6 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:

Other processing details in the following topics:

- ["Resource URL, Prefixes, and Patterns"](#)
- ["Query String Name and Value Parameters for Resource Definitions"](#)
- ["Literal Query Strings in Resource Definitions"](#)
- ["Managing Run Time Policy Evaluation Caches"](#)

Process overview: Resource evaluation

1. A user specifies the URL for a requested resource.
2. Access Manager creates a fully qualified URL that includes the URL pattern, based on the host identifier and URL.
3. Access Manager compares the incoming URL for the requested resource to the fully-qualified URL constructed from 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 the requester is allowed access, the resource is served.

[Table 25-7](#) describes the possible outcomes.

Table 25-7 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 25-7 (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 an 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
```

It does not match, for example, `xyzindex.html`.

- `/oracle/.../*.html` matches:

```
/oracle/index.html
/oracle/sales/order.html
and so on
```

Resource Scope Examples

- Resource scope of the following resource definition (includes the asterisk):

```
/mybank/.../*
```

includes all URLs prefixed with `"/mybank/"`

- Resource scope of the following resource definition (no special characters in the definition):

```
/mybank/account.html
```

includes only one URL: `"/mybank/account.html"`

25.5.2 Defining Resources in an Application Domain

Users with valid Administrator credentials can add the resource definitions to protect to the corresponding Application Domain.

Resource protection based on a list of discrete query parameters is more secure and easier to administer than literal query strings. You might want to create a policy based on resource URL with query parameters (string and name-value pairs).

**Note:**

An error can occur if you specify a host identifier value that is invalid: The challenge URL is invalid.

Prerequisites

The Resource Type must be defined as a Shared Component. Several elements in the Resource definition page are based on the defined and selected Resource Type. For details, see "[Managing Resource Types](#)".



See Also:

"[Resources in an Application Domain](#)"

To add resource definitions to an Application Domain

1. In the Oracle Access Management Console, locate and view the desired Application Domain, as described in "[Searching for an Existing Application Domain](#)".
2. In the Application Domain, click the **Resources** tab, then click the **New Resource** button in the upper-right corner of the Search page.
3. On the **Resource Definition** page:
 - a. Select or enter your details for a single resource ([Table 25-1](#)):
 - Type
 - Description
 - Host Identifier
 - Resource URL ([Table 25-4](#))
 - Operations
 - Query String ([Table 25-6](#))
 - Protection Level
 - Authentication Policy (*if level is Protected*)
 - Authorization Policy (*if level is Protected and Authentication 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.
4. Proceed by adding defined resources to specific policies in the Application domain as described in:
 - [Defining Authentication Policies for Specific Resources](#)
 - [Defining Authorization Policies for Specific Resources](#)

25.5.3 Searching for a Resource Definition

This section provides the following topics:

- [Search Elements and Results for Resource Definitions in an Application Domain](#)
- [Searching for a Specific Resource Definition](#)

25.5.3.1 Search Elements and Results for Resource Definitions in an Application Domain

You can simply click the Search button using the defaults or refine your search by supplying information needed to find the resource.

Figure 25-12 shows the default Search elements and Search Results table for resource definitions in an Application Domain.

Figure 25-12 Resource Search within an Application Domain

Use the search tool to find an existing Resource or click the New Resource button to create a new one.

Search

Resource Type: HTTP (dropdown)
 Host Identifier:
 Resource URL:
 Query String:
 Authentication Policy: (dropdown)
 Authorization Policy: (dropdown)

Search Results

Actions:

Row	Resource Type	Host Identifier	Resource URL	Query String	Authentication Policy	Authorization Policy
1	HTTP	IAMSuiteAgent	/ucs/**			
2	HTTP	IAMSuiteAgent	/reqsvc/**			
3	HTTP	IAMSuiteAgent	/sts/**			
4	HTTP	IAMSuiteAgent	/oamfed/**			
5	HTTP	IAMSuiteAgent	/oam/server/fed/authn/sc...		LocalAuthnFederationFA...	Protected Resource Policy

Table 25-8 lists out the search elements and details.

Table 25-8 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

Table 25-8 (Cont.) Search Elements for a Resource in an Application Domain

Search Elements	Description
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. 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.

25.5.3.2 Searching for a Specific Resource Definition

Users with valid Administrator credentials can search for a specific resource definition.

To find a resource definition:

1. In the Oracle Access Management Console, locate and view the desired Application Domain, as described in "[Searching for an Existing Application Domain](#)".
2. Click the **Resources** tab to display Resources Search controls.
3. Fill in your search criteria ([Table 25-8](#)), and click the **Search** button.
4. In the Search Results table, click the desired resource definition and take the desired action:
 - **Actions Menu:** Select an item to Create, Edit, or Delete the selected resource.
 - **View Menu:** Select an item to alter the appearance of the results table.
 - **Edit Button:** Click the button in the tool bar to display the configuration page.
 - **Delete:** See "[Viewing, Editing, or Deleting a Resource Definition](#)".
 - **Detach:** Click Detach in the tool bar to expand the table to a full page.

25.5.4 Viewing, Editing, or Deleting a Resource Definition

Users with valid Administrator credentials can modify resource definitions within a specific Application Domain.

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.



Note:

During a Delete, you are alerted if the resource is associated with a policy. Without a policy association, the resource is deleted.

Prerequisites

You must have the desired resource type defined as a shared component. For details, see "[Managing Resource Types](#)".



See Also:

["Resources in an Application Domain"](#)

To view, modify, or delete resource definitions

- 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).
 - **Delete:**
 - Open the resource definition and confirm this is the one to be deleted, then close the page.
 - Click the name of the desired resource definition and then click the Delete button in the tool bar.
 - In the Confirmation window, click **Delete** (or click Cancel to dismiss the window). If the Resource is associated with a policy, remove it from the policy first.
 - Repeat as needed to delete other resources in the Application Domain.

25.6 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:

["Understanding Application Domain and Policy Management"](#)

This section provides the following topics:

- [Authentication Policy Page](#)
- [Creating an Authentication Policy for Specific Resources](#)

- [Searching for an Authentication Policy](#)
- [Viewing or Editing an Authentication Policy](#)
- [Deleting an Authentication Policy](#)

25.6.1 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 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.

Authentication Policy Guidelines

1. Authentication policies include resources, success responses, and an authentication scheme.
2. Authentication and Authorization policies can evaluate to Success or Failure.
3. Query Builder and support for LDAP filters (for retrieving matches based on an attribute of a certain display type, for example).
4. Define a policy for resource: `/.../*` which can be used within a determined scope.
5. Token Issuance Policies can be defined using resources and user- or partner-based conditions.

Figure 25-13 shows the Authentication Policies page of an Application Domain.

Figure 25-13 Sample Authentication Policies Page in the Application Domain

IAM Suite Application Domain
 Application Domain provides a logical container for resources or sets of resources, and the associated policies that dictate who can access specific protected resources.

Summary Resources **Authentication Policies** Authorization Policies Token Issuance Policies Administration

Select an existing Authentication Policy from the list or click the Create Authentication Policy button to create a new one.

Actions View **Create** Duplicate Edit Delete Detach

Row	Name	Description
1	LocalAuthnFederationFAAuthScheme	Policy Component created for Local Authentication in IdP mode - Do not modify
2	LocalAuthnFederationLDAPScheme	Policy Component created for Local Authentication in IdP mode - Do not modify
3	LocalAuthnFederationBasicFAScheme	Policy Component created for Local Authentication in IdP mode - Do not modify
4	LocalAuthnFederationBasicScheme	Policy Component created for Local Authentication in IdP mode - Do not modify

Figure 25-14 shows a specific Authentication Policy. The resources assigned to this policy are displayed on the Resources tab of the policy.

Figure 25-14 Sample Individual Authentication Policy Page

ESSOAuthnPolicy Authentication Policy Duplicate Apply

Authentication Policy defines the type of verification that must be performed to provide a sufficient level of trust for Access Manager to grant access to the user making the request. A single policy can be defined to protect one or more resources in the Application Domain.

* Name: Success URL:

Description: Failure URL:

* Authentication Scheme:

Resources Responses Advanced Rules

Resource Type	Host Identifier	Resource URL	Query String
HTTP	IAMSuiteAgent	/oamreauthenticate/**	
HTTP	IAMSuiteAgent	/logonmanager/**	
HTTP	IAMSuiteAgent	/idaas/am/esso/**	
HTTP	IAMSuiteAgent	/wfm/**	

Table 25-9 describes authentication policy elements.

Table 25-9 Authentication Policy Elements and Descriptions

Element	Description
Name	A unique name used as an identifier.
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 " for details.
Success URL	The redirect URL to be used upon successful authentication.
Failure URL	The redirect URL to be used if authentication fails.
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: " Resources in an Authentication Policy ".
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 ".

25.6.1.1 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.

25.6.2 Creating an Authentication Policy for Specific Resources

Users with valid Administrator credentials can 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:

- ["Authentication Policy Page"](#)
- ["Managing Authentication Schemes"](#)

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. Locate the desired domain as described in ["Searching for an Existing Application Domain"](#).
2. Click the **Authentication Policies** tab, then click the **Create Authentication Policy** button to open a fresh page.
3. **Required Elements:** Add your information for this policy.
 - Name
 - Authentication Scheme
4. **Optional Elements** ([Table 25-9](#)): Add as needed for your policy.
 - Description (optional)
 - Success URL
 - Failure URL

Tip:

See [Configuring Success and Failure URLs for Authorization Policies](#).

5. **Add Resources:** A Resource must be defined within the Application Domain before you can add the resource to a specific policy.
 - Click the Resources tab on the Authentication Policy page.
 - Click the Add button on the Resources tab.
 - Click the Search button.
 - Click a URL in the Results table, then click Add Selected.
 - Repeat these steps as needed to add more resources.

6. Click **Apply** to save changes and close the Confirmation window.
7. **Responses:** Add policy Responses as described in "[Adding and Managing Policy Responses for SSO](#)".
8. Close the page when you finish.

25.6.3 Searching for an Authentication Policy

Users with valid Administrator credentials can search for a specific authentication policy.

To search for an authentication policy in an Application Domain

1. Locate the desired domain as described in "[Searching for an Existing Application Domain](#)".
2. Click the **Authorization Policies** tab and:
 - **Edit:** See "[Viewing or Editing an Authentication Policy](#)".
 - **Delete:** "[Deleting an Authentication Policy](#)".
 - **Detach Table:** Click **Detach** in the tool bar to expand the table to a full page.
 - **View Menu:** Select a menu item to alter the appearance of the results table.

25.6.4 Viewing or Editing an Authentication Policy

Users with valid Administrator credentials can 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:

["Authentication Policy Page"](#)

To view or modify an authentication policy

1. Locate the desired policy as described in "[Searching for an Authentication Policy](#)".
2. Click the desired policy name to display its configuration.
3. Edit Policy Elements ([Table 25-9](#)):
4. **Resource:** Click the Resources tab and:
 - **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. **Responses:** View or edit responses as described in "[Adding and Managing Policy Responses for SSO](#)".
7. Close the page when you finish.

25.6.5 Deleting an Authentication Policy

Users with valid Administrator credentials can 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.



Note:

During a Delete operation, you are alerted to confirm removal of the policy. Confirmation is required to complete the operation.

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](#)".



See Also:

["Authentication Policy Page"](#)

To delete an authentication policy

1. Locate the desired policy as described in "[Searching for an Authentication Policy](#)".
2. Click the desired policy name to display and confirm this configuration.
3. Ensure that resources governed by this policy are added to a different policy.
4. Delete all responses, as described in "[Adding and Managing Policy Responses for SSO](#)".
5. On the **Authentication Policies** tab, click the Serial Number beside the policy, then click the Delete button in the tool bar.
6. In the Confirmation window, click **Delete** to confirm (or click Cancel to dismiss the window).

25.7 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:

["Understanding Application Domain and Policy Management"](#)

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:

- [Authorization Policies for Specific Resources](#)
- [Creating an Authorization Policy and Specific Resources](#)
- [Searching for an Authorization Policy](#)
- [Viewing or Editing an Authorization Policy and Resources](#)
- [Deleting an Entire Authorization Policy](#)

25.7.1 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 25-15](#) shows the Authorization Policy page within an Application Domain. The resources assigned to this policy are displayed on the Resources tab of the policy.

Figure 25-15 Sample Individual Authorization Policy Page

Create Authorization Policy Authorization Policy Duplicate Apply

Authorization policy contains a set of conditions that define whether a user should be permitted or denied access to the resources protected by the policy. Authorization rules and conditions apply to all resources within a specific Authorization policy.

Summary Resources Conditions Rules Responses

* Name

Description

Success URL

Failure URL

[Table 25-10](#) describes authorization policy elements. The elements are the same regardless of the domain; only the details will differ.

Table 25-10 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.
Summary	General information (usually Name and optional Description).
Resources	One or more previously-defined resource URLs to be protected by this authorization policy.
Conditions	See Also " Introduction to Authorization Policy Rules and Conditions ".
Rules	See Also " Introduction to Authorization Policy Rules and Conditions ".
Responses	See Also " Introduction to Policy Responses for SSO ".

25.7.2 Creating an Authorization Policy and Specific Resources

Users with valid Administrator credentials can 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:

["Authorization Policies for Specific Resources"](#)

To create an authorization policy and resources

1. Locate the desired domain as described in "[Searching for an Existing Application Domain](#)".
2. Click the **Authorization Policies** tab, then click the **Create** button to open a fresh page.
3. **Summary Tab:** Add your information to the Summary tab ([Table 25-10](#)).
4. **Add Resources:** The Resource must be defined in the Application Domain before you can add the resource to a specific policy.
 - Click the **Resources** tab on the Authorization Policy page.
 - Click the **Add** button on the Resources tab.
 - Click the **Search** button.
 - Click a URL in the Results table, then click **Add Selected**.
 - Repeat these steps to add more resources.

5. Click **Apply** to save changes and close the Confirmation window.
6. **Responses:** Add policy Responses as described in "[Adding and Managing Policy Responses for SSO](#)".
7. **Conditions:** Add authorization conditions, as described in "[Defining Authorization Policy Conditions](#)".
8. **Rules:** Add authorization rules, as described in "[Defining Authorization Policy Rules](#)".
9. Close the page when you finish.

25.7.3 Searching for an Authorization Policy

Users with valid Administrator credentials can locate a specific authorization policy.

To search for an authorization policy

1. Locate the desired domain as described in "[Searching for an Existing Application Domain](#)".
2. Click the **Authorization Policies** tab and:
 - **Edit:** See "[Viewing or Editing an Authorization Policy and Resources](#)".
 - **Delete:** "[Deleting an Entire Authorization Policy](#)".
 - **Detach Table:** Click **Detach** in the tool bar to expand the table to a full page.
 - **View Menu:** Select a menu item to alter the appearance of the results table.

25.7.4 Viewing or Editing an Authorization Policy and Resources

Users with valid Administrator credentials can view or modify an authorization policy within an Application Domain.



See Also:

["Authorization Policies for Specific Resources"](#)

To view or edit an authorization policy

1. Locate the desired domain as described in "[Searching for an Authorization Policy](#)".
2. **Summary:** Edit as needed ([Table 25-10](#)):
3. **Resource:** Click the **Resources** tab and add or delete resources 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 then confirm.
4. Click **Apply** to submit changes and close the Confirmation window (or close the page without applying changes).
5. **Conditions:** See "[Viewing, Editing, or Deleting Authorization Policy Conditions](#)".
6. **Rules:** See "[Defining Authorization Policy Rules](#)".
7. **Responses:** See "[Viewing, Editing, or Deleting a Policy Response for SSO](#)".

8. Close the page when you finish.

25.7.5 Deleting an Entire Authorization Policy

Users with valid Administrator credentials can delete an authorization policy or simply delete resources within the policy.



Note:

During a Delete operation, you are alerted to confirm removal of the policy. Confirmation is required to complete the operation.

When you remove the entire policy, all resource definitions remain within the Application Domain. However, the authorization policy and the conditions and rules governing access are eliminated.

To simply alter an element in the policy see "[Viewing or Editing an Authentication Policy](#)".



See Also:

"[Authorization Policies for Specific Resources](#)"

Prerequisites

Assign resources governed by this policy to another authorization policy, either before or after deleting the policy.

To delete an authorization policy

1. Locate the desired domain as described in "[Searching for an Authorization Policy](#)".
2. **Optional:** Double-click the policy name to review its content, and then close the page when finished.
3. **Delete:** 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.

25.8 Configuring Success and Failure URLs for Authorization Policies

When an Authorization Success or Failure redirect URL is set, the target URL for which the end user is seeking access should be passed along as a parameter.

The following information has relevance when configuring an Authorization policy Success or Failure URL.

- The original resource location will be URL encoded and added as a value to the `oam_res` query parameter before redirecting to the success or failure URL. The following rules are relevant to building the `oam_res` value; during an authorization call, only the `HostIdentifier` is passed so building the URL with a fully qualified host and port is slightly more involved. Here are two examples.

Using the `HostIdentifier`, we find the first fully qualified host:port entry and construct the URL with it. The rest of the entries are then added as query parameters to the resource URL. For example:

```
HostList = [Host hostName="adc00oyf.us.example.com", port=7777",
Host hostName="11gAgent", port=null",
Host hostName="adc00oyf.us.example.com", port=80"] ,
HostIdentifier = 11gAgent
```

The resource URL built will be:

```
HTTP://adc00oyf.us.example.com:7777/index.html?Host1=adc00oyf.us.example.com:80
```

In this second example:

```
HostList =[Host hostName="adc00oyf.us.example.com", port=7777",
Host hostName="11gAgent", port=null] ,
HostIdentifier = 11gAgent
```

The resource URL built will be:

```
HTTP://adc00oyf.us.example.com:7777/index.html
```

- To send a Hashed value of the resource URL for security reasons, run the `displayAuthZCallBackKey()` WLST command. This will return a Base64 encoded string value of the AES 128 key which is generated. This key can be used by the OAM server and the receiving app. It is stored in the `oam-config.xml`. The entry in `oam-config.xml` is found under `/DeployedComponent/Server/NGAMServer/Profile`.

```
<Setting Name="AuthZCallBack" Type="htf:map">
<Setting Name="AuthZHashKey"
  Type="xsd:string">1E8461DFA32AD746AF28BAAAA9F327327941C14CAC216DCFA9AC17985E09
  7A0DD603EC1DF5C6D9F5C904ED44952A5D5F</Setting>
<Setting Name="AuthZCallBackEnabled" Type="xsd:boolean">true</Setting>
</Setting>
```

Note:

See *Access Manager WLST Commands* for details on the `displayAuthZCallBackKey()` WLST command.

- If WLST in step 2 is enabled, we also send a hashed value of the original resource URL as a value of the `oam_res_hash` query parameter. For example:

```
http://adc00oyf.us.example.com:7001/SampleLoginWAR/pages/MFALogin.jsp?
oam_res=HTTP%3A%2F%2Fadc00oyf.us.example.com%3A0%2Findex.html%3FHost1
%3D11gAgent%3Anull&oam_res_hash=45438D536865B256681D328AA1BFD47D5D4D0039
```

25.9 Introduction to Authorization Policy Rules and Conditions

In Access Manager 11g, each Authorization policy includes a rule that defines whether the policy allows or denies access to resources protected by the policy. The rule references

conditions that define the user or population to be granted or denied access and other considerations for authorization. Authorization rules and conditions apply to all resources within a specific authorization policy.

Evaluation of conditions and rules determines if the authorization policy applies to the incoming request. The appropriate obligations take effect after successful authentication and work in concert with defined authorization rules, conditions, and responses. For each incoming request, the authorization policy determines if there are any conditions that apply. If so, these conditions are evaluated.

This section provides the following topics:

- [About Allow or Deny Rules](#)
- [Authorization Policy Conditions](#)
- [About Classifying Users and Groups for Conditions](#)
- [Guidelines for Authorization Responses Based on Conditions](#)

25.9.1 About Allow or Deny Rules

In an authorization policy, a Rule contains all (or a subset) of conditions defined for the policy. The effect of the Rule determines the effect of the policy. You can set one or more rule effects (outcomes) per policy. However, you can specify only one Rule per outcome.

The following outcomes can be applied to authorization and token issuance policies:

- Allow authorized users access to a protected resource. If Allow 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.
- Deny authorized users access to a protected resource.

You can develop simple rules that rely on a single condition, or use expressions to define more complex rules based on multiple conditions. For more information, see "[Expressions and Expression-Based Policy](#)".

25.9.2 Authorization Policy Conditions

A condition is an element that specifies one or more criteria to be satisfied by the access request. Each authorization policy can contain one or more conditions.

In structure, conditions are similar to constraints (in 11.1.1.3 and 11.1.1.5). However, earlier constraints included Allow and Deny rules that are now specified independently on the Rules tab.

Using different condition types, you can:

- Identify the users or groups of users who are either allowed or denied access (based on the rule) to protected resources.
- Stipulate the range of IP addresses who are either allowed or denied access to protected resources.

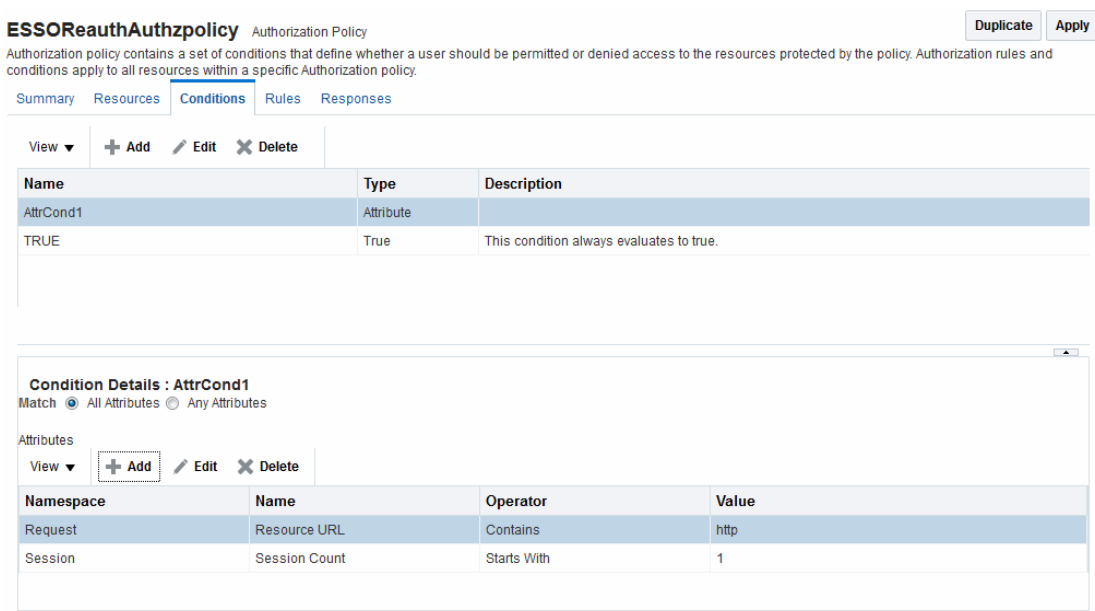
 **Note:**

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 condition applies.
- Specify attributes that enforces evaluation of request context, user session state, and user attributes

The Conditions tab provides a table of defined conditions, organized by name, and a table of details for the selected condition, as shown in [Figure 25-16](#).

Figure 25-16 Individual Authorization Policy Conditions Tab



ESSOReauthAuthzpolicy Authorization Policy Duplicate Apply

Authorization policy contains a set of conditions that define whether a user should be permitted or denied access to the resources protected by the policy. Authorization rules and conditions apply to all resources within a specific Authorization policy.

Summary Resources **Conditions** Rules Responses

View ▾ + Add Edit Delete

Name	Type	Description
AttrCond1	Attribute	
TRUE	True	This condition always evaluates to true.

Condition Details : AttrCond1

Match All Attributes Any Attributes

Attributes

View ▾ + Add Edit Delete

Namespace	Name	Operator	Value
Request	Resource URL	Contains	http
Session	Session Count	Starts With	1

[Table 25-11](#) describes elements and controls on the Conditions tab.

Table 25-11 Authorization Policy Condition Tab

Element	Description
Conditions Table Elements	Lists all conditions defined for this policy.
Name	A unique name used as an identifier for the condition.

Table 25-11 (Cont.) Authorization Policy Condition Tab

Element	Description
Type	The kind of condition you want to use. Only one Type can be specified: <ul style="list-style-type: none"> • Identity • IP4 Range • Temporal • Attribute • True (see Table 21-5)
Description	Optional unique text that describes this condition.
Condition Details Section	Depending on the Type of the selected condition, the information in this table will differ. For details, see: <ul style="list-style-type: none"> • "About Identity Conditions" • "IP4 Range Condition Types" • "Temporal Conditions" • "Attribute-Type Conditions"

25.9.3 About Classifying Users and Groups for Conditions

Oracle recommends that you consider the same information for the policies and conditions 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 Type) while another might be constrained to a specific group of users (Identity Type).



Note:

Do not be concerned about users who are denied access under any conditions. Users are denied access by default if none of the conditions qualify them for access.

When classifying users Oracle recommends that you divide the users, and groups of users, into groups for whom different conditions apply. For example, conditions 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 conditions.

To create policies for subsets of resources in an Application Domain and protect them with different authorization rules and conditions, consider the same information: who can access the resources protected by this policy and under what conditions you want explicitly to allow or deny access to the resources.

25.9.4 Guidelines for Authorization Responses Based on Conditions

For each condition type, consider the response actions that you want to occur for authorized users.

You might want the system to return user profile information and pass that information to a downstream application. For example:

- If the user is authorized, you might 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 might also want to return information about the user for security purposes.

25.10 Defining Authorization Policy Conditions

The mechanism to add a condition is the same regardless of the type you choose.

You use conditions in an authorization policy to:

- Identify the users by user name, role, or an LDAP filter whose criteria the user must satisfy.
- Stipulate the computers where users can access resources.
- Set a time period when the rule applies.
- Specify attributes that enforces evaluation of request context, user session state, and user attributes

A dialog box pops up where you define the name and type to create the condition container. Afterward you are presented with controls to define the specifics of the condition.

This section is divided as follows:

- [Choosing a Condition Type](#)
- [Defining Identity Conditions](#)
- [Defining IP4 Range Conditions](#)
- [Defining Temporal Conditions](#)
- [Defining Attribute Conditions](#)
- [Viewing, Editing, or Deleting Authorization Policy Conditions](#)

25.10.1 Choosing a Condition Type

This section provides the following topics:

- [Condition Window and Elements](#)
- [Choosing a Condition Type](#)

25.10.1.1 Condition Window and Elements

You can have more than one instance of a given type of condition within a policy.

When an Administrator adds a condition to an authorization policy, a window (Figure 25-17) appears where you enter capture the Name, Type, and optional Description. When submitted, this information is used to create a container for condition details that must be also specified.

Figure 25-17 Add Condition Window

Table 25-12 describes the Add Condition elements.

Table 25-12 Add Condition Window Elements

Element	Description
Name	A unique name for this condition.
Type	Only one Type can be specified: <ul style="list-style-type: none"> • Identity (See "About Identity Conditions") • IP4 Range (See "IP4 Range Condition Types") • Temporal (See "Temporal Conditions") • Attribute (See "Attribute-Type Conditions")
Description	Optional.

After the container is added it is displayed on the Condition tab as shown in Figure 25-18. The Name, Type, and Description are displayed in the Results table at the top of the tab. The lower panel contains the details of the condition

Figure 25-18 Condition Containers on the Authorization Policy Page

Summary Resources Conditions Rules Responses			
View ▼	+ Add	✎ Edit	✕ Delete
Name	Type	Description	
AttrCond1	Attribute		
TRUE	True	This condition always evaluates to true.	

**See Also:**

"[Defining Authorization Policy Conditions](#)" for information and procedures

25.10.1.2 Choosing a Condition Type

Users with valid Administrator credentials can choose a condition class for the authorization policy.

**Note:**

You can have more than one instance of a given class of condition in a policy.

Prerequisites

The Application Domain must exist.

**See Also:**

"[Condition Window and Elements](#) "

To choose a condition class

1. Locate the desired policy as described in "[Searching for an Authorization Policy](#)".
2. Click the policy name to open its configuration.
3. On the individual policy page, click the **Conditions** tab.
4. Click the **Add (+)** button and ([Table 25-12](#)):
 - Name: Enter a unique name.
 - Type list, choose the kind of condition (Identity, for example).
 - Click the Add Selected button.

5. Proceed to one of the following topics to complete your definition:
 - [Defining Identity Conditions](#)
 - [Defining IP4 Range Conditions](#)
 - [Defining Temporal Conditions](#)
 - [Defining Attribute Conditions](#)

25.10.2 Defining Identity Conditions

This section provides all information about Identity Conditions in the following topics:

- [About Identity Conditions](#)
- [Specifying Identity Type Conditions](#)

25.10.2.1 About Identity Conditions

When defining an Identity Condition, you must add one or more members of a user population from one or more User Identity Stores.

You can add the user population as a list of users or groups. Alternatively, you can add LDAP search filters to be used at runtime to identify the user population. LDAP search filters provide a simple way to specify a target identity population without having to reorganize or create new groups in the identity store (directory server). For details see:

- [Identity Conditions and User Populations](#)
- [LDAP Search Filter Support in Identity Conditions](#)
- [LDAP Search Filter Syntax](#)

25.10.2.1.1 Identity Conditions and User Populations

After opening the condition container, any defined user population is displayed. As with the other condition types, the Identity type can be used in conjunction with identity and temporal conditions.

When adding an identity condition, you open the popup menu beside the Add (+) button (labeled 1 in [Figure 25-19](#)), choose to Add Users and Groups or Add Search Filter (2). [Figure 25-19](#) shows the popup menu and the Add Identities window that appears (3). After locating the desired identities, select the desired Identities and click Add Selected (4).

Figure 25-19 Add Identities Window

Add Identities [X]

Search and select users and/or groups to add to the current condition.

Search

Store Name: UserIdentityStore1

Entity Type: Group

Entity Name: Administrators

[Search] [Reset]

View ▾

Entity Name	Entity Type	Store Name	Distinguished Name
Administrators	Group	IDSPROFILE-id...	cn=Administrators,ou=groups,ou=myrealm,dc=base_d...
OMSM_ADMINI...	Group	IDSPROFILE-id...	cn=OMSM_ADMINISTRATORS,ou=groups,ou=myrealm...

Columns Hidden 1

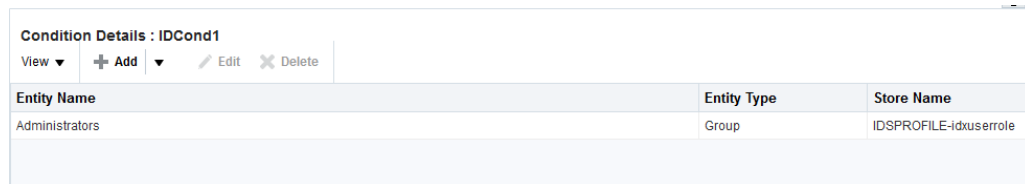
[Add Selected] [Cancel]

Table 25-13 describes the Add Identities elements.

Table 25-13 Add identities Elements

Element	Description
Store Name	Select the desired LDAP store for this search from the list of registered LDAP stores.
Entity type	Choose either Users, Groups, or All to define your search criteria.
Entity Name	Enter information to further refine your search criteria.
Search	Click this button when your search criteria are defined.
Results table	Displays the results of your search.
Add Selected	Click to add the selected users or groups from the results table to the Condition's Details.

After selecting one or more identities and clicking the **Add Selected** button, your Conditions tab might look something like [Figure 25-20](#).

Figure 25-20 Identity Condition and Details

Entity Name	Entity Type	Store Name
Administrators	Group	IDSPROFILE-idxuserrole

To save these details as a condition, click the Save button in the upper-right corner of the tab.

25.10.2.1.2 LDAP Search Filter Support in Identity Conditions

Access Manager 11g authorization conditions accept a list of users, groups, and LDAP search filters as part of allowed or denied identities. An LDAP filter is a text string that expresses specific criteria for the search operation. LDAP search filters provide a simple way to specify a target population without reorganizing or creating new groups in the identity store (directory server).

Access Manager 11g accepts LDAP search filter data for the following conditions and resource types:

- Identity Conditions
- Token Requestor Identity Conditions
- All resource types (HTTP, TokenServiceRP, and other custom resource types)

When a user tries to access a resource protected by a condition containing an LDAP search filter, Access Manager performs a directory lookup (LDAP search) on the identity domain (identity store) specified as a part of the filter. Search results are cached to avoid repeated directory server lookups.

If you choose `Add Search Filter ...`, the controls shown in [Figure 25-21](#) appear. You can add more than one LDAP Search Filter in an authorization rule for evaluation at runtime. The field where you enter your LDAP search filter is used to identify allowed/denied users.

Figure 25-21 Add Search Filter Controls

Add Search Filter

Specify the filter criteria for search operation. The Ldap Search filter is used to identify the allowed or denied users.

* Store Name: UserIdentityStore1

* Search Filter: ((dept=sales)(dept=support))

Test Filter

Test Filter

Test Results

View ▾ Detach

Identifier

No Data to Display

Add Selected Cancel

Table 25-14 describes elements associated with adding a Search Filter.

Table 25-14 Add Search Filter Elements

Element	Description
Domain	The Identity Domain (registered user identity store) in which the search should be conducted during runtime. Each filter must be associated with a specific user identity store. With Access Manager 11g, a directory lookup (LDAP Search) is performed only on the specified identity domain (identity store).

Table 25-14 (Cont.) Add Search Filter Elements

Element	Description
Search Filter	The field where you enter your LDAP search filter. For example: ((dept=sales)(dept=support)) See Also: " LDAP Search Filter Syntax "
Test Filter	This button enables you to test your LDAP Search Filter to ensure it returns the expected result.
Test Results	The results of your filter test are displayed with your own designations for: <ul style="list-style-type: none"> Type: LDAPSearchFilter Identifier: Your LDAP Search Filter
Add Filter	Click to Add the filter to this identity condition.
Cancel	Click to dismiss the Add Search Filter dialog without adding a filter.

[Figure 25-22](#) shows the Identity Conditions: Details page, displayed after adding an LDAP Search Filter.

Figure 25-22 Identity Conditions: Details

Condition Details : IDCond1		
View ▾	+ Add ▾	Edit ✎ Delete ✕
Entity Name	Entity Type	Store Name
((dept=sales)(dept=support))	LDAP Search Filter	UserIdentityStore1
Administrators	Group	IDSPROFILE-idxuserrole

See Also:

- "[LDAP Search Filter Syntax](#)"
- "[Defining Identity Conditions](#)"

25.10.2.1.3 LDAP Search Filter Syntax

Only standard LDAP attributes can be used when defining an LDAP search filter. Exact syntax depends on your identity store; see your vendor documentation.

[Table 25-15](#) illustrates LDAP Search Filter examples for Access Manager.

Table 25-15 LDAP Search Filter Examples for Access Manager

Filter Type and Operators	Description	Syntax Example
Static LDAP Search Filters	<p>When you implement a static search filter, all search results must match a fixed value. For example, you can restrict a search to return only people whose directory profiles show an organizational unit of Sales.</p> <p>As an example of a simple static filter, suppose you want to provide Selector searches for the seeAlso attribute. The filter returns search results that show only people whose directory profiles contain a businessCategory value of dealership.</p>	<p>(attribute=value)</p> <p>For example:</p> <p>(businessCategory=dealership)</p>
Static Searches Using Wild Cards	<p>As an example of a static filter that uses wild cards, suppose you want only people with the word Manager in their title to be returned on a search using the Selector. You can create a filter that searches for the string Manager with the asterisk (*) wildcard.</p>	<p>(attribute=*value*)</p> <p>For example:</p> <p>(title=*manager*)</p>
Dynamic LDAP Search Filters	<p>A dynamic filter allows a search to return results that are based on a user profile. A dynamic filter is a conventional LDAP search filter with filter substitution syntax.</p>	<p>(attribute=\$attribute\$)</p>
Substitution syntax	<p>Substitution syntax is evaluated dynamically, according to the person executing a task. For instance, you can enter substitution syntax where the attribute value for the source DN (the person logged into the application) is substituted and evaluated against the target DN (the entry you are trying to view).</p> <p>Note: Setting a searchbase can present significant administrative overhead. A filter-based approach accomplished by substitution syntax can provide the same functionality in a more scalable and simplified design.</p> <p>Using substitution syntax, you can create a function that starts searches higher in the directory structure, but filters the search data by comparing an attribute of information from the search initiator's record (for example, using the substitution \$ou\$) to an attribute of data on each possible result (for example, ou=).</p> <p>You can use substitution syntax for attribute access control and searchbases. For example, by placing a filter on the type attribute Login for inetOrgPerson, the ability of a user to view any records outside their scope is removed.</p> <p>Note: For the selected searchbase, users can search only for entries from the same ou as their own. This applies only to the attribute on the person's record, not the ou of the branch of the directory in which they reside. Additionally, users from ou=people can search for entries within the selected searchbase.</p>	<p>(attribute=\$attribute\$)</p> <p>For example: The following filter finds all those in the same organizational unit as the person logged in to the application:</p> <p>(ou=\$ou\$)</p>

Table 25-15 (Cont.) LDAP Search Filter Examples for Access Manager

Filter Type and Operators	Description	Syntax Example
Dynamic Searches Using Wild Cards	Wildcards are supported in a dynamic filter. For example, suppose you want to supply a contactPerson attribute in an organizationalUnit object. The contactPerson attribute should return people in same Zip code as the organizationalUnit object. If the organizationalUnit profile contains an attribute zipCode, and the Zip code is specified at the end of a postalAddress directory attribute.	(attribute=*\$attribute\$) For example: (postalAddress=*\$zipCode\$)
Searches Using the Not Operator: (!)	The Not operator is supported when constructing a filter. The optimized algorithm causes the filter (!(sn=white)) to not give the expected result.	((!(sn=white)) (objectclass=personOC))

**See Also:**[Specifying Identity Type Conditions](#)**See Also:**[Upgrading Oracle Internet Directory](#)

25.10.2.2 Specifying Identity Type Conditions

Users with valid Administrator credentials can add identity type conditions to an Application Domain.

**Note:**

You must save each condition definition individually, before adding or selecting another condition.

Prerequisites

The Application Domain must exist.

 **See Also:**

- ["About Identity Conditions"](#)
- ["LDAP Search Filter Support in Identity Conditions"](#)

To add identity conditions to an authorization policy

1. Locate the desired policy as described in ["Searching for an Authorization Policy"](#).
2. Click the **Conditions** tab, click the Add (+) button.
3. Enter a **Name**, select **Identity** from the **Type** list (or Token Requestor Identity) and click **Add Selected**.
4. **Add Users/Groups:**
 - In the Condition Details section click the **Add (+)** button.
 - Choose **Add Users and Groups** from the list.
 - Store Name: Choose the desired name from the list of registered LDAP stores.
 - Enter criteria (Identity Type and Identity Name) for the population you want to find, and click the Search button.
 - Select desired results.
 - Click **Add Selected**.
 - Repeat to add another User or Group condition.
5. **Add Search Filter:**
 - In the Condition Details section click the **Add (+)** button.
 - Domain Name: Choose the desired user identity store for this filter.
 - Search Filter: Enter your search filter syntax ([Table 25-14](#)).
 - Test: Click the Test Filter button and review the results table.
 - Click the **Add Selected** button.
 - Repeat to add another LDAP Search Filter condition.
6. Click **Apply** and then close the Confirmation window.
7. Close the page when you finish.
8. Verify the Conditions by logging in as different users and test access to the resource.

25.10.3 Defining IP4 Range Conditions

This section provides the following information:

- [IP4 Range Condition Types](#)
- [Defining IP4 Range Conditions](#)

25.10.3.1 IP4 Range Condition Types

With the IP4 Range condition type, Administrators can specify a list of IP address ranges that will either be allowed or denied access. Like the other authorization conditions, IP4 Range condition types can be used in conjunction with identity and temporal conditions.

Explicit Addresses: Each IP address you specify must be an explicit, valid address (format *nnn.nnn.nnn.nnn*): *192.2.2.2*, for example.

IP4 Range: You define a range by entering *From* (start) and *To* (end-range) address values. Each IP address you specify must be an explicit, valid address (format *nnn.nnn.nnn.nnn*): *192.2.2.2*, for example. The address specified in the *To* field should be greater than the address specified in the *From* field. During authorization, Access Manager checks to ensure that the client IP address falls between the *From* (start) and *To* (end-range) addresses specified. If multiple overlapping ranges are specified, and the client's IP address falls within even one of the ranges, the condition evaluates to "true" and allows (or denies) access based on the condition that was set for the condition.

If multiple overlapping ranges are specified, and the client's IP address falls within any one of the ranges, the condition evaluates to "true" and allows (or denies) access based on the condition.



Note:

If the *From* IP address is greater than the *To* address, the condition cannot match any client IP address.

Figure 25-23 illustrates the IP4 Range Conditions table with a sample starting and ending IP4 Range. If you enter an invalid range, you are notified and unable to save it.

Figure 25-23 IP4 Range Conditions

The screenshot shows the Oracle Access Manager interface for an authorization policy named "ESSOReauthAuthzpolicy". The "Conditions" tab is active, displaying a table of conditions. The "IPRange1" condition is highlighted, showing a "From" value of "10.0.0.1" and a "To" value of "10.0.0.128". Below the table, the "Condition Details" for "IPRange1" are shown, with the same "From" and "To" values.

Name	Type	Description
AttrCond1	Attribute	
IDCond1	Identity	
IPRange1	IP Range	
TRUE	True	This condition always evaluates to true.

From	To
10.0.0.1	10.0.0.128

25.10.3.2 Defining IP4 Range Conditions

Users with valid Administrator credentials can add IP4 Range type conditions to an Application Domain. You must save each condition definition individually, before adding or selecting another condition.

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.

See Also:

["IP4 Range Condition Types"](#)

To add IP4 Range type conditions to an authorization policy

1. Locate the desired policy as described in ["Searching for an Authorization Policy"](#).
2. Click the **Conditions** tab, click the **Add (+)** button.
3. Enter a **Name**, select **IP Range** from the **Type** list, enter an optional Description, and click **Add Selected**.
4. Add the desired IP address range ([Figure 25-23](#)):
 - In the Details panel, click the **Add (+)** button to display the Add IP Range dialog.
 - **From**: Enter the start of the range.
 - **To**: Enter the end of the range.
 - Click the **Add** button to include this range in the Condition Details section.
 - Repeat these steps to add another range.
5. Click **Apply** and then close the Confirmation window.
6. Verify your IP4 Range Conditions by logging from different clients with different IP addresses to test access to the protected resource.

25.10.4 Defining Temporal Conditions

This section provides the following topics:

- [Temporal Conditions](#)
- [Defining Temporal Conditions](#)

25.10.4.1 Temporal Conditions

With the Temporal condition type, Administrators must add the start and end time and the range of days. Like the other conditions, this one can be used in conjunction with identity and IP4 Range conditions.

By default, all days in the range are enabled (though none are checked in the form as shown in [Figure 25-24](#)).

Figure 25-24 Temporal Condition Type Details Page

The screenshot shows the Oracle Authorization Policy interface. At the top, it displays 'ESSOReauthAuthzpolicy' as the selected policy. Below this, there are tabs for 'Summary', 'Resources', 'Conditions', 'Rules', and 'Responses'. The 'Conditions' tab is active, showing a table of conditions. The 'Temporal1' condition is selected and highlighted in blue. Below the table, the 'Condition Details : Temporal1' form is visible. It includes input fields for 'Start Time' (00:00:00) and 'End Time' (12:00:00). At the bottom, there are checkboxes for the days of the week: Monday, Tuesday, Wednesday, Thursday, Friday (checked), Saturday, and 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 25-16 Temporal Condition Details


Elements	Description
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 condition 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 condition 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 Conditions"


25.10.4.2 Defining Temporal Conditions

Users with valid Administrator credentials can add temporal type conditions to an Application Domain.

 **Note:**
You must save each condition definition individually, before adding or selecting another condition.

Prerequisites

The Application Domain must exist.

 **See Also:**
"Temporal Conditions"

To add temporal conditions to an authorization policy

1. Locate the desired policy as described in "[Searching for an Authorization Policy](#)".
2. Click the Conditions tab, click the **Add (+)** button.
3. Enter a **Name**, select **Temporal** from the **Type** list, enter an optional Description, and click **Add Selected**.
4. In the Details panel ([Table 25-16](#)): Click the condition name in the table to open the details panel:
 - Enter the Start time.
 - Enter the End time.
 - Click the days of the week to which this condition applies (or leave all blank to specify every day of the week).
 - Click **Save**.
5. Click **Apply** and then close the Confirmation window.
6. Verify the Temporal Conditions by logging in at different times to validate access to the protected resource.

25.10.5 Defining Attribute Conditions

This section provides the following topics:

- [Attribute-Type Conditions](#)
- [Defining Attribute Type Conditions](#)

25.10.5.1 Attribute-Type Conditions

An attribute-type condition enforces the evaluation of request context, user session state and user attributes for Allow or Deny access pertaining to all resource types and authorization policies in the Application Domain.

With an attribute-type condition defined, access is based on a list of name-value pairs scoped by the:

- Request context: Information on the requested resource, the client making the request, and the policy that was matched during evaluation.
- Session: User Session details (pre-defined session attributes or a reference to an arbitrary session attribute) when the user has an established session.
- User: User attribute information (reference to a LDAP attribute). This condition is used to define a condition on a reference to a user's arbitrary LDAP attribute only. However, conditions based on userID or groupID are defined using Identity Conditions.

Attribute type conditions are required when access is based on one of the situations described in [Table 25-17](#).

Table 25-17 Access Conditions that Require Attribute-Type Conditions

When Access is based on ...	Description
Session attribute	A user is authorized to access the resource if the session attribute "Authentication level" is xx and Session Attribute "s1" = "v1" and Session Start Time = "xxxx". See: Table 25-20
Requested resource	hostname and port number See: Table 25-19
User details	A user is authorized to access the resource if its "Empno" = "xxxx" (department=sales, for example) See: Table 25-21
Token Issuance based on a session attribute	The Requester Partner can issue a token to the Relying Party if the claim contains an attribute "SessionActiveTime" = "15000". You define claims-based conditions of the Token Issuance policy based on the assertions created using session data.

An Administrator defining attribute type conditions enters data into fields for built-in attributes and known attributes. The attribute name can be entered in a text field or selected from a list of values. The condition to be executed is constructed using "AND" or "OR" conjunctions on the condition. [Figure 25-25](#) illustrates the Attribute Conditions page.

Figure 25-25 Attribute Conditions Page

ESSOReauthAuthzpolicy Authorization Policy Duplicate Apply

Authorization policy contains a set of conditions that define whether a user should be permitted or denied access to the resources protected by the policy. Authorization rules and conditions apply to all resources within a specific Authorization policy.

Summary Resources **Conditions** Rules Responses

View ▾ + Add Edit Delete

Name	Type	Description
AttrCond1	Attribute	
TRUE	True	This condition always evaluates to true.

Condition Details : AttrCond1

Match All Attributes Any Attributes

Attributes

View ▾ + Add Edit Delete

Namespace	Name	Operator	Value
Request	Resource URL	Contains	http
Session	Session Count	Starts With	1

Figure 25-26 shows the Add Attribute Condition dialog box. Each attribute condition is defined by the fields described in Table 25-18.



See Also:

"Defining Attribute Conditions"

Figure 25-26 Add Attribute Condition Dialog

Add Attribute Condition ✕

* Namespace ▾

* Attribute Name ▾

* Operator ▾

* Attribute Value

Table 25-18 Attribute Condition Elements

Condition	Description
Namespace	Supported namespaces: <ul style="list-style-type: none"> Request Built-ins Session Built-ins Session (User Session) User (User Attributes)
Name	Attribute name, which can be added as follows, depending on the: <ul style="list-style-type: none"> Selected from a list if the Namespace is Request (Table 25-19) or Session (Table 25-20) Entered manually into a text field if the Namespace is User
Operator	Allowed operators: <ul style="list-style-type: none"> STARTS WITH EQUALS CONTAINS ENDS WITH
Value	Literal value with no special wildcard characters.

Request Built-ins

[Table 25-19](#) identifies the list of built-in attribute names for Request Built-ins:

Table 25-19 Attribute Names for Request Built-ins

Attribute Name	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.
policy_name	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.

Session Built-ins

[Table 25-20](#) identifies the list of attribute names for Session-based attribute-type conditions.

Table 25-20 Attribute Names for Session Built-ins

Attribute Name	Description
Authentication Level	Current authentication level for the session.

Table 25-20 (Cont.) Attribute Names for Session Built-ins

Attribute Name	Description
Authentication Scheme	Name of the authentication scheme executed to achieve the current authentication level.
Session Count	Session count for the user bound to this session.
Session Creation Time	Session creation time.
Session Expiry Time	Session expiration time.

Example: Attribute Condition Data (Aggregation of Conditions)

[Table 25-21](#) illustrates sample condition data for each allowable namespace.

Table 25-21 Attribute Condition Data (Aggregation of Conditions)

Namespace	Name	Operator	Value
Request-Builtins	Res_host	Equals	7777
Session-Builtins	Authn_level	Equals	2
Session	<i>Sessionattr1</i>	Contains	Foo
User	<i>department</i>	Equals	sales

25.10.5.2 Defining Attribute Type Conditions

Users with valid Administrator credentials can add attribute type conditions to an Application Domain.

 **Note:**

You must save each condition definition individually, before adding or selecting another condition.

Prerequisites

The Application Domain must exist.

 **See Also:**

["Attribute-Type Conditions"](#)

To add attribute type conditions to an authorization policy

1. Locate the desired policy as described in ["Searching for an Authorization Policy"](#).
2. Click the **Conditions** tab, click the **Add (+)** button.

3. Enter a **Name**, select **Attribute** from the **Type** list, enter an optional Description, and click **Add Selected**.
4. **Add Details for Attribute Condition:** Click the name of the condition to expand the details panel, and:
 - Match: Click either All or Any.
 - Namespace: Select from the list (Table 25-18).
 - Name: Select from the list or enter manually (Table 25-19 or Table 25-20).
 - Operator: Select from the list (Table 25-18).
 - Value: Enter manually (Table 25-21).
 - Click **Save**.
 - Repeat as needed.
5. Click **Apply** and then close the Confirmation window.
6. Verify the Attribute Conditions by logging in with different scenarios.

25.10.6 Viewing, Editing, or Deleting Authorization Policy Conditions

Users with valid Administrator credentials can add identity type conditions to an Application Domain.

Prerequisites

The Application Domain and authorization policy exist.



See Also:

["Introduction to Authorization Policy Rules and Conditions"](#)

To view, edit, or delete authorization policy conditions

1. Locate the desired policy as described in ["Searching for an Authorization Policy"](#).
2. Click the **Conditions** tab.
3. **Edit Condition Details:** Click the desired condition, click the Edit button to display the Details panel. Depending on the condition type, perhaps only the Description can be edited.
 - ["Defining Identity Conditions"](#)
 - ["Defining IP4 Range Conditions"](#)
 - ["Defining Temporal Conditions"](#)
 - ["Defining Attribute Conditions"](#)
 - True: Click the name, click the Edit button; only the Description can be edited.
4. **Delete Conditions:** Click the condition to remove and click the Delete button on the Condition tab.
5. Click **Apply** and then close the Confirmation window.
6. Close the page when you finish.

7. Verify the Conditions by accessing the resource and evaluating the results.

25.11 Defining Authorization Policy Rules

When Allow access rules, Deny access rules, or both are specified and do not apply to a user, the user is not qualified by the rule, and is denied access to the requested resource by default.

To specify who is allowed or denied access to the resource, the rule can do the following:

- Identify the users by user name, role, or an LDAP filter whose criteria the user must satisfy.
- Stipulate the computers where users can access resources.
- Set a time period when the rule applies.

This section provides the following topics:

- [Authorization Policy Rules](#)
- [Expressions and Expression-Based Policy](#)
- [Defining Rules in an Authorization Policy](#)

25.11.1 Authorization Policy Rules

Rules are new constructs in the Access Manager 11g policy model. A Rule specifies of how to combine condition evaluation outcomes. Each Rule also contains a rule effect (ALLOW or DENY), which determines the overall policy outcome.

Authorization rules define the actions to take during evaluation of the policy, conditions, and rules as well as what to do based on the outcome. There are three possible outcomes:

- True (Allow access): If the user meets the Allow access condition, the user qualifies for the Allow access part of the rule.
- False (Deny access): If the user meets the Deny access condition, the user qualifies for the Deny access part of the rule.
- Inconclusive: If the user satisfies neither the Allow access nor the Deny access conditions, the rule is said to be unqualified for that user. You can also think of this as the user not qualifying for the rule. If evaluation of a rule results in an unqualified user, the user is denied access to the resource based on that rule.

In some cases, a single authorization rule is all that is required to protect the resources of an Application Domain or a policy. You can configure a rule to identify who is allowed access to the resources it protects, who is denied access to them, and under what conditions these controls apply (for example, when they apply and from which computer). An authorization rule does not need to cover all users in its Allow access and Deny access conditions. Users who request access to a resource that is protected by the rule but do not qualify for any of the conditions are, by default, denied access to the resource.

For other cases, it may be necessary to configure multiple authorization conditions into rules to protect resources. You can impose complex conditions on different users. For example, you can define a rule that includes several authorization conditions, one or more of which a user must meet to qualify for access to a protected resource (or to qualify for denial of access to it). For example, you might require the user to meet two conditions—such as belonging to one group and using a computer assigned a specific IP address—to be granted access to the resource.

Oracle Access Management Console makes it easy for you to form expressions for an authorization rule. Conditions are declared outside of rules and are referenced within rules. Evaluation outcomes are combined in either Simple mode or Expression mode. [Figure 25-27](#) shows the Rules tab in an authorization policy.

Figure 25-27 Authorization Policy Rules Tab: Simple Mode

Summary Resources Conditions **Rules** Responses

* Rule Mode Simple Expression

▲ **Allow Rule**

Match All selected conditions Any of the selected conditions

Available Conditions

Selected Conditions

TRUE (True)

▲ **Deny Rule**

Match All selected conditions Any of the selected conditions

Available Conditions

TRUE (True)

Selected Conditions

[Table 25-22](#) describes the elements and controls on the Rules tab for Simple Mode evaluations.

Table 25-22 Authorization Policy Rules Elements

Element	Description
Rule Mode	<p>The method used for evaluation of conditions and rules:</p> <ul style="list-style-type: none"> Simple: Accepts a list of condition names that are combined using a simple algorithm: ALLOW conditions are combined using logical AND. All Allow conditions must be met to get access. DENY conditions are combined using logical OR. Any Deny condition that is true denies access. DENY always takes precedence over ALLOW. Expression: Accepts a user-specified Boolean expression to combine conditions using condition names, "(", ")", " ", "&" and "!" special characters. Combines conditions into complex policies. See Also: "Expressions and Expression-Based Policy" A policy in which there are one or more conditions that are not part of either Allow rule or Deny rule is treated as a valid policy.
Allow Rule	The rule that allows access based on evaluation of your rules and the Selected Conditions list.
Deny Rule	The rule that denies access based on the evaluation of your rules and the Selected Conditions list.
Match	Criteria you choose to either match All conditions in the Selected Conditions list or Any conditions the Selected Conditions list.
Available Conditions	A list of all defined conditions for this authorization policy.
Selected Conditions	A list of the specific conditions that you build by moving items from the Available Conditions list to this list for use during the policy evaluation process.
Arrow Controls	Controls in the form of arrows enable you to add a condition to the Selected Conditions list (or vice versa to remove a condition from those selected).

25.11.2 Expressions and Expression-Based Policy

When a user requests access to a resource that is protected by an authorization condition and rule, information about the user is checked against the rule. If the condition stipulates other information, such as time period or time of day the condition applies, that, too, is checked. This process is referred to as *evaluation of the rule*.

An authorization expression consists of a single rule or a group of rules combined to express more complex conditions. For example, you can create an expression that requires a user to meet the Allow access conditions of two rules to be granted access to the resource. You use the Oracle Access Management Console to create these expressions, which include the following elements:

- Authorization conditions that you select from those that are defined and available in the authorization policy
- Operators that you use to combine rules to provide the kind of authorization protection that you want ([Table 25-24](#))

For expressions that contain multiple conditions, a user may qualify for none of the expression's conditions, one of the conditions, or for the conditions of multiple rules. In any

case, it is the result of evaluation of the expression—all of its conditions and how they are combined—not any one condition, that determines whether a user is allowed or denied access to a resource.

About the Definitive Result of an Authorization Expression: Access Manager evaluates the rules of an expression until it can produce a definitive result. Evaluation of an authorization expression may produce a definitive Allow access result, a Deny access result, or an Inconclusive result.

Figure 25-28 shows the Rules tab when you use Expression as a Rule Mode.

Figure 25-28 Rules Tab: Expression Rule Mode

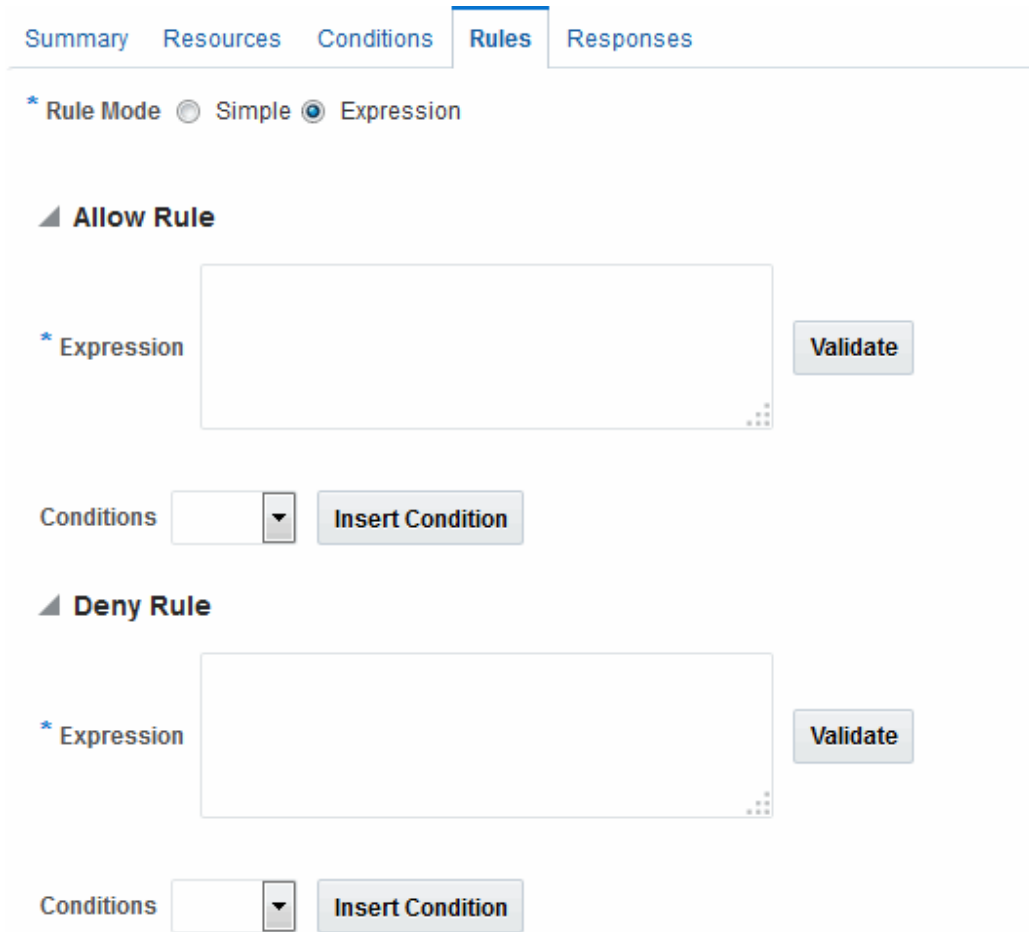


Table 25-23 describes the elements on the Rule tab in Expression mode.

Table 25-23 Rule Tab in Expression Mode

Element	Description
Rule Mode	<p>The method used for evaluation of conditions and rules:</p> <ul style="list-style-type: none"> • Expression: Accepts a user-specified Boolean expression to combine conditions using condition names, "(", ")", " ", "&" and "!" special characters. Combines conditions into complex policies. • A policy in which there are one or more conditions that are not part of either Allow rule or Deny rule is treated as a valid policy. <p>See Also: Table 25-24</p>
Allow Rule	The rule that allows access based on evaluation of your rules and the Selected Conditions list.
Deny Rule	The rule that denies access based on the evaluation of your rules and the Selected Conditions list.
Conditions	Provides a list of all conditions defined for this authorization policy.
Insert Condition	Adds the selected Condition to the expression window.
Validate	Automatically tests the validity of the expression and reports results.

[Table 25-24](#) identifies the operators you can use when building an authorization expression.

Table 25-24 Operators for Expressions in Authorization Rules

Operator	Description
()	<p>By default, two rules on either side of an AND operator compose the compound AND condition. Rules on either side of an OR operator are alternatives. When no parenthesis are used to enforce grouping of rules, the AND operator takes precedence over the OR operator.</p> <p>You can use parenthesis to override the default way in which the rules of an expression are grouped. Evaluation still occurs from left to right, but the rules are organized within the couplings and groups you create through use of parenthesis.</p>
&	<p>The AND operator, which you use to form a compound condition which combines authorization rules. Any number of rules can be combined using the AND operator to implement the full scope of conditions a user must meet to satisfy the authorization requirement. However, a user must satisfy the same kind of condition—either Allow Access or Deny Access—of all of the rules of the AND compound condition for the AND clause to produce a definitive result.</p> <p>An authorization expression can contain more than one coupling or grouping of rules combined using AND. For example, it may contain several AND clauses, one connected to another by an OR operator.</p>

Table 25-24 (Cont.) Operators for Expressions in Authorization Rules

Operator	Description
	<p>The AND operator. An authorization expression can include a complex rule containing two or more alternative authorization conditions. Authorization rules forming a complex condition are combined using the OR operator. Each of the authorization rules specified by a complex OR condition stands on its own. Unlike compound conditions using the AND operator, the user need qualify for the condition of only one of the authorization rules connected by OR operators.</p> <p>An authorization expression can contain as many authorization rules connected using the OR operator as are required to express the authorization policy for the resources it protects. You can use the OR operator to connect authorization rules all of which have Deny Access conditions, all of which have Allow Access conditions, or which specify a mix of Deny Access and Allow Access conditions. You can connect single rules to single rules using OR, and you can connect a single rule to a clause containing rules combined using AND.</p>

25.11.2.1 Expression Evaluation in Authorization Rules

The result of evaluation of an authorization rule, in conjunction with other authorization rules, if more than one is included in the expression, determines if a user is granted access to the requested resource.

Evaluation of the rule occurs as follows:

- Each authorization rule specified in the expression is evaluated from left to right. The outcome is combined progressively with the previously evaluated rules.
- When the evaluation outcome is good enough to decide the overall policy outcome without having to evaluate any more rules, evaluation stops and the overall outcome is returned.
- Each evaluation outcome can be either True, False, or Inconclusive.

Authorization Success: In this case, the user succeeds in gaining access to the requested resource. This result is associated with the Allow Access condition of the expression.

Authorization Failure: In this case, the user fails to gain access to the requested resource. This result is associated with the Deny Access condition of the expression.

Authorization Inconclusive: In this case, the rules of the expression produce conflicting results, and the user is denied access to the resource. If the match for Identity, IP4 address, or timing condition fails then expression evaluation stops and the result of the overall evaluation is deemed Inconclusive. However, based on the other rules present in the expression, this result might not affect the overall policy evaluation.

For example, the following expression:

```
(Rule1 AND Rule 2) OR (Rule 3 AND Rule 4)
```

Yields the following outcomes:

- Rule1 - INCONCLUSIVE
- Rule2 - FALSE

- Rule3 - TRUE
- Rule4 - TRUE
- Overall: TRUE (Allow)

The following sample expression uses (in order of type) Identity, Temporal, IP4Range, and Attribute conditions:

```
(IsEMEAemployee & IsEMEAWorkingHours & !(ConnectedOverVPN |NotReadDisclaimer))
```

Condition names that include spaces, tabs, or special characters (if properly escaped when defining the expression) are properly handled

25.11.3 Defining Rules in an Authorization Policy

Users with valid Administrator credentials can add rules to an authorization policy.

Prerequisites

[Defining Authorization Policy Conditions.](#)



See Also:

["Authorization Policy Rules "](#)

To define authorization policy rules

1. Locate the desired domain as described in ["Searching for an Authorization Policy "](#).
2. Click the **Rules** tab.
3. **Expression:**
 - a. Click **Expression** as the Rule Mode.
 - b. In the Allow Rule Expression field, build your expression by entering operators ([Table 25-24](#)) and choosing and inserting conditions ([Table 25-23](#)).
 - c. Click the **Validate** button to confirm your expression.
 - d. Repeat Steps b and c for the Deny Rule.
 - e. Click Apply.
4. **Simple Rule Mode:**
 - a. Click **Simple** as the Rule Mode.
 - b. **Allow Rule:**

Click to **Match** either:

 - All selected conditions
 - Any of the selected conditions

Using arrows for Allow (or Deny) Rule, move desired conditions from the Available Conditions column into the Selected Conditions column.

Click **Apply**.

- c. Repeat step b for the Deny Rule.
5. Click **Apply** and then close the Confirmation window.
6. Verify the rules by accessing the resource and evaluating the results.

25.12 Configuring Policy Ordering

Previous releases of Access Manager used a policy matching algorithm to match incoming resource URLs with the stored patterns in an Application Domain. A best match is arrived at based on a predefined algorithm. (This algorithm can not be changed.) If multiple patterns are matched with an incoming URL, the best match pattern is selected and its associated policy is evaluated.

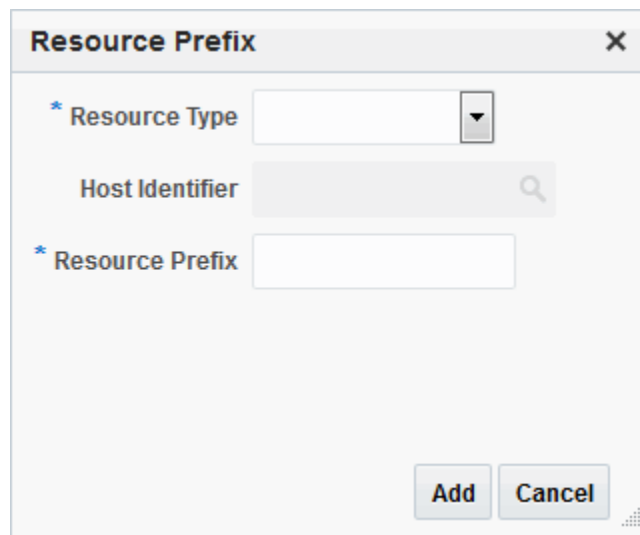
With this 11gR2 PS2 release, rather than the best match algorithm, an Administrator manually designates the order of policies within an Application Domain. To turn on Policy Ordering, the Administrator must first add one or more resource prefixes to the Application Domain. Once these have been added, you can click the Enable Policy Ordering flag. (See [Figure 25-2](#).)

Note:

You may create resource prefixes and not enable policy ordering. In this case, the resource prefixes are ignored and the best match algorithm is used.

[Figure 25-29](#) is a screenshot of the Resource Prefix configuration pop up.

Figure 25-29 Adding a Resource Prefix for Policy Ordering



The screenshot shows a dialog box titled "Resource Prefix" with a close button (X) in the top right corner. The dialog contains three input fields: "Resource Type" (a dropdown menu), "Host Identifier" (a text field with a search icon), and "Resource Prefix" (a text field). The "Resource Type" and "Resource Prefix" fields are marked with an asterisk (*). At the bottom right of the dialog, there are two buttons: "Add" and "Cancel".

During runtime, the incoming URL of the protected resource is checked to determine if it starts with any resource prefix defined in the Application Domain. If the URL matches a resource prefix, the policies in the Application Domain configured with that resource prefix are checked (in the order defined by the Administrator) to see if any resource in

the policy matches the incoming resource. If the incoming resource matches a particular policy, it is evaluated and the results are returned; the other policies are not checked.

To configure Policy Ordering

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, select **Create Application Domain** from the **Create (+)** drop-down menu
3. On the Create Application Domain page, add a unique name and an optional description.
4. Click **Add** to add a Resource Prefix.
5. Tick the **Enable Policy Ordering** box.
6. Select the Resource Type from the drop down list.
See [Table 25-1](#) for definitions of the default Resource Types.
7. Add an optional host identifier.
Host identifier is mandatory for an HTTP Resource Type.
8. Add the Resource Prefix.
For example, if the policy Resource being protected is `/em/**`, the Resource Prefix is `/em`.
If the policy Resource being protected is `/blog/**`, the Resource Prefix is `/blog`.
9. Click **Add**.

25.13 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:

- [Authentication and Authorization Policy Responses for SSO](#)
- [About the Policy Response Language](#)
- [Namespace and Variable Names for Policy Responses](#)
- [About Constructing a Policy Response for SSO](#)
- [About Policy Response Processing](#)
- [Assertion Claims and Processing](#)

25.13.1 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.

There are no default response provided. [Figure 25-30](#) illustrates an Authorization Policy Response defined by an Administrator in the Oracle Access Management Console. Authorization responses can operate in conjunction with authorization conditions.

Figure 25-30 Authorization Policy Response in the Console

The screenshot shows the Oracle Access Management Console interface for an authorization policy named "ESSOReauthAuthzpolicy". The "Responses" tab is selected, showing a table with one response named "REAUTH" of type "Session" and value "TRUE".

Name	Type	Value
REAUTH	Session	TRUE

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 conditions and responses (expressions).

Response Guidelines

1. Cookie, Header, and Session responses are supported.
2. URL redirection can be set.
3. Response definitions are part of each policy. Response values can be literal strings or can contain additional embedded expressions that derive values from request, user, and session attributes.

Administrators set Responses in the Oracle Access Management Console, as described [Table 25-25](#).

Table 25-25 Response Elements

Element	Description
Name	A unique name to distinguish this response from other responses that use the same mechanism (type).
Type	<p>The mechanism used to convey the response. form of the action to be taken with the value string:</p> <ul style="list-style-type: none"> • 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. <p>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.</p> <ul style="list-style-type: none"> • Asserted Attribute: With this type, Identity Assertion must be enabled for the policy to collect Assertion Attribute type responses when this policy is executed. The Name list provides valid identifiers from which to choose.
Value	<p>The response expression, set as a variable.</p> <p>For more information, see "About the Policy Response Language".</p>

25.13.2 About the Policy Response Language

Access Manager 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`

25.13.3 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 25-26](#)
- [Table 25-27](#)
- [Table 25-28](#)

Table 25-26 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_eval_success_conditions	List of policy conditions that evaluated to true, separated by COLON or configured response separator
policy_eval_failure_conditions	List of policy conditions that evaluated to false, separated by COLON or configured response separator
policy_res	Resource host ID and URL pattern matched for the request
policy_name	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 path
res_complete_url	Requested resource URL path with query string

Table 25-27 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 25-28 Namespace User Variables

Namespace	Description
attr.<attrName>	Value of user attribute attrName. If attrName is multivalued, list of values, separated by COLON or configured response separator.
groups	List of user's group membership, separated by COLON or configured response separator.
userid	The user ID

Table 25-28 (Cont.) Namespace User Variables

Namespace	Description
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

25.13.4 About Constructing a Policy Response for SSO

This section is divided as follows:

- [Simple Responses](#)
- [Compound and Complex Responses](#)
- [Multi-Valued Responses](#)



See Also:

[Guidelines for Authorization Responses Based on Conditions](#)

25.13.4.1 Simple Responses

After deciding on the response type and determining which namespace and variable, you simply enter the response attributes in the Oracle Access Management Console.

A simple response might look like one of the several authorization responses shown in [Figure 25-31](#).

Figure 25-31 Simple Response Samples

ESSOReauthAuthzpolicy Authorization Policy Duplicate Apply

Authorization policy contains a set of conditions that define whether a user should be permitted or denied access to the resources protected by the policy. Authorization rules and conditions apply to all resources within a specific Authorization policy.

Summary Resources Conditions Rules **Responses**

Identity Assertion
This will cause an assertion to be generated for the user, optionally containing any Asserted Attribute set below.

Responses + Add Edit Delete

Name	Type	Value
REAUTH	Session	TRUE

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 25-29](#) illustrates several simple responses and a description of what each one returns.

Table 25-29 Simple Responses and Descriptions

Name	Type	Value (Simple \$Namespace.Variable)	Returned Environment Variables and Values
oam_sessioncount	Header	\$session.count	HTTP_OAM_SESSIONCOUNT <i>integer</i>
oam_userid	Header	\$user.userid	HTTP_OAM_USERID <i>name</i>
oam_ipaddress	Header	\$request.client_ip	HTTP_OAM_IPADDRESS <i>nnn.nn.nn.nnn</i>
oam_literal	Header	This is a response string.	HTTP_OAM_LITERAL <i>This is a response string</i>

25.13.4.2 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 25-32 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 25-32 Complex Response Sample

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 25-30 describes the complex responses shown in Figure 25-32.

Table 25-30 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

Table 25-30 (Cont.) Complex Responses

Name	Value	Returned Environment Variables and Values
oam_userinfo	<code>\${user.userid}'s groups: \${user.groups}, description: \${user.attr.description}</code>	HTTP_OAM_USERINFO <i>WebLogic's groups: Administrators, description: This user is the default Administrator</i>
oam_sessioninfo	Session creation/expiration/count: <code>\${session.creation}/\${session.expiration}/\${session.count}</code>	HTTP_OAM_SESSIONINFO Session creation/expiration/count: <i>Tue Oct 23 17:47:42 PST 2011/Wed Oct 24 01:47:42 PST 2011/7</i>
oam_app_user	<code>\$user.userid</code>	HTTP_OAM_USERID <i>name</i>

For more information, see ["About Policy Response Processing"](#).

25.13.4.3 Multi-Valued Responses

Access Manager 11g supports responses with multiple values. These can be multivalued user attribute responses, user's group membership responses and the like. For multivalued responses, Access Manager uses a COLON as the separator and a BACKSLASH as the escape character.

For example, if a user attribute `genType` has the values "Gold", "Platinum" and "Silver", the policy response for `$user.attr.genType` would be:

```
"Gold:Platinum:Silver"
```

If a COLON appears in any of the attribute values, it will be escaped with BACKSLASH. For example, for a user with group memberships as "Administrators", "Special:Users", the policy response for `$user.groups` would be

```
"Administrators:Special\:Users"
```

It is possible to change the default separator and escape character using the `configurePolicyResponses(responseSeparator, responseEscapeChar)` WLST command.

25.13.5 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.

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.

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.

25.13.6 Assertion Claims and Processing

For details, see [Using Identity Context](#) .

25.14 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](#)".

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](#)

25.14.1 Adding a Policy Response for SSO

Users with valid Administrator credentials can add a policy response for authentication or authorization to the Protected Resource Policy.

For example, you can collect the DN of the realm that is created when Oracle Internet Directory is installed. .

Prerequisites

Analyze desired conditions 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](#)

To add a policy Response

1. Locate the desired domain as described in [Searching for an Authorization Policy](#) .
2. In the individual policy page, click the **Responses** tab, then click the **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:

[Namespace and Variable Names for Policy Responses](#)

- Repeat as needed.
3. Click **Apply**, then close the Confirmation window.
 4. Close the page when you finish.
 5. Verify the Responses based on your definitions.

25.14.2 Viewing, Editing, or Deleting a Policy Response for SSO

Users with valid Administrator credentials can 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"](#)

To view, modify, or delete a policy response

1. Locate the desired domain as described in "[Searching for an Existing Application Domain](#)".
2. Click the Authentication (or Authorization) Policies tab, then click the desired policy name.
3. On the individual policy page, click the **Responses** tab and proceed as needed:
 - **Add:** See "[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. Close the Confirmation window.
 5. Close the page when you finish.
 6. Verify Responses based on your definitions for:
 - Header
 - Session
 - Cookie: Use a browser plug-in tool or turn on the browser "show cookies" settings.
 - Assertion Claim

25.15 Validating Authentication and Authorization in an Application Domain

You can validate authentication and authorization by confirming you are redirected to the login page, and after sign-in redirected to the requested resource. The procedure here provides several methods for confirming that Agent registration and authentication and authorization policies are operational.

Prerequisites

- Users and groups who are granted access must exist in the primary LDAP User Identity Store that is registered with Oracle Access Management
- Agents must be registered to operate with Access Manager. 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.
- Logout should be configured as described in [Configuring Centralized Logout for Sessions Involving OAM WebGates](#)

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://exampleWebserverHost.example.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 Management Console.

**See Also:**

[Validating Connectivity and Policies Using the Access Tester](#)

25.16 Understanding Remote Policy and Application Domain Management

Several remote management modes enable Administrators to update, or validate, or delete an existing agent registration.

This section provides the following topics:

- [Remote Policy Management Modes, Templates, and Flags](#)
- [Create Policy Request Template](#)
- [Update Policy Request Template](#)
- [Remote Policy Management Template Elements](#)

25.16.1 Remote Policy Management Modes, Templates, and Flags

Access Manager provides two modes to manage Application Domains and their policies without registering or modifying the companion agent. Remote policy and Application Domain management supports only create and update functions. Remote management does not support removing Application Domains or policies.

**Note:**

Application Domain removal is a manual task that must be performed using the Oracle Access Management Console.

[Table 25-31](#) 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 25-31 Remote Policy Management Modes, Templates, and Flags

Mode and Template	Description
policyCreate \$OAM_REG_HOME/input/ <i>CreatePolicyRequest.xml</i>	Allows Administrators to create Host Identifiers and an Application Domain without registering an Agent. <pre>./bin/oamreg.sh policyCreate input/ myCreatePolicyRequest.xml</pre> See Also: " Create Policy Request Template "
policyUpdate \$OAM_REG_HOME/input/ <i>UpdatePolicyRequest.xml</i>	Allows Administrators to update existing Host Identifiers and Application Domain without updating an Agent. <pre>./bin/oamreg.sh policyUpdate input/ UpdatePolicyRequest.xml</pre> See Also: " Update Policy Request Template "
Flag	Optional
[prompt_flag] value: [-noprrompt]	When the optional -noprrompt flag is used, oamreg can read input from system.in by using echo and pipe to pass data. Examples from \$OAM_REG_HOME location: <pre>(echo username; echo password; echo webgate_password;) ./bin/oamreg.sh inband input/ Request.xml -noprrompt component.oam.conf</pre> <pre>(echo username; echo password; echo webgate_password; echo httpscert_trust_prompt;) ./bin/oamreg.sh inband input/Request.xml -noprrompt</pre> <pre>(echo username; echo password; echo webgate_password; echo cert_password;) ./bin/ oamreg.sh inband input/Request.xml -noprrompt</pre> <pre>(echo username; echo password; echo webgate_password; echo httpscert_trust_prompt; echo cert_password;) ./bin/oamreg.sh inband input/ Request.xml -noprrompt</pre>

Table 25-31 (Cont.) Remote Policy Management Modes, Templates, and Flags

Mode and Template	Description
component.oam.config_file	<p>Optional. Remote registration accepts a configuration file with a URI list as an argument. component.oam.config_file defines the full path to a file containing any number of protected or public URIs. Ensure that the file uses the following syntax and format:</p> <ul style="list-style-type: none"> • At least one protected URI is required • Only one product family is allowed per file • Comments begin with '#' • Keyword 'public_uris': list public URIs on separate lines after this key word. • Keyword 'protected_uris': list URIs to be protected on separate lines after this key word <p>Note: You can configure the authentication scheme for a policy using the following format (the policy name and authentication scheme name must be separated by a Tab character):</p> <pre><Policy Name> 'tab' <Authentication Scheme Name></pre> <p>For example:</p> <pre>##### protected_uris ##### protected policy1 Basic Over LDAP /finance/protected1/** /finance/protected2/** protected policy2 Client Certificate /finance/protected3/*.js,*.png,*.gif ##### public_uris ##### /finance/public /finance/test1/public</pre>

25.16.2 Create Policy Request Template

The `CreatePolicyRequest.xml` file with the remote `policyCreate` mode allows Administrators to create Host Identifiers and an Application Domain 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.

Many of the same parameters are found in the `CreatePolicyRequest.xml` file and the expanded (full) Agent registration templates discussed earlier.

`CreatePolicyRequest.xml` provides elements for Authentication and Authorization Policies and resources (with no `<agentName>` element).

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



See Also:

["Remote Policy Management Template Elements"](#)

25.16.3 Update Policy Request Template

`UpdatePolicyRequest.xml` and `CreatePolicyRequest.xml` are nearly identical. Both provide the same elements, with the exception of the `<protectedAuthnScheme>` element.



See Also:

["Remote Policy Management Template Elements"](#)

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.(with or without query strings).
- 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

25.16.4 Remote Policy Management Template Elements

This topic describes the unique remote management elements for Application Domain management found in the `CreatePolicyRequest.xml` and `UpdatePolicyRequest.xml` files.

These elements are described in [Table 25-32](#).



See Also:

[Table 15-8](#) for a description of elements common to remote registration and remote management.

Table 25-32 Remote Management Template Elements

Element	Description	Example
<code><rregAuthenticationPolicies></code> <code><rregAuthenticationPolicy</code> <code>></code>	Specifies the name and description for the Authentication Policy (to use when creating a new policy or updating an existing policy).	<pre> <rregAuthenticationPolicies> <rregAuthenticationPolicy> <name>AuthenticationPolicy1</name> <description>Authentication policy created using policyUpdate mode of rreg tool</description> . . </rregAuthenticationPolicy> </rregAuthenticationPolicies> </pre>
<code><authnSchemeName></code>	Specifies the Authentication Scheme to use in the Authentication Policy.	<pre> <rregAuthenticationPolicies> . . <authnSchemeName>LDAPScheme </authnSchemeName> . . </rregAuthenticationPolicy> </rregAuthenticationPolicies> </pre>
<code><uriList></code>	Identifies a resource that requires authentication using the policy.	<pre> <rregAuthenticationPolicies> . . <uriList> - <uriResource> <uri>/res1</uri> <queryString /> </uriResource> </uriList> . . </rregAuthenticationPolicy> </rregAuthenticationPolicies> </pre>

Table 25-32 (Cont.) Remote Management Template Elements

Element	Description	Example
<pre><rregAuthorizationPolicies> <rregAuthorizationPolicy></pre>	Specifies the name and description for the Authorization Policy (to use when creating it anew or updating an existing policy).	<pre><rregAuthorizationPolicies> <rregAuthorizationPolicy> <name>AuthorizationPolicy1</name> <description>Authorization policy created using policyUpdate mode of rreg tool</description> . </rregAuthorizationPolicy> </rregAuthorizationPolicies></pre>
<pre><uriList></pre>	Identifies a resource that requires Authorization using the Authorization Policy.	<pre><rregAuthorizationPolicies> . . <uriList> - <uriResource> <uri>/res1</uri> <queryString /> </uriResource> </uriList> . . </rregAuthorizationPolicy> </rregAuthorizationPolicies></pre>

25.17 Managing Policies and Application Domains Remotely

Administrators can create or update existing policies remotely, without revising an agent's registration.

Prerequisites

Review [Remote Policy Management Modes, Templates, and Flags](#)

To managing policies or an Application Domain remotely without an Agent

1. Set up the registration tool as described in, "[Acquiring and Setting Up the Remote Registration Tool](#)".
2. Copy the appropriate request template and develop your own policy-management request (including any Application Domain revisions needed):
 - Create Policy Request File
 - Update Policy Request File
3. On the Agent host, run the following command with the appropriate mode and your own *Request*.xml input file. For example:

policyCreate Mode:

```
./bin/oamreg.sh policyCreate input/myCreatePolicyRequest.xml
```

policyUpdate Mode:

```
./bin/oamreg.sh policyUpdate input/myUpdatePolicyRequest.xml
```

4. Provide the registration Administrator user name and password when asked.
5. Confirm success by reading on-screen messages, then use the Oracle Access Management Console to manage the domain and policies:

```
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/`

25.18 Application and Application-types

Application contains launch-url, icons, description, and other meta-data, and supports SSO agent, federation service provider partner and form-fill applications.

An Application contains:

- launch-url (that will be used by the end user of the application)
- icons, description and other meta-data that is used to display it in the Access Portal (which is user facing)

The following Application types are supported:

- SSO Agent Application (protected by a WebGate)
- Federation Service Provider Partner Application (through Federation, launch a third-party partner application)
- Form-Fill Application (Access Portal Application template based application)

The Application should have the configuration required to SSO enable it. However, for PS2, only Form-Fill application and Federation SP applications have their configuration in the Application. SSO applications have only the launch URL. Additional functionality will be added in subsequent releases.

 **Note:**

Application registration will work only if ESSO is configured and enabled. In order to register an Application, an ESSO IDS Profile must be created because the Application's policy information is stored in the ESSO directory store.

Validating Connectivity and Policies Using the Access Tester

Oracle provides a portable, stand-alone Java application, Access Tester, which 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.

IT professionals and Administrators 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.

The following topics introduce the Access Tester and how to use it in the following sections:

- [Prerequisites to Using the Access Tester to Validate Connectivity and Policies](#)
- [Introduction to the Access Tester for Access Manager 12c](#)
- [Installing and Starting the Access Tester](#)
- [Access Tester Console, Navigation, and Controls](#)
- [Testing Connectivity and Policies from the Access Tester Console](#)
- [Creating and Managing Test Cases and Scripts](#)
- [Evaluating Scripts, Log File, and Statistics](#)

26.1 Prerequisites to Using the Access Tester to Validate Connectivity and Policies

Before you proceed with validation of connectivity and policies ensure the system level requirements are met.

Following are the requirements to perform tasks in this chapter:

- Ensure that the Oracle Access Management Console and OAM Server are running.
- Confirm the Application Domain and policies for one or more resources, as described in [Managing Policies to Protect Resources and Enable SSO](#).

26.2 Introduction to the Access Tester for Access Manager 12c

The Access Tester is a portable, stand-alone Java application that ships with Access Manager 12c. 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 Access Manager.

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 be 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](#)

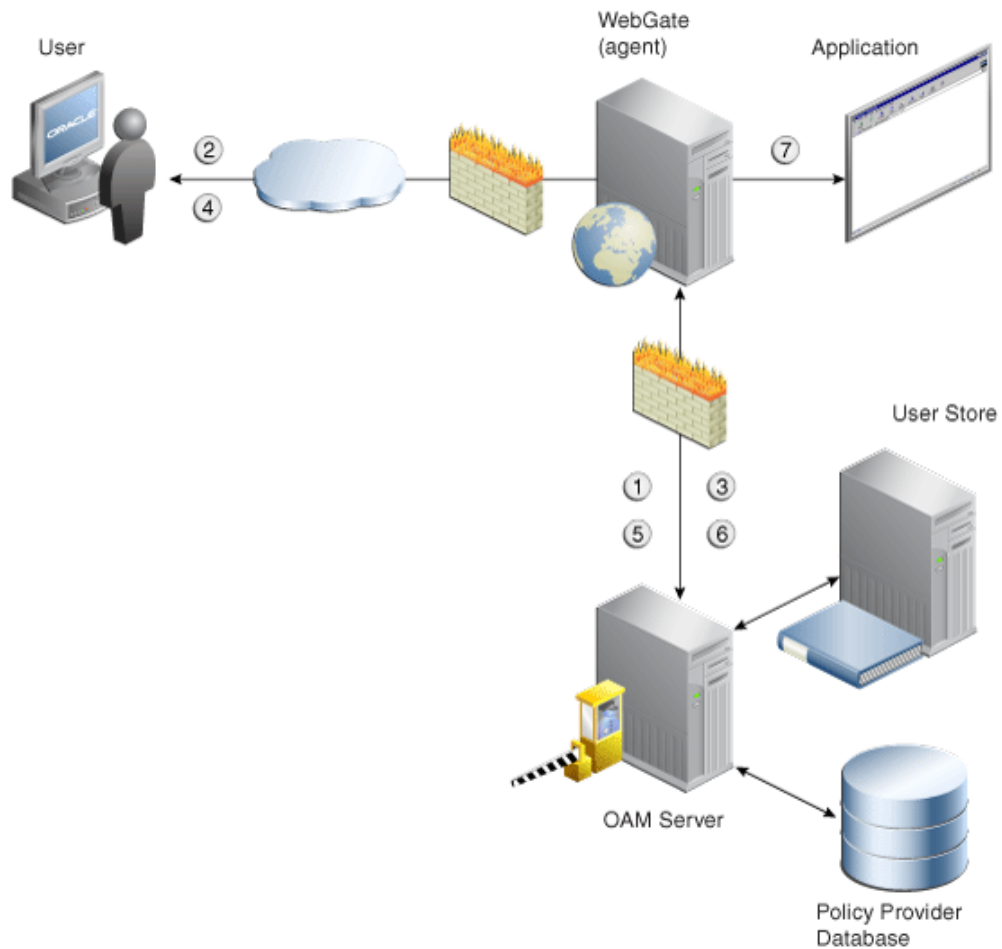
26.2.1 About OAM Agent and Server Interoperability

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 26-1](#) illustrates a typical OAM Agent and OAM Server interaction during a user's request for a resource.

Figure 26-1 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 access 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.

26.2.2 About Access Tester Security and Processing

This topic provides information about secure communications, connections, storage, input, logging, 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 containing 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:

- [Securing Communication](#) for details about Simple and Cert mode configuration for OAM Server and clients (Webgates)
- ["Access Tester Console, Navigation, and Controls"](#)

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 "[Saved Connection Configuration File](#)".
- **Input Script:** script.xml represents a script that is generated by the Access Tester after capturing one or more test cases. For details, see "[Generated Input Test Script](#)".
- **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 "[Target Output File Containing Test Run Results](#)". For example: -
`Dscript.scriptfile="script.xml" -jar oamtest.jar`
- **Statistics:** oamtest_stats.xml is generated together with the output script. For details, see "[Statistics Document](#)".
- **Execution Log:** lamtest_log.log is generated together with the output script. For details, see "[Execution Log](#)".

For more information, see "[About Access Tester Modes and Administrator Interactions](#)".

26.2.3 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 26-2 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 26-2 User Interactions with the Access Tester

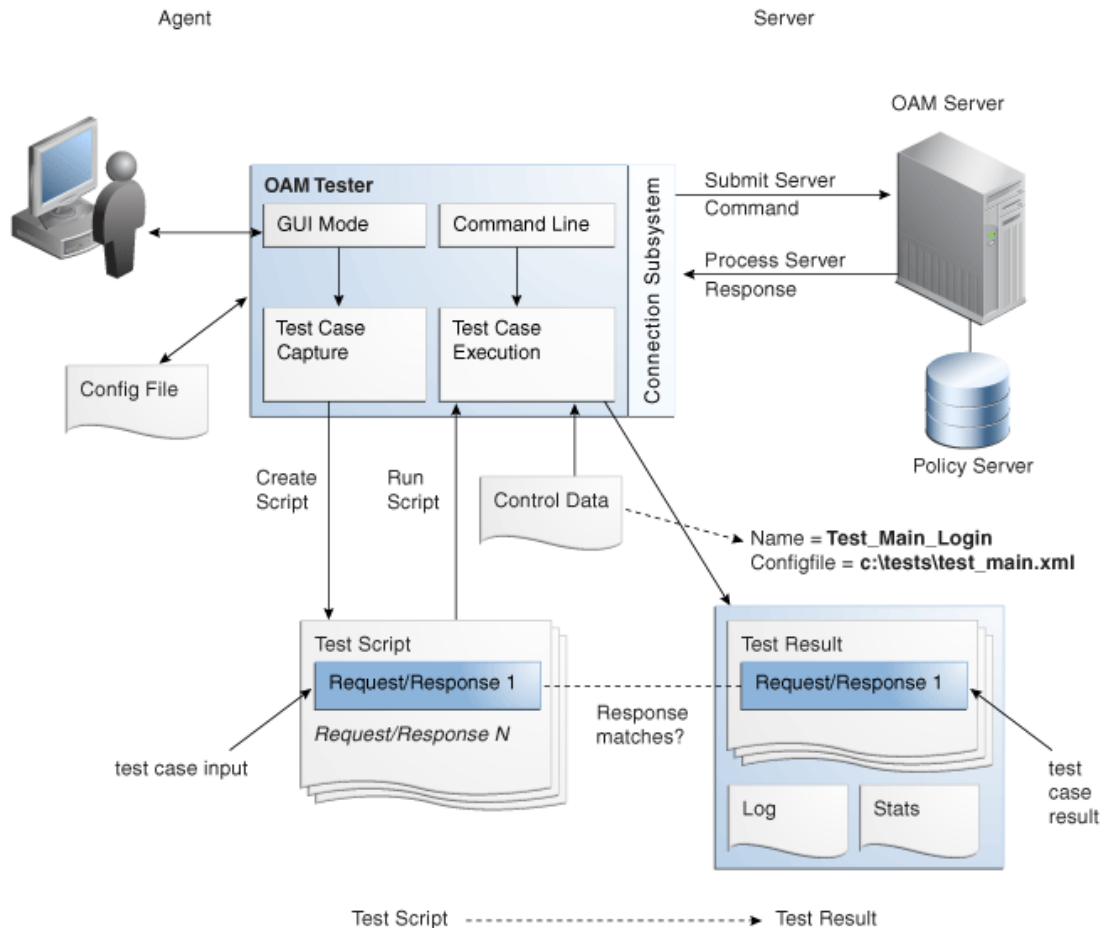


Table 26-1 describes the process flow of information during both Tester Console mode operations and command-line mode operations.

Table 26-1 User Interactions: 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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 Access Manager connections and policies

1. Review the following topics:
 - [Installing and Starting the Access Tester](#)
 - [Access Tester Console, Navigation, and Controls](#)
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](#)"

26.3 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. Installing the Access Tester, 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](#)
- [System Properties Supported by the Access Tester](#)
- [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](#)

26.3.1 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 Management Console. See [Introduction to Agents and Registration](#).
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"

26.3.2 System Properties Supported by the Access Tester

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.

[Table 26-2](#) describes all supported system properties.

Table 26-2 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>"
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"

Table 26-2 (Cont.) Access Tester Supported System Properties

Property	Access Tester Mode	Description and Command Syntax
control.displayiterations tats	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"

26.3.3 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:

- ["System Properties Supported by the Access Tester "](#)
- ["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:

- [Access Tester Console, Navigation, and Controls](#)
- [Testing Connectivity and Policies from the Access Tester Console](#)
- [Creating and Managing Test Cases and Scripts](#)

26.3.4 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](#)

26.3.4.1 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:

["System Properties Supported by the Access Tester "](#)

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.

26.3.4.2 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:

["System Properties Supported by the Access Tester "](#)

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
```

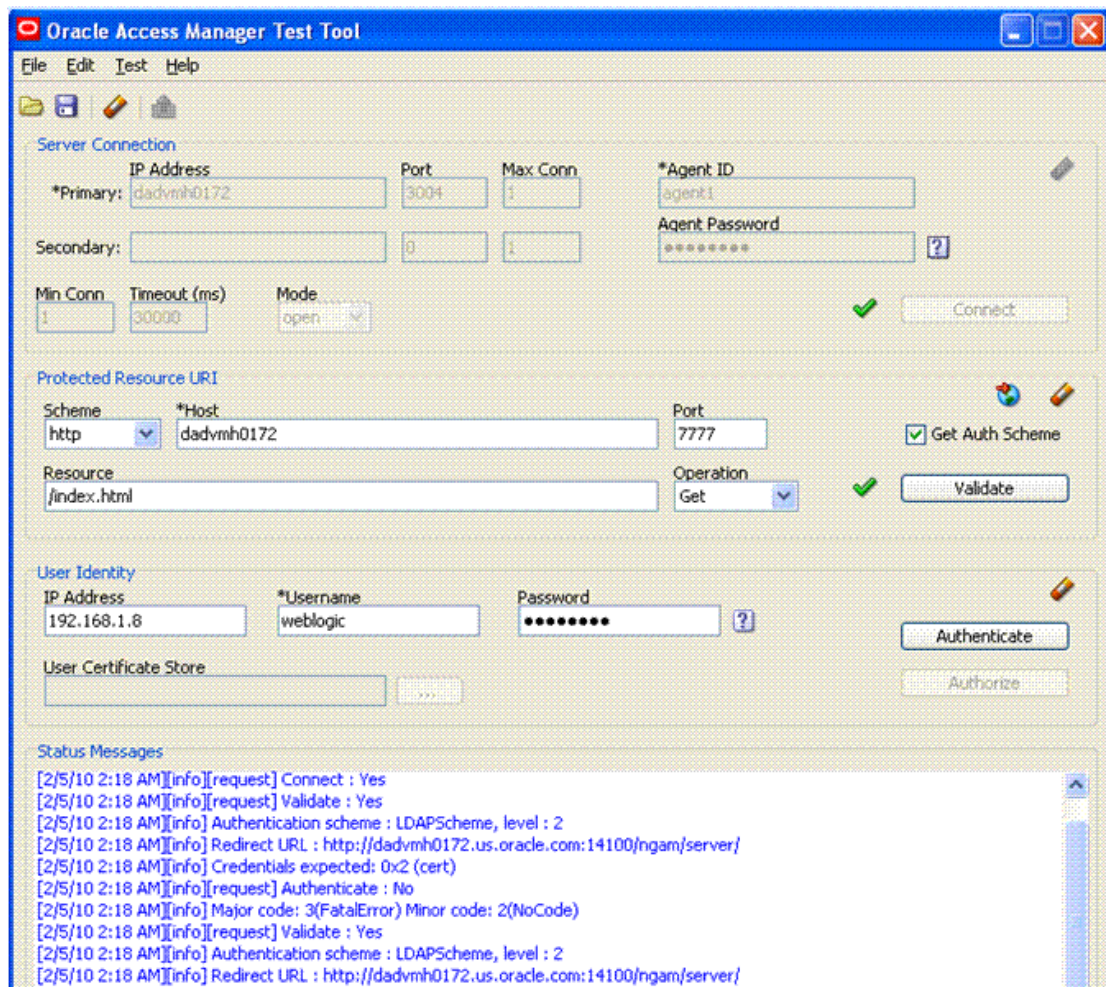
2. After startup, proceed to one of the following topics for more information:
 - [Testing Connectivity and Policies from the Access Tester Console](#)
 - [Creating and Managing Test Cases and Scripts](#)

26.4 Access Tester Console, Navigation, and Controls

Using Access Tester console you can establish a connection to the OAM server, validate protected status of a resource, and confirm authentication and authorization.

Figure 26-3 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 26-3 Access Tester Console



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 26-3](#).

Table 26-3 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 ".
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 ".
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 ".
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 ".

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 26-4](#).

Table 26-4 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](#)"

26.4.1 Access Tester Menus and Command Buttons

Additional Access Tester Console control buttons and command buttons provide a tip when the cursor is on the button.

[Table 26-5](#) identifies additional Access Tester Console buttons and their use.

Table 26-5 Additional Access Tester Buttons

Command Buttons	Description
Open Folder	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.
Disk	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.
Eraser	Clears fields on a panel containing the icon. Tool bar action clears all fields except connection fields if the connection has already been established.
Blue Up Arrows	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.
Paper Scroll	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.
Paper Scroll with right facing arrow	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.
Globe with right facing red arrow	Imports a copied URI from the clipboard after parsing it to populate fields in the URI panel.
Question Mark	Displays a dialog showing the password in clear text

The Access Tester provides the menus described in [Table 26-6](#). 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 26-6 Access Tester Menus

Menu Title	Menu Commands
File	<ul style="list-style-type: none"> • Open Configuration • Save Configuration • Exit <p>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.</p>
Edit	<p>Provides standard editing commands, which act on fields:</p> <ul style="list-style-type: none"> • Cut • Copy • Paste • Clear all fields • Import URI fields from a saved URL
Test	<ul style="list-style-type: none"> • Capture last "... request (for example, Capture last "authorize" request) • Save test script • Run test script <p>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.</p>
Help	The command About, which displays usage information.

26.5 Testing Connectivity and Policies from the Access Tester Console

You can 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"](#).

Task overview: Performing spot checks from the Access Tester Console

1. Start the Access Tester, as described in "Installing and Starting the Access Tester".
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".
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".
4. Add relevant details to the User Identity panel and click Authenticate, as described in "Testing User Authentication from the Access Tester Console".
5. After successful authentication, click Authorize in the User Identity panel, as described in "Testing User Authorization from the Access Tester Console".
6. Check the latency of requests, as described in "Observing Request Latency".

26.5.1 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.

- [Server Connection Panel in the Access Tester](#)
- [Connecting the Access Tester with the OAM Server](#)

26.5.1.1 Server Connection Panel in the Access Tester

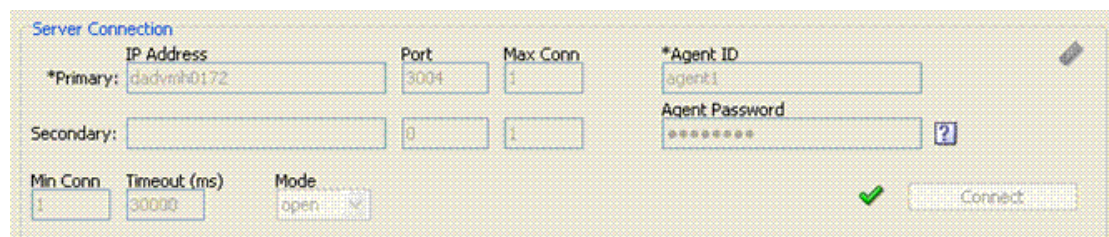
You enter required information for the OAM Server and the Agent you are emulating in the Access Tester Connection panel and then click the Connect button. The Tester initiates the connection, and displays the status in the Status Messages panel. Once the connection is established, it is used for all further operations.

Caution:

Once the connection is established, it cannot be changed until you restart the Access Tester Console.

Figure 26-4 illustrates the Server Connection panel and controls. This panel contains information needed to establish a connection to the OAM Server's Proxy port.

Figure 26-4 Server Connection Panel in the Access Tester



The screenshot shows the 'Server Connection' panel with the following fields and values:

Field	Value
*Primary: IP Address	dadvnh0172
Port	3004
Max Conn	1
*Agent ID	agent1
Secondary: IP Address	
Port	0
Max Conn	1
Agent Password	*****
Min Conn	1
Timeout (ms)	30000
Mode	open

A green checkmark and a 'Connect' button are located at the bottom right of the panel.

Table 26-7 describes the information needed to establish the connection. The source of your values is the Oracle Access Management Console, System Configuration tab.

Table 26-7 Connection Panel Information

Fields	Description
IP Address	<p>The IP Address of the Primary and Secondary OAM Proxy listens on for this set of tests.</p> <p>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.</p>
Port	<p>Enter the port number of the Primary and Secondary OAM Server.</p>
Max Conn	<p>The maximum number of physical connection (TCP) sockets the Access Tester will use. Access Tester emulates a single threaded Agent.</p> <p>Note: Oracle recommends that you accept the default value, 1.</p>
Min Conn	<p>The minimum number of physical connection (TCP) sockets the Access Tester will use. The Access Tester emulates a single threaded Agent.</p> <p>Note: Oracle recommends that you accept the default value, 1.</p>
Timeout	<p>The number of milliseconds the Access Tester should wait for the connection to be established or to receive a response from the OAM Server.</p> <p>Note: Oracle recommends that you accept the default value.</p>
Mode	<p>The level of communication security that is designated for the Agent to be emulated.</p> <ul style="list-style-type: none"> 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". Cert--Presents a Configure Certs ... button that opens a dialog asking for the following: <ul style="list-style-type: none"> 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. <p>See Also: "About Access Tester Security and Processing", and "Generating Client Keystores for OAM Tester in Cert Mode".</p>
Agent ID	<p>Enter the identity of the OAM Agent the Tester is simulating.</p>
Agent Password	<p>Enter the password for the OAM Agent the Tester is simulating, if there is one configured.</p>
Question Mark	<p>Click ? beside the Agent Password field for help.</p>

Table 26-7 (Cont.) Connection Panel Information

Fields	Description
Green Check Mark	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.
X in red circle	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"](#)

26.5.1.2 Connecting the Access Tester with the OAM Server

You can submit your connection details for the OAM Server.

**Note:**

Cert mode requires the presence of keystores generated as described in [Securing Communication](#)

Prerequisites

[Installing and Starting the Access Tester](#)

**See Also:**

["Server Connection Panel in the Access Tester"](#)

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"](#).
2. In the Server Connection Panel ([Table 26-7](#)), 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.

26.5.2 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.

- [Protected Resource URI Panel in the Access Tester](#)
- [Validating Resource Protection](#)

26.5.2.1 Protected Resource URI Panel in the Access Tester

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 26-5](#) 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 26-5 Protected Resource URI Panel in the Access Tester

The screenshot shows a window titled "Protected Resource URI". It contains the following fields and controls:

- Scheme:** A dropdown menu with "http" selected.
- *Host:** A text input field containing "dadvmh0172".
- Port:** A text input field containing "7777".
- Resource:** A text input field containing "/index.html".
- Operation:** A dropdown menu with "Get" selected.
- Get Auth Scheme:** A checkbox that is checked.
- Validate:** A button with a green checkmark icon to its left.

[Table 26-8](#) describes the information needed to perform this validation.

Table 26-8 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 Management 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 Management Console. Note: If protected, the resource identifier that you provide here must match the one specified in an authorization policy in the Oracle Access Management Console.
Globe with red arrow	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.
Green Check Mark	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.
X in red circle	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"](#)

26.5.2.2 Validating Resource Protection

You can 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:**

- ["Protected Resource URI Panel in the Access Tester"](#)

To confirm that a resource is protected

1. In the Access Tester Protected Resource URI panel, enter or import your own resource information ([Table 26-8](#)).
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"](#).
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"](#)

26.5.3 Testing User Authentication from the Access Tester Console

This topic provides the following information:

- [User Identity Panel in the Access Tester](#)
- [Testing User Credential Authentication](#)

26.5.3.1 User Identity Panel in the Access Tester

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 26-6 illustrates the Access Tester panel where you enter the information needed to test authentication.

Figure 26-6 Access Tester User Identity Panel

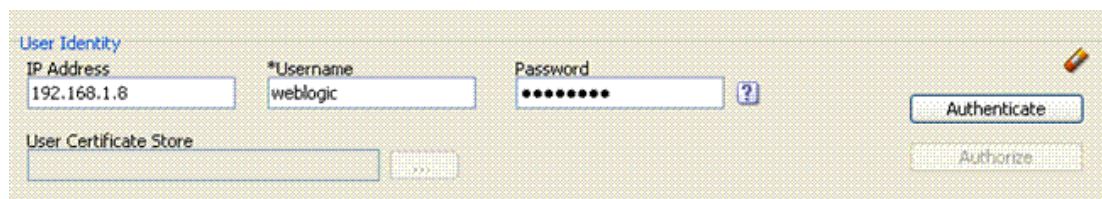


Table 26-9 describes the information you must provide.

Table 26-9 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.

Table 26-9 (Cont.) Access Tester User Identity Panel Fields and Controls

Field or Control	Description
User Certificate Store	<p>The PEM format file containing the X.509 certificate of the user whose credentials should be authenticated.</p> <p>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.</p> <p>Note: For certificate-based authentication to work, the OAM Server must be properly configured with root CA certificates and SSL keystore certificates. See Securing Communication for details about securing communication between OAM Servers and Webgates.</p>
...	Click this button to browse the file system for the user certificate store path.
Authenticate	<p>Click the Authenticate button to submit the request to the OAM Server and look for a response in the Status Messages panel.</p> <p>Note: The type of credentials supplied (username/password or X.509 certificate) must match the requirements of the authentication scheme that protects the URI.</p> <p>Note: For certificate-based authentication, the OAM Server deployment must be properly configured with certificates as described in Securing Communication.</p>
Authorize	<p>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.</p> <p>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.</p>
Green Check Mark	<p>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.</p> <p>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.</p>
X in red circle	<p>A red circle appearing beside the Authenticate button indicates authentication failure; The Status Messages panel also indicates "no" authentication was not successful.</p> <p>A red circle appearing beside the Authorize button indicates authorization failure; The Status Messages panel also indicates "no" authorization was not successful.</p>

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"](#)

26.5.3.2 Testing User Credential Authentication

You can 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:**

["User Identity Panel in the Access Tester"](#)

To test user credential authentication

1. In the Access Tester User Identity panel, enter information for the user to be authenticated ([Table 26-9](#)).
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 Management Console for an active user session that you might need to end, as described in [Maintaining Access Manager Sessions](#).
4. Capture the request and response to create a test case for use later, as described in ["Creating and Managing Test Cases and Scripts"](#).
5. Retain the URI and user identity information and proceed to ["Testing User Authorization from the Access Tester Console"](#).

26.5.4 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. You verify the authenticated end user's authorization for the resource. All relevant authorization conditions 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:

["User Identity Panel in the Access Tester"](#)



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 26-9](#)).
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 Management 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"](#).
6. Proceed to:
 - [Observing Request Latency](#)
 - [Creating and Managing Test Cases and Scripts](#)
 - [Evaluating Scripts, Log File, and Statistics](#)

26.5.5 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"](#)

Task overview: Observing request latency includes

1. ["Validating Resource Protection "](#)
2. ["Testing User Authentication from the Access Tester Console"](#)

3. ["Testing User Authorization from the Access Tester Console"](#)
4. Check latency information in the execution log file as shown here, as well as in other files generated during a test run. For example:

```
...
[2/3/12 11:03 PM][info] Summary statistics
[2/3/12 11:03 PM][info] Matched 4 of 4, avg latency 232ms vs 238ms
[2/3/12 11:03 PM][info] Validate: matched 2 of 2, avg latency 570ms vs 578ms
[2/3/12 11:03 PM][info] Authenticate: matched 1 of 1, avg latency 187ms vs 187ms
[2/3/12 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](#)

26.6 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](#)

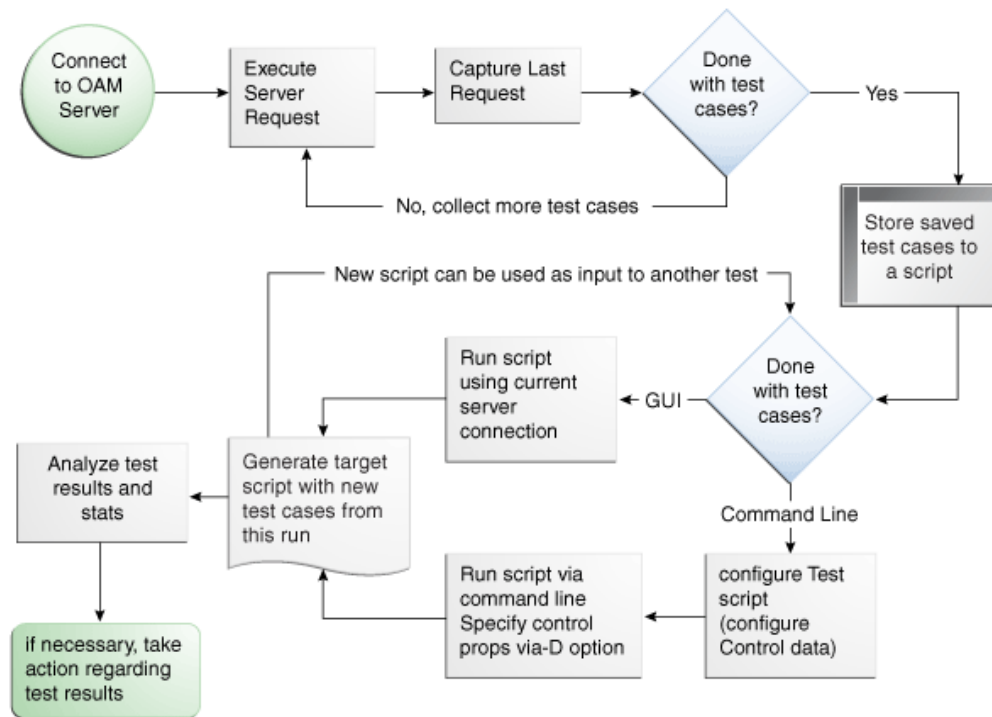
26.6.1 About Test Cases and Test Scripts

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. Test scripts can be configured, run, and generated from the Access Tester Console.

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 26-7](#).

Figure 26-7 Test Case Workflow



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.

26.6.2 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 26-10 describes the location of the capture options.

Table 26-10 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).
Blue up arrow	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](#)".
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.
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](#)".

26.6.3 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. The following sections provide more details:

- [About Input Test Script](#)
- [Generating an Input Test Script](#)

 **Note:**

Do not clear the capture queue until you have captured all the test cases you want to include in the script.

26.6.3.1 About 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 26-11 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.
Paper Script Scroll	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.

26.6.3.2 Generating an Input Test Script

You can capture test cases that you want in your test script and record it.

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](#)".
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 Warning 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.

26.6.4 Personalizing an Input Test Script

This section describes how to personalize and customize a test script.

- [Test Script Control Parameters](#)
- [Customizing a Test Script](#)

26.6.4.1 Test Script Control Parameters

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 26-12](#) describes the control elements and how to customize these.

Table 26-12 Test Script Control Parameters

Control Parameter	Description
<code>ignorecontent=true</code>	<p>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.</p> <p>Default: false (Compare Content sections).</p> <p>Values: true or false</p> <p>In command line mode, use <code>ignorecontent=true</code> to over ride the specified value in the Control section of the input script.</p>
<code>testname="oamtest"</code>	<p>Specifies a prefix to add to file names in the "results bundle" as described in the previous section.</p> <p>In command line mode, use <code>Testname=name</code> to over ride the specified value in the Control section.</p>
<code>configfile="config.xml"</code>	<p>Specifies the absolute path to a configuration XML file that was previously created by the Access Tester.</p> <p>In command line mode, this file is used by the Access Tester to locate connection details to establish a server connection.</p>
<code>numthreads="1"</code>	<p>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.</p> <p>Default: 1</p> <p>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.</p>

Table 26-12 (Cont.) Test Script Control Parameters

Control Parameter	Description
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

26.6.4.2 Customizing a Test Script

You can personalize a test script generated by the Access Tester.

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](#)".

26.6.5 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](#)

26.6.5.1 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 26-13](#) describes the commands to execute a test script.

Table 26-13 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.
Paper Script Scroll with green arrow	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](#)".

26.6.5.2 Running a Test Script

You can submit your test script for processing through Access Tester Console or opt for command line processing.

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](#)".
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](#)".

```
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](#)".

26.7 Evaluating Scripts, Log File, and Statistics

Access Tester generates statistics file along with the target output script and execution log.

This section provides the following information:

- [About Evaluating Test Results](#)
- [Saved Connection Configuration File](#)
- [Generated Input Test Script](#)
- [Target Output File Containing Test Run Results](#)
- [Statistics Document](#)
- [Execution Log](#)

26.7.1 About Evaluating Test Results

You can evaluate test results using the "results bundle" that gets generated at the end of a test run. This bundle contains three documents: target script, execution log, and execution statistics.

Target script is an XML document containing new test cases. 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 "[Generated Input Test Script](#)".



Note:

The target script is not created if the Access Tester is configured to run in multi-threaded mode.

The execution statistics document contains the test metrics, summary and detail statistics, and 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 26-14](#).

Table 26-14 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.

26.7.2 Saved Connection Configuration File

The Saved Connection Configuration File is the output file 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.

Connection Configuration File

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<oamtestconfig xmlns="http://xmlns.example.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>

```

26.7.3 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.

Generated Input Test Script

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<oamtestscript xmlns="http://xmlns.example.com/idm/oam/oamtest/schema"
version="1.0">
  <history description="Manually generated using agent 'agent1'"
createdon="2012-02-03T22:28:00.468-05:00" 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://
emerald.uk.example.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=uk,dc=example,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>

```

26.7.4 Target Output File Containing Test Run Results

Here is an 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 the Example: Output File Generated During a Test Run. As shown in the execution log, this test run found no mismatches, and shows that 4 out of 4 requests matched.

Output File Generated During a Test Run

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<oamtestscript xmlns="http://xmlns.example.com/idm/oam/oamtest/schema"
version="1.0">
  <history description="Generated from script 'script.xml' using agent 'agent1'"
createdon="2012-02-03T23:03:02.171-05:00" 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://
emerald.uk.example.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>
  </cases>
</oamtestscript>
```

```

        </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>

```

26.7.5 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 the Example: Sample Statistics Document. The various sections that provide statistics for this run, which you can compare against statistics for an earlier "known good" run.

Sample Statistics Document

```
A sample statistics document is shown here. Notice,
<oamteststats xmlns="http://xmlns.example.com/idm/oam/oamtest/schema"
version="1.0">
  <history description="Generated from script 'script.xml' using agent
    'agent1'" createdon="2012-02-03T23:03:02.171-05:00" createdby="test_user"/>
  <summary>
    <total>
      <nummatched>4</nummatched>
      <numtotal>4</numtotal>
      <avgelapsedsource>238</avgelapsedsource>
      <avgelapsedtargget>232</avgelapsedtargget>
    </total>
    <validate>
      <nummatched>2</nummatched>
      <numtotal>2</numtotal>
      <avgelapsedsource>578</avgelapsedsource>
      <avgelapsedtargget>570</avgelapsedtargget>
    </validate>
    <authenticate>
      <nummatched>1</nummatched>
      <numtotal>1</numtotal>
      <avgelapsedsource>187</avgelapsedsource>
      <avgelapsedtargget>187</avgelapsedtargget>
    </authenticate>
    <authorize>
      <nummatched>1</nummatched>
      <numtotal>1</numtotal>
      <avgelapsedsource>188</avgelapsedsource>
      <avgelapsedtargget>172</avgelapsedtargget>
    </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>
      </validate>
    </target>
  </detail>
</oamteststats>
```

```

        <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>

```

26.7.6 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.

Execution Log

```

[2/3/12 11:02 PM][info] Setting up to run script 'script.xml'
[2/3/12 11:02 PM][info] Loading test cases and control parameters from script
[2/3/12 11:02 PM][info] Loaded 4 cases
[2/3/12 11:02 PM][info] Control data for this test run:
[2/3/12 11:02 PM][info] Test name : 'samplerun1'
[2/3/12 11:02 PM][info] Configuration file : 'config.xml'
[2/3/12 11:02 PM][info] Ignore content : 'false'
[2/3/12 11:02 PM][info] Loading server configuration from file
[2/3/12 11:02 PM][info] Loaded server configuration
[2/3/12 11:02 PM][info] Connecting to server as agent 'oam_agent1'
[2/3/12 11:03 PM][info][request] Connect : Yes
...
[2/3/12 11:03 PM][info] Test 'samplerun1' will process 4 cases
[2/3/12 11:03 PM][info][request] Validate : Yes
[2/3/12 11:03 PM][info] Authentication scheme : LDAPScheme, level : 2
[2/3/12 11:03 PM][info] Redirect URL :
http://oam_server1.uk.example.com:2100/server/
[2/3/12 11:03 PM][info] Credentials expected: 0x01 (password)
[2/3/12 11:03 PM][info][request] Authenticate : Yes
[2/3/12 11:03 PM][info] User DN : cn=admin1,dc=us,dc=company,dc=com
[2/3/12 11:03 PM][info] Session ID : -1
[2/3/12 11:03 PM][info][request] Authorize : Yes

```

```
[2/3/12 11:03 PM][info][request] Validate : Yes  
[2/3/12 11:03 PM][info] Summary statistics  
[2/3/12 11:03 PM][info] Matched 4 of 4, avg latency 232ms vs 238ms  
[2/3/12 11:03 PM][info] Validate: matched 2 of 2, avg latency 570ms vs 578ms  
[2/3/12 11:03 PM][info] Authenticate: matched 1 of 1, avg latency 187ms vs 187ms  
[2/3/12 11:03 PM][info] Authorize: matched 1 of 1, avg latency 172ms vs 188ms  
[2/3/12 11:03 PM][info] Generated target script 'samplerun1_0302171__target.xml'  
[2/3/12 11:03 PM][info] Generated statistics log 'samplerun1_0302171__stats.xml'
```

Configuring Centralized Logout for Sessions Involving OAM WebGates

You can use the Access Manager single logout (also known as global logout) for sessions that involve OAM WebGates. With Access Manager, single logout refers to the process of terminating an active session. Oracle recommends that you use the logout mechanism that is provided by Access Manager in the manner described in the following topics (not custom logout scripts).

The following topics describe how to configure centralized logout for sessions with OAM WebGates:

- [Prerequisites for the Configuration of Centralized Logout Sessions Involving OAM WebGates](#)
- [Introduction to Centralized Logout for Access Manager](#)
- [Configuring Centralized Logout for OAM WebGates](#)
- [Validating Global Sign-On and Centralized Logout](#)

27.1 Prerequisites for the Configuration of Centralized Logout Sessions Involving OAM WebGates

Before you proceed with the logout sessions configuration ensure the system level requirements are met.

Following are the requirements to perform tasks in this chapter:

- The application must be deployed on the Web server where the agent is configured and registered with Access Manager.
- One OAM Agent, on any supported Web server and platform, must be running and registered with Access Manager 12c.
- Policies must be configured to protect the resource in an Access Manager 12c Application Domain.

27.2 Introduction to Centralized Logout for Access Manager

Centralized logout refers to the process of terminating an active session. OAM provides centralized logout for sessions.

Unless explicitly stated, information in this chapter applies to OAM WebGate Agents using the default embedded credential collector (ECC).

This section provides the following topics:

- [About Centralized Logout for OAM WebGates](#)
- [About Logout Parameters for OAM WebGates](#)

27.2.1 About Centralized Logout for OAM WebGates

Access Manager provides centralized logout (also known as global log out) for sessions.

Centralized logout refers to the process of terminating an active session, which means that:

- 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 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.

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.

[Table 27-1](#) describes the circumstances under which centralized logout occurs. When the logout URL is encountered and the OAMAuthnCookie cookie for OAM Webgates is removed Webgate logs out the user and requires user re-authentication.

Table 27-1 Centralized Logout Circumstances

Circumstance	Description
Explicitly	<p>The client state is invalidated and the session ends. If a new attempt is made to access the resource, the client must re-authenticate.</p> <ul style="list-style-type: none"> • When the user logs out. • When the Administrator terminates the session • When the session is terminated based on changes on the identity side
Implicitly	<p>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.</p> <p>With Access Manager, the user is not logged out if OAM Webgate simply encounters a logout URL unless the logout.html provides an explicit redirection to the Server logout. The Webgate redirects the user to the Server logout.</p>

27.2.2 About Logout Parameters for OAM WebGates

Generally speaking, during centralized logout, the SSO Engine receives a `user-session-exists` request and sends out a `Session Cleared` response.

When the SSO Engine receives a `user-session-exists` request, the Session Management Engine looks up the session and responds with the `session-exists` response. The SSO engine sends a `Clear Session` request. The Session management engine clears the token and session context. The SSO engine then sends a `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 application.

Configuring OAM WebGates for logout against OAM Servers requires a `Logout Callback URL` (Table 15-3). Centralized logout for 12c agents sets the cookie from `loggedout` to empty and expires `OAMAuthnCookie_<host:port>_<random number>` to explicitly clear it during logout, (rather than leaving behind an empty or logged out cookie).

The SSO Engine supports the central logout page on the OAM Server and:

- Calls back to `Logout Callback URL` of OAM WebGates during logout

The WebGate parameter `Logout Callback URL` can be configured using a URI format (recommended), without `host:port`. OAM Server dynamically constructs the full URL based on the `host:port` in the original request and calls back on it. This can also be a full URL format with a `host:port`, where OAM Server calls back directly without reconstructing callback URL.

- Lands on `end_url` (passed in as query parameter) after logout

Several elements in the OAM Webgate registration page enable centralized logout for OAM WebGates. After registration, the `ObAccessClient.xml` file is populated with the information in Table 27-2.

Table 27-2 Logout Details After Registration (ObAccessClient.xml)

Element	Description
Logout URL OAM WebGates	<p>The Logout URL triggers the logout handler, which removes the cookie (OAMAuthnCookie for OAM WebGates) and requires the user to re-authenticate the next time he accesses a resource protected by Access Manager.</p> <ul style="list-style-type: none"> • 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. <p>Default = [] (not set)</p> <p>Note: This is the standard OAM WebGate configuration parameter used to trigger initial logout through a customized local logout page.</p>
Additional Logout for OAM WebGate Only	<p>For OAM 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 OAMWebGate single sign-off only through a customized local logout page.</p>

Table 27-2 (Cont.) Logout Details After Registration (ObAccessClient.xml)

Element	Description
Logout Callback URL	<p>The URL to <code>oam_logout_success</code>, which clears cookies during the call back. This can be a URI format without <code>host:port</code> (recommended), where the OAM Server calls back on the <code>host:port</code> of the original resource request. For example:</p> <p>Default = <code>/oam_logout_success</code></p> <p>This can also be a full URL format with a <code>host:port</code>, where OAM Server calls back directly without reconstructing callback URL.</p> <p>When the request URL matches the Logout Callback URL, Webgate clear its cookies and streams an image <code>.gif</code> in the response.</p> <p>When Webgate redirects to the server logout page, it records an "end" URL as a query parameter (<code>end_url=http://host:port/...</code>), which becomes the landing page that the OAM Server redirects back to after logout.</p> <p>Note: In the remote registration template this parameter is named <code>logoutCallbackUrl</code> (Table 15-10).</p> <p>Other Oracle Access Management services support the central logout page on the server. The <code>end_url</code> relies on the target URL query parameter passed from OPSS integrated applications. See Also: "Configuring Centralized Logout for Oracle ADF-Coded Applications".</p>
Logout Redirect URL	<p>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:</p> <p>Default = <code>http://OAMServer_host:14200/oam/server/logout</code></p> <p>The Logout URL triggers the logout handler, which removes the <code>OAMAuthnCookie_<host:port>_<random number></code> and requires the user to re-authenticate the next time he accesses a resource protected by Access Manager.</p> <ul style="list-style-type: none"> • When Webgate logout handler is triggered, it redirects to the central logout page specified by the Logout Redirect URL parameter if it is configured. • If this is explicitly cleared (and not configured), then 12c behavior is triggered. 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	<p>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.</p> <p>Default: <code>end_url</code></p> <p>Note: The <code>end_url</code> value is configured using <code>param.logout.targeturl</code> in <code>jps-config.xml</code>.</p> <ul style="list-style-type: none"> • If Logout Target URL is configured, Webgate searches for the value passed in the logout request's query parameter and passes it as <code>end_url</code> 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 <code>end_url</code> query parameter along.

27.3 Configuring Centralized Logout for OAM WebGates

You can configure centralized logout for both ECC and DCC enabled OAM WebGates.

This section provides the following topics:

- [Configuring Centralized Logout for OAM WebGates When the ECC is Used](#)
- [Configuring Logout When Using Detached Credential Collector-Enabled WebGate](#)



See Also:

[Configuring Centralized Logout for Oracle ADF-Coded Applications](#)

27.3.1 Configuring Centralized Logout for OAM WebGates When the ECC is Used

During 12c Resource WebGate registration or editing, you configure the logout parameters.



Note:

If the `LogOutUrl` parameter is already configured for the OAM WebGate (with a value other than `/oamssso/logout.html`), then ensure that is also present as part of the `LogOutUrl` parameter.



See Also:

["Configuring Logout When Using Detached Credential Collector-Enabled WebGate"](#)

To configure centralized logout for OAM WebGates:

1. Choose your method for registration described in [Registering and Managing OAM Agents](#)
2. When creating or editing an agent registration, include appropriate logout values for your environment ([Table 27-2](#)):
 - Logout URL
 - Logout Callback URL
 - Logout Redirect URL
 - Logout Target URL
3. Finish and save your agent registration, as usual.
4. **Multiple DNS Domains:** Perform the following steps if you have multiple DNS domains configured for Access Manager 12c SSO.



Note:

The `Logout Callback URL` can be unique for each WebGate; however, to construct the `Logout 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 Logout Callback URL as the second value in the `logOutUrls` parameter on each resource WebGate.

Logout Callback URL is the location on WebGate that the request must be sent to, for clearing the SSO Cookie in that domain. The Logout Callback URL cannot be `logout.html`.

- b. Ensure that a file physically exists on each Web server at the Logout Callback URL 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 the Logout Callback URL of `logout.png` would be:

```
/oamssso/logout.png
```

5. Perform steps in "[Validating Global Sign-On and Centralized Logout](#)".

27.3.2 Configuring Logout When Using Detached Credential Collector-Enabled WebGate

When the DCC receives a logout request from the Agent, the DCC:

- Decrypts the logout request, if needed
- Retrieves the `end_url`, constructs the full URL with the Agent's `host:port` if needed
- Clears the DCC cookie (`DCCtxCookie`)
- Sends the logout request across the back channel to terminate the session
- Logout Callback URLLogout Callback URLsGets a logout page containing links to all visited agent from OAM Sever (which has this information), or get only a list of the visited from OAM Sever to construct a logout page locally, and redirect user to this page on DCC.
- Returns to the `end_url` after logout completes

To configure logout for Resource Webgates separate from DCC:

1. Confirm that the Perl scripts for DCC logout include the actual location of the Perl executable on the Webgate host `$WEBGATE_HOME/oamssso-bin/*.pl`.
2. **Resource Webgate:** Modify the Logout Redirect URL to point to DCC's `logout.pl`:
 - a. **Find the Resource Webgate Registration:** See "[WebGate Search Controls](#)".
 - b. Modify the Logout Redirect URL to point to the DCC's `logout.pl`. For example:

```
http://DCCWghost:port/oamssso-bin/logout.pl
```

Note:

The DCC ignores the Logout Redirect URL parameter in the Webgate registration page. However, if the Resource Webgate Logout Redirect URL is anything other than `logout.*`, then that URL must be defined in DCC Logout URLs. See [Table 24-3](#)

3. Perform steps in "[Validating Global Sign-On and Centralized Logout](#)".

27.4 Validating Global Sign-On and Centralized Logout

You can validate single sign-on global login with different applications, and centralized logout for single or two applications.

This section provides the following topics:

- [Confirming Global Sign-On](#)
- [Observing Centralized Logout](#)

27.4.1 Confirming Global Sign-On

You can observe single sign-on global login.

You must meet the following prerequisites:

- Agents and Servers must be registered with Access Manager and running
- Resources and policies controlling SSO must be defined within Access Manager Application Domains

To observe global sign-on:

1. From a browser, enter the URL to a protected resource.
2. On the login page, sign in using proper credentials.
3. Verify that the resource is presented; do not log out.
4. In the same browser window, enter the URL to another protected resource and confirm that the resource is presented without having to re-authenticate.

27.4.2 Observing Centralized Logout

You can observe centralized logout with OAM Agents.

- 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.

You must meet the following prerequisites:

- Agents must be registered and running
- Resources must be protected by Access Manager Application Domains
- Single sign-on must be configured with authentication and authorization policies and responses in Access Manager 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 and confirm that the second application does not require another login.
 - d. Log out of the first application.
 - e. Access the second application and confirm that the login page appears.

Part VII

Managing Oracle Access Management Identity Federation

Oracle Identity Federation enables business partners to achieve a federated environment by providing the mechanism with which companies can form a federation and securely share services and data across their respective security domains.

This section contains the following chapters:

- [Introducing Identity Federation in Oracle Access Management](#)
- [Managing Identity Federation Partners](#)
- [Managing Settings for Identity Federation](#)
- [Managing Federation Schemes and Policies](#)

Introducing Identity Federation in Oracle Access Management

A *federation* is defined as "an association formed by merging several groups or parties". A federated environment (as defined in the identity management realm) is one in which organizations that provide services and identity data (business partners) have established trust in order to share access to a set of protected resources while protecting the same from unauthorized access. Oracle Identity Federation enables business partners to achieve this by providing the mechanism with which companies can form a federation and securely share services and data across their respective security domains.

With the 11g Release 2 (11.1.2.3) of Oracle Access Management, the standalone Oracle Identity Federation product has begun its integration with Oracle Access Manager. This chapter introduces the integrated Identity Federation and includes the following topics

- [Integrating Identity Federation with Access Manager](#)
- [Deploying Identity Federation with Oracle Access Management](#)
- [Understanding How Identity Federation Works](#)
- [Using Identity Federation](#)
- [Initiating Federation SSO](#)
- [Exchanging Identity Federation Data](#)
- [Administering Identity Federation](#)
- [Enabling Identity Federation](#)

28.1 Integrating Identity Federation with Access Manager

The Oracle Identity Management framework supports two approaches to cross-domain single sign-on. You cannot mix-and-match these approaches as each stands on its own.

Based on your setup, perform one of the following:

1. Beginning with the 11g Release 2 (11.1.2), the Oracle Access Management Access Manager server (OAM Server) has been integrated with an Oracle Access Management Identity Federation server. All configuration for the Identity Federation server is performed using the Oracle Access Management Console.
2. Previous, separate releases of Oracle Identity Federation (11.1.1) and Oracle Access Manager can still be deployed to provide federation capabilities. Both servers must be configured and managed for this integration. This approach existed in 11g Release 1 (11.1.1) and is still available.

 **Note:**

The topics in this book presume familiarity with federation and how it works. See Introduction to Oracle Identity Federation in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation* for background and conceptual information. This current document is limited to describing Oracle Identity Federation functionality as it has been integrated with Access Manager in 12c.

Benefits of using the new Identity Federation 12c server integrated with Access Manager include:

- Eliminating the need to install and maintain separate servers.
- Simplifying post-install configuration of the federation features, particularly when accessing those features through the Oracle Access Management Console.
- Improving the scalability of the two services working together.
- Providing enhanced diagnostics and troubleshooting.

28.2 Deploying Identity Federation with Oracle Access Management

From a functional perspective, the components using the Identity Federation service (when a user attempts to log in to a protected resource using a Web browser) include the Access Manager server, Oracle WebLogic server hosts, and data stores.

Access Manager Server

The Access Manager server contains all the components needed to provide access management services in the federated context, including:

- a credential collector
- a federation authentication plugin
- the Identity Federation engine to generate and process assertions
- a federation data cache

Oracle WebLogic Server

Oracle WebLogic Server hosts and provides key infrastructure services, including:

- the authorization engine, which interacts with Oracle Entitlement Server
- federation data including circle of trust details and other configuration

Data Stores

Data stores, including the identity store, maintain the identity data needed for authentication tasks. Identity Federation supports the Access Manager common user store and provides multiple identity store support. Federation data for persistent account linking can be stored in a database.

**Note:**

Calls are routine HTTP calls.

28.3 Understanding How Identity Federation Works

A federation can comprise any number of identity providers and service providers. An instance of Identity Federation in a federated network can serve as either an identity provider, a service provider, or both.

One common federated network topology is referred to as the hub-and-spoke model. In this topology, there is either a single service provider accepting authentication from multiple identity providers, or a single identity provider authenticating users for multiple service providers.

- A service provider (SP) is a commercial or not-for-profit organization that offers a web-based service such as a news portal, a financial repository, or retail outlet. When configured as the SP in a federated network and a user wants to access a resource protected by an authentication engine such as Oracle Access Manager, Identity Federation redirects the user to an IdP for global authentication. The IdP will obtain credentials, authenticate the user, and redirect the user back to the Identity Federation server instance - which retrieves the asserted identity from the IdP and redirects the authenticated user to the authentication engine which provides access to the protected resource.
- An identity provider (IdP) is a service provider that stores identity profiles. (Identity providers might also offer services above and beyond those related to identity profile storage.) When configured as the IdP in a federated network and a user wants to access a protected resource, the resource's SP directs the user to the Identity Federation server instance - which uses the Access Manager authentication engine to obtain credentials and authenticate the user. Following successful authentication, the Identity Federation instance can assert the user's identity to the resource's SP - which then authenticates the user itself and provides access to the requested resource.

The integrated Identity Federation server can operate as an IdP or an SP. See [Managing Identity Federation Partners](#) for information on configuring Identity Federation to operate in one of these provider modes and communicate with remote partners in the federation.

28.4 Using Identity Federation

In SP-initiated SSO, the federated SSO process begins when the SP sends an authentication request to the IdP. In IdP-initiated SSO, the IdP sends the SP an unsolicited assertion response (in the absence of an authentication request from the SP). Supported runtime flows in both modes include SSO, Logout (initiated from a remote federation partner or Access Manager protected application) and Attribute Query.

This section describes the following topics:

- [Achieving SSO](#)
- [Logging Out](#)
- [Authorizing](#)
- [Forcing Authentication](#)

- [Indicating a Passive Identity Provider](#)
- [User and Assertion Mapping](#)
- [Platform Dependencies](#)

28.4.1 Achieving SSO

When the Identity Federation (acting as an IdP) is performing federated SSO with an SP, the Access Manager server authenticates the user or ensures an authenticated user doesn't need to be challenged due to inactivity.

Additionally, the Access Manager server will check that any requested federation authentication method specified by the SP does not require a challenge based on authentication level. The Authentication Scheme mappings to the authentication methods will determine this. (If the SP does not specify a Federation Authentication Method, the IdP will use the one specified for the SP partner in the defaultschemeid property.) See [Initiating Federation SSO](#) for details.

28.4.2 Logging Out

With Identity Federation, a logout operation is dissociated from the authentication operation. Logout can be initiated by user the (Access Manager server) or a partner in the federation.

- When initiated by the user accessing the Access Manager Logout service, Access Manager kills the user's Access Manager session and displays a logout page that will instruct the various WebGate agents to remove the user cookies. Access Manager then redirects the user to the Identity Federation Logout service which notifies each partner involved in this session by either redirecting the user with a Logout Request message via HTTP Redirect or HTTP POST or by directly sending a Logout Request message via SOAP. Identity Federation then kills the OIF session and redirects the user to the defined return URL.
- When initiated by the user on a web site from a partner in the federation, the partner redirects the user to the Identity Federation server which marks the user session as logging out. Identity Federation then redirects the user to the Access Manager server which kills the user's Access Manager session. Access Manager then displays a logout page that will instruct the various WebGate agents to remove the user cookies, and redirects the user back to Identity Federation to resume the Federation logout process by notifying each partner involved in this session (except the one who first redirected the user) by either redirecting the user with a Logout Request message via HTTP Redirect or HTTP POST or by directly sending a Logout Request message via SOAP. Identity Federation then kills the OIF session and redirects the user with a Logout Response message to the partner who first redirected the user to the Identity Federation server.

28.4.3 Authorizing

When the Identity Federation server acts as an IdP, it has the need to issue an Identity Token to the SP during the Federation SSO operation.

The Identity Token will contain user information as well as session information. By default, the authorization feature is turned off. It can be enabled or disabled using the `configureFedSSOAuthz WLST` command. You also need to create a resource of type `TokenServiceRP` (with the Resource URL set to the SP Partner ID) and a Token

Issuance Policy to which the Resource is added. The Token Issuance Policy indicates the conditions under which the token should be issued.

28.4.4 Forcing Authentication

SAML 2.0 and OpenID 2.0 provide a way for a SP to indicate during Federation SSO whether the user should be challenged by the IdP, even if a valid user session already exists.

In this case, the SP will send an authentication request with a parameter indicating that the IdP should re-challenge the user or force authentication.

28.4.5 Indicating a Passive Identity Provider

SAML 2.0 and OpenID 2.0 provide a way for the SP to indicate during Federation SSO whether the Identity Provider should interact with the user.

In this case, the SP will send an authentication request with a parameter indicating that the IdP should not interact with the user or is passive. The IdP recognizes the parameter and returns to the SP:

- An error if the IdP must interact with the user but cannot because of this parameter.
- A Federation Assertion that indicates whether the user has a valid session.

28.4.6 User and Assertion Mapping

In Identity Federation, after a SP validates the SAML assertion created by its IdP partner, it can map the assertion to the local user.

In one of the following ways, Identity Federation maps SAML assertion to the local user:

- By mapping the SAML subject to a user record with a user attribute (for example, `mail`).
- By mapping a SAML Assertion Attribute to a user record with a user attribute (for example, the SAML Assertion Attribute `emailAddress` mapped to `mail`).
- By mapping one or more attributes contained in the SAML assertion's `AttributeStatement` element or the SAML subject with an LDAP query. You must configure both the SAML attribute name and the user attribute to which it is mapped.

28.4.7 Platform Dependencies

The architecture leverages the Oracle Fusion Middleware platform for the Credential Store Framework (CSF).

CSF securely stores keystore passwords as well as server credentials such as HTTP Basic Authentication usernames and passwords.

28.5 Initiating Federation SSO

Federation SSO process can be initiated when Identity Federation is working as an IdP or SP.

This section describes the following topics:

- [IdP Initiated Federation SSO Service](#)
- [SP Initiated Federation SSO Service](#)

- [Attribute Consuming Service](#)

28.5.1 IdP Initiated Federation SSO Service

The IdP Initiated Federation SSO Service has three query parameters: `providerid`, `returnurl`, and `acsurl`.

When Identity Federation is working as an IdP, the URL for initiating Federation SSO is:

```
http://public-oam-host:public-oam-port/oamfed/idp/initiatessso
```

The query parameters are:

- `providerid`: name of the SP partner with which to perform Federation SSO or the issuer ID / provider ID of the SP partner with which to perform Federation SSO. (required)
- `returnurl`: the SP URL where the user will be redirected after a successful Federation SSO (optional)
- `acsurl`: the SAML 2.0 Assertion Consumer Service URL where Identity Federation will redirect the user with the SAML 2.0 Assertion. This URL must be declared in the SP SAML 2.0 Metadata. (optional)

[Multivalue Attributes in SAML Assertion](#)

28.5.1.1 Multivalue Attributes in SAML Assertion

The default behavior of the feature is, during SSO, IDP sends the Group attributes in comma separated format if the user belongs to multiple groups and `always_send` is set to `true`. As an enhanced behavior, During SSO, IdP sends the Group attributes in separate SAML statements instead of comma separated if the `multivaluegroups` flag is set to `true`.

The following SSO protocols support Multi-Valued Groups SAML Attributes

- SAML 2.0
 - SAML 1.1
1. To enable this feature, OAM configuration should be updated depending on the requirement. The `multivaluegroups` attribute setting is disabled by default and is not present in `oam-config.xml`. The User has to add this setting in `oam-config.xml` using WLST commands and set it to `true` to enable multiple attribute statements for Group attribute.
 2. Add `multivaluegroups` attribute setting to `oam-config.xml` at the Partner level or Partner Profile level or Global level using WLST commands and set it to `true`.

```
<Setting Name="multivaluegroups" Type="xsd:boolean">true</Setting>
```

- Enable or disable the `multivaluegroups` at partner level

```
updatePartnerProperty (partnerName="spPartnername",
partnerType="SP",propName="multivaluegroups",propValue="true/false",type="boolean");
```
- Enable or disable the `multivaluegroups` at partner profile level

```
putBooleanProperty("/fedpartnerprofiles/saml20-sp-partner-  
profile/multivaluegroups", "true/false");
```

- **Enable or disable the multivaluegroups at global level**

```
putBooleanProperty("/idpglobal/multivaluegroups", "true/  
false");
```

28.5.2 SP Initiated Federation SSO Service

The SP Initiated Federation SSO Service has two query parameters: `providerid` and `returnurl`.

When Identity Federation is working as an SP, the URL for initiating Federation SSO is:

```
http://public-oam-host:public-oam-port/oamfed/sp/initiatesso
```

The query parameters are:

- `providerid`: name of the IdP partner with which to perform Federation SSO or the issuer ID / provider ID of the IdP partner with which to perform Federation SSO. (required)
- `returnurl`: the URL where the user will be redirected after a successful Federation SSO (optional)

28.5.3 Attribute Consuming Service

OAM Federation service is enhanced to support standard SAML2v-based interfaces and elements.

This section describes the following topics:

- [Elements of Attribute Consuming Service](#)
- [WLST Commands for Attribute Consuming Service](#)

28.5.3.1 Elements Of Attribute Consuming Service

The attribute consuming service includes three elements: `AttributeConsumingService`, `AttributeConsumingServiceIndex`, and `NameQualifier`.

AttributeConsumingService

The `AttributeConsumingService` element is included in the SP metadata. This element contains the following fields:

- `ServiceName`
- `ServiceDescription`
- `index`
- `isDefault`
- `RequestedAttribute` contains the following fields:
 - `acsIndex`
 - `rqstAttrName`

 **Note:**

The `rqstAttrName` field can be any user defined value such as name, fiscal number, email, and so on.

- `rqstAttrNameFormat`
- `rqstAttrFriendlyName`
- `rqstAttrIsRequired`

Sample SP metadata:

```
<md:AttributeConsumingService index="1" isDefault="false">
  <md:ServiceName xml:lang="en_US">serviceName1</md:ServiceName>
  <md:ServiceDescription xml:lang="en_US">serviceDesc1</md:ServiceDescription>
  <md:RequestedAttribute FriendlyName="friendlyName1" Name="email"
NameFormat="sample:urn:format" isRequired="true"/>
</md:AttributeConsumingService>

<md:AttributeConsumingService index="1" isDefault="true">
  <md:ServiceName xml:lang="eng">Updated-Service-Name1</md:ServiceName>
  <md:ServiceDescription xml:lang="eng">updatedServiceDesc</
md:ServiceDescription>
  <md:RequestedAttribute FriendlyName="friendlyName1" Name="email"
NameFormat="sample:urn:format" isRequired="true"/>
  <md:RequestedAttribute FriendlyName="" Name="empNum" NameFormat="empFormat1"
isRequired="false"/>
  <md:RequestedAttribute FriendlyName="fname" Name="empFirstName"
NameFormat="firstnameformat1" isRequired="true"/>
</md:AttributeConsumingService>
```

AttributeConsumingService

The `AttributeConsumingServiceIndex` element is included in the SAML 2.0 authentication request. In the runtime SSO, pass the `attributeconsumingserviceindex` parameter in the SP initiated URL, so that `AttributeConsumingServiceIndex` is displayed in the `authnrequest`.

For example, `http://sp-host:sp-managed-port/oamfed/sp/initiatesso?providerid=http://idp-host:idp-managed-port/oam/fed&returnurl=http://sp-host:webgate-port/cgi-bin/headers.cgi&attributeconsumingserviceindex=1`

Sample authentication request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <samlp:AuthnRequest 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:2.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
AttributeConsumingServiceIndex="1" ID="id-atMY1jR9Vh7PBcWSjdqmyxIc1JNMSFD-
zQ1d71f8" Version="2.0" IssueInstant="2016-09-15T22:32:37Z" Destination="http://
slc05ynv.us.oracle.com:21328/oamfed/idp/samlv20">
  <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">http://slc06fcv.us.oracle.com:23768/oam/fed</saml:Issuer>
  <samlp:NameIDPolicy AllowCreate="true"/>
</samlp:AuthnRequest>
```

NameQualifier

The `NameQualifier` element is included in the `<samlp:issuer>` tag.

Example:

```
<saml:Issuer NameQualifier=" http://spid-sp.it"
Format=" urn:oasis:names:tc:SAML:2.0:nameid format:entity"> SPID-sp-test </saml:Issuer>
```

By default, `NameQualifier` is set to `false`. You can set `NameQualifier` to `true` in the **oam-config.xml** file using the WLST commands. For more information on WLST commands, see the *WLST Command Reference for WebLogic Server*.

The following table illustrates how to enable or disable the `NameQualifier` element using the WLST commands:

Action	WLST Command Examples
Enable <code>NameQualifier</code> at the partner level.	<code>updatePartnerProperty (partnerName="idp-partner", partnerType="IDP", propName="samlrequestissuernamequalifier", propValue="http://sample.sp.it", type="string")</code>
Enable <code>NameQualifier</code> at the partner profile level.	<code>putStringProperty ("/fedpartnerprofiles/saml20-idp-partner-profile/samlrequestissuernamequalifier", "http://profile-sample.it")</code>
Enable <code>NameQualifier</code> at the global level.	<code>putStringProperty ("/spglobal/samlrequestissuernamequalifier", "http://spglobal.it")</code>
Disable <code>NameQualifier</code> at the partner level.	<code>deletePartnerProperty (partnerName="idp-partner", partnerType="IDP", propName="samlrequestissuernamequalifier")</code>
Disable <code>NameQualifier</code> at the partner profile level.	<code>deleteStringProperty ("/fedpartnerprofiles/saml20-idp-partner-profile/samlrequestissuernamequalifier")</code>
Disable <code>NameQualifier</code> at the global level.	<code>deleteStringProperty ("/spglobal/samlrequestissuernamequalifier")</code>

28.5.3.2 WLST Commands For Attribute Consuming Service

Attribute Consuming Service is supported with ten WebLogic Scripting Tool (WLST) commands.

More information in the following sections:

- [getDefaultACS](#)
- [getAllRqstAttrsForACS](#)
- [getAllACS](#)
- [getACS](#)

- [addACS](#)
- [addRqstAttrToACS](#)
- [updateACS](#)
- [updateRqstAttrForACS](#)
- [deleteACS](#)
- [deleteRqstAttrForACS](#)

28.5.3.2.1 getDefaultACS

This command retrieves the default attribute consuming service.

Description

The `getDefaultACS` command retrieves the default attribute consuming service.

Syntax

```
getDefaultACS()
```

Example

This example illustrates the use of `getDefaultACS` command.

```
getDefaultACS()
```

28.5.3.2.2 getAllRqstAttrsForACS

This command retrieves the list of requested attributes under specified attribute consuming service, `acsIndex`.

Description

The `getAllRqstAttrsForACS` command retrieves the list of requested attributes under the specified attribute consuming service, `acsIndex`.

Syntax

```
getAllRqstAttrsForACS(acsIndex)
```

Arguments	Definition
<code>acsIndex</code>	[Mandatory] Index of the attribute consuming service.

Example

This example illustrates the use of the `getAllRqstAttrsForACS(acsIndex)` command.

```
getAllRqstAttrsForACS(1)
```


28.5.3.2.3 getAllACS

This command retrieves the list of all attribute consuming service configured.

Description

The `getAllACS` command retrieves the list of all attribute consuming service configured.

Syntax

```
getAllACS()
```

Example

This example illustrates the use of the `getAllACS()` command.

```
getAllACS()
```

28.5.3.2.4 getACS

This command retrieves the specified attribute consuming service, `acsIndex`.

Description

The `getACS` command retrieves the specified attribute consuming service, `acsIndex`.

Syntax

```
getACS(acsIndex)
```

Arguments	Definitions
<code>acsIndex</code>	[Mandatory] Index of the attribute consuming service.

Example

This example illustrates the use of the `getACS(acsIndex)` command.

```
getACS(1)
```

28.5.3.2.5 addACS

This command creates a new entry of attribute consuming service with `acsIndex`, `serviceName`, `attributeConsumingIsDefault`, `rqstAttrName`, `rqstAttrNameFormat`, `rqstAttrFriendlyName`, `rqstAttrIsRequired`, and `serviceDescription`.

Description

This command creates a new entry of attribute consuming service with `acsIndex`, `serviceName`, `attributeConsumingIsDefault`, `rqstAttrName`, `rqstAttrNameFormat`, `rqstAttrFriendlyName`, `rqstAttrIsRequired`, and `serviceDescription`.

It is mandatory to provide details of at least one requested attribute when you create an attribute consuming service. The `<xml:lang>` parameter is updated with the server locale automatically.

**Note:**

you must create `addACS` with `acsIndex` to execute GET and DELETE WLST commands.

Syntax

```
addACS(acsIndex, serviceName, rqstAttrName, rqstAttrNameFormat,
rqstAttrFriendlyName="", rqstAttrIsRequired="false", serviceDescription="",
attributeConsumingIsDefault="false")
```

Arguments	Definitions
<code>acsIndex</code>	[Mandatory] Specifies the index of the attribute consuming service.
<code>serviceName</code>	[Mandatory] Specifies the name of the service.
<code>rqstAttrName</code>	[Mandatory] Specifies the name of the requested attribute.
<code>rqstAttrNameFormat</code>	[Mandatory] Specifies the format of the requested attribute.
<code>rqstAttrFriendlyName</code>	[Optional] Specifies the friendly name of the attribute consuming service.
<code>rqstAttrIsRequired</code>	[Optional] Determines if the requested attribute is required. The valid values are <code>true</code> and <code>false</code> .
<code>serviceDescription</code>	[Optional] Provides the description of the service. The default value is "".
<code>attributeConsumingIsDefault</code>	[Optional] Accepts the value to set the default attribute consuming service. The default value is <code>false</code> .

Example

This example illustrates the use of the `addACS` command.

```
addACS(1, "Updated-Service-Name1", "email", "sample:urn:format",
rqstAttrFriendlyName="", rqstAttrIsRequired="false",
serviceDescription="updatedServiceDesc", attributeConsumingIsDefault="true")
```

28.5.3.2.6 addRqstAttrToACS

This command adds a new requested attribute such as `rqstAttrName`, `rqstAttrNameFormat`, `rqstAttrFriendlyName`, and `rqstAttrIsRequired` under the list of specified attribute consuming service, `acsIndex`.

Description

The `addRqstAttrToACS` command adds a new requested attribute such as `rqstAttrName`, `rqstAttrNameFormat`, `rqstAttrFriendlyName`, and `rqstAttrIsRequired` under the list of specified attribute consuming service, `acsIndex`.

Syntax

```
addRqstAttrToACS(acsIndex, rqstAttrName, rqstAttrNameFormat,
rqstAttrFriendlyName=None, rqstAttrIsRequired="false"):
```

Arguments	Definitions
acsIndex	[Mandatory] Specifies the index of the attribute consuming service.
rqstAttrName	[Mandatory] Specifies the name of the requested attribute.
rqstAttrNameFormat	[Mandatory] Specifies the format of the requested attribute.
rqstAttrFriendlyName	[Optional] Specifies the friendly name of the attribute consuming service.
rqstAttrIsRequired	[Optional] Determines if the requested attribute is required. The valid values are true and false.

Example

This example illustrates the use of the `addRqstAttrToACS` command.

```
addRqstAttrToACS(1, "empNumber", "empFormat1", rqstAttrFriendlyName=None,
rqstAttrIsRequired="false"):
```

28.5.3.2.7 updateACS

This commands updates any or all fields of the specified attribute consuming service, `oldACSIndex`.

Description

The `updateACS` command updates any or all fields (that is, `newServiceName`, `newServiceDescription`, `newAttributeLang`, `newIsDefault`, and `newACSIndex`) of the specified attribute consuming service, `oldACSIndex`.

Syntax

```
updateACS(oldACSIndex, newServiceName=None, newServiceDescription=None,
newAttributeLang=None, newIsDefault=None, newACSIndex=None)
```

Arguments	Definitions
oldACSIndex	[Mandatory] Specifies the name of the existing attribute consuming service index.
newServiceName	[Optional] Specifies the updated name for the attribute consuming service.
newServiceDescription	[Optional] Specifies the updated description of the attribute consuming service.
newAttributeLang	[Optional] Specifies the updated "xml:lang" for the attribute consuming service name and description.
newIsDefault	[Optional] Accepts the values such as true or false to set the new default value to attribute consuming service. The valid values are true and false.
newACSIndex	[Optional] Specifies the name of the new attribute consuming service index.

Example

This example illustrates the use of the `updateACS` command.

 **Note:**

At least one optional parameter is required with `acsIndex` to successfully update the ACS.

- To update only the `newServiceName` field, use the following command:

```
updateACS(1, newServiceName="SampleAttributeName");
```
- To update both `newServiceName` and `isdefault`, use the following command:

```
updateACS(1, newServiceName="SampleAttributeName", newIsDefault="true");
```

28.5.3.2.8 updateRqstAttrForACS

This command updates all the fields of the specified requested attribute, `oldRqstAttrName` under the specified attribute consuming service, `acsIndex`.

Description

The `updateRqstAttrForACS` command updates all the fields of the specified requested attribute, `oldRqstAttrName` under the specified attribute consuming service, `acsIndex`.

Syntax

```
updateRqstAttrForACS(acsIndex, oldRqstAttrName, newRqstAttrName=None,  
newRqstAttrFriendlyName=None, newRqstAttrNameFormat=None,  
newRqstAttrIsRequired=None)
```

Arguments	Definitions
<code>acsIndex</code>	[Mandatory] Specifies the index of the attribute consuming service.
<code>oldRqstAttrName</code>	[Mandatory] Specifies the name of the existing requested attribute that updates the fields.
<code>newRqstAttrName</code>	[Optional] Specifies the updated name of the requested attribute.
<code>newRqstAttrFriendlyName</code>	[Optional] Specifies the updated friendly name of the requested attribute.
<code>newRqstAttrNameFormat</code>	[Optional] Specifies the updated format of the requested attribute name.
<code>newRqstAttrIsRequired</code>	[Optional] Determines if the requested attribute is required from the attribute consuming service. The valid values are <code>true</code> and <code>false</code> .

Example

This example illustrates the use of the `updateRqstAttrForACS` command.

 **Note:**

At least one optional parameter is required with `acsIndex` to successfully update the specified requested attribute.

- To update only `newRqstAttrName`, use the following command:

```
updateRqstAttrForACS(acsIndex, oldRqstAttrName,  
newRqstAttrName="SAMPLE_RQST_ATTR");
```

- To update `newRqstAttrName` and `newRqstAttrNameFormat` of the requested attribute, use the following command:

```
updateRqstAttrForACS(acsIndex, oldRqstAttrName,  
newRqstAttrName="SAMPLE_RQST_ATTR", newRqstAttrNameFormat="urn:oasis:sample");
```

28.5.3.2.9 deleteACS

This command deletes the specified attribute consuming service, `acsIndex`.

Description

The `deleteACS` command deletes the specified attribute consuming service, `acsIndex`.

Syntax

```
deleteACS(acsIndex)
```

Arguments	Definition
<code>acsIndex</code>	[Mandatory] Specifies the index of the attribute consuming service.

Example

This example illustrates the use of the `deleteACS` command.

```
deleteACS(1)
```

28.5.3.2.10 deleteRqstAttrForACS

This command deletes the requested attribute, `rqstAttrName`, from the specified Attribute Consuming Service, `acsIndex`.

Description

The `deleteRqstAttrForACS` command deletes the requested attribute, `rqstAttrName`, from the specified attribute consuming service, `acsIndex`.

Syntax

```
deleteRqstAttrForACS(acsIndex, rqstAttrName)
```

Arguments	Definition
<code>acsIndex</code>	[Mandatory] Specifies the index of the attribute consuming service.
<code>rqstAttrName</code>	[Mandatory] Specifies the name of the requested attribute.

Example

This example illustrates the use of the `deleteRqstAttrForACS` command.

```
deleteRqstAttrForACS(1, rqstAttrName="empFirstName")
```

28.6 Exchanging Identity Federation Data

The integrated Identity Federation server supports the transport and receipt of request and response messages using either the Security Access Markup Language (SAML) 2.0 specifications, SAML 1.1, OpenID 2.0 or WS-Federation 1.1.

This section describes the following topics:

- [Using SAML 2.0](#)
- [Using SAML 1.1](#)
- [Using OpenID 2.0](#)
- [Using WS-Federation 1.1](#)



Note:

The specification describing how SAML might be used in a given context is referred to as a SAML profile. The specification describing how a SAML assertion and/or message is conveyed in, or transported over, another protocol is referred to as a SAML Binding.

28.6.1 Using SAML 2.0

SAML uses an eXtensible Markup Language (XML) framework to define a simple request-response protocol in order to achieve interoperability between vendor platforms that provide SAML assertions.

A SAML requester sends a SAML Request element to a responder. Similarly, a SAML responder returns a SAML Response element to the requester. Within the SAML 2.0 protocol, Identity Federation supports the functionality described in the following sections.

- [SAML 2.0 Bindings for SSO and Federation](#)
- [SAML 2.0 Bindings for Single Logout](#)
- [SAML 2.0 NameID Formats](#)
- [Securing SAML 2.0 Data](#)
- [OAM SAML 2.0 Supported Encryption Algorithms](#)
- [Changing Default Encryption Algorithm](#)
- [SAML 2.0 Service Details](#)

28.6.1.1 SAML 2.0 Bindings for SSO and Federation

SSO and Federation relies on SAML artifacts and assertions to relay authentication information.

The following bindings are supported for the exchange of data regarding SSO and federation.

- The HTTP Artifact Binding uses the Artifact Resolution Protocol and the SAML SOAP Binding (over HTTP) to resolve a SAML message by reference. The IdP will store the Assertion in its repository and redirect the user to the SP with a string (artifact) that references the stored Assertion. The SP will retrieve the Assertion by connecting to the IdP directly over SOAP/HTTP and presenting the artifact
- The HTTP POST Binding relies on an HTML form to communicate authentication information between providers. For example, the service provider may use HTTP Redirect to send a request while the identity provider uses HTTP POST to transmit the response. The IdP can also redirect the user to the SP in an HTML FORM that contains the Assertion itself.
- The Reverse SOAP binding (PAOS) is only supported when Access Manager is configured as an IdP. In this flow, the client sends a SOAP request containing a SAML 2.0 Authn Request message to the IdP. The IdP authenticates the user locally, and returns a SOAP response containing a SAML 2.0 Assertion. The client then presents the results to the remote SP.

28.6.1.2 SAML 2.0 Bindings for Single Logout

Single Logout defines how providers notify each other of logout events. This message exchange terminates all sessions when a logout occurs at the SP or IdP.

The following profiles are supported for exchanging data regarding single logout.

- The HTTP Redirect profile relies on HTTP redirects between providers. For example, the IdP redirects the user to the SP using a 302 redirect operation with the URL containing the Logout Request/Response message. This profile can be used for sending and receiving data regarding single logout.
- The HTTP POST profile occurs when the IdP redirects the user to the SP using an HTML FORM containing the Logout Request/Response message. This profile can be used for sending and receiving data regarding single logout.
- The SOAP Binding Profile allows the IdP to connect directly with the SP and send a Logout Request message. During logout, the IdP redirects the user to the various SPs in a sequential manner. The SP will respond with a Logout Response message. This profile relies on asynchronous SOAP over HTTP messaging calls between providers and can be used only for sending data regarding single logout.

28.6.1.3 SAML 2.0 NameID Formats

The Name Identifier Mapping defines how an SP can obtain name identifiers assigned to a principal that has authenticated in the name space of a different SP.

When a principal authenticated to one SP requests access to a second site, the second SP can use this protocol to obtain the name identifier and communicate with the first SP about the principal - even though no federation for the principal exists between them. The SAML 2.0 NameID formats listed in [Table 28-1](#) are supported in both IdP and SP mode.

Table 28-1 Supported SAML 2.0 NameID Formats

NameID Format	Description
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified	SP/IdP will use the applicable user attribute to populate/process the NameID value

Table 28-1 (Cont.) Supported SAML 2.0 NameID Formats

NameID Format	Description
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress	SP/IdP will use the applicable user attribute to populate/process the NameID value
urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName	SP/IdP will use the applicable user attribute to populate/process the NameID value
urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName	SP/IdP will use the applicable user attribute to populate/process the NameID value
urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos	SP/IdP will use the applicable user attribute to populate/process the NameID value
urn:oasis:names:tc:SAML:2.0:nameid-format:persistent	SP/IdP will use either: <ul style="list-style-type: none"> • A user attribute to populate the NameID value • A user attribute (such as DN) to set the value (same for every operations) • A random value and will store that value in the Federation Data Store (only this mode will require the use of a Federation Data Store)
urn:oasis:names:tc:SAML:2.0:nameid-format:transient	IdP will generate a random value
custom value	When this NameID format is used, OIF/IdP will use a user attribute to populate the NameID value

28.6.1.4 Securing SAML 2.0 Data

Regarding the security of identity data transported using the SAML 2.0 specifications, the following is true.

- All outgoing Assertions will be signed.
- All outgoing responses containing Assertions will not be signed.
- All outgoing requests/responses not containing Assertions will be signed.
- The signing certificate will not be included in the messages.
- Identity Federation (acting as the IdP) will not require signatures on any messages except when specified in the SP Partner metadata.
- NameIDs, attributes and Assertions will not be encrypted.
- Information on the default XML Encryption algorithm is located at <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/Overview.html#aes128-cbc>
- The hashing algorithm for signatures is SHA-1 by default. Identity Federation can be configured to use SHA-256.

28.6.1.5 OAM SAML 2.0 Supported Encryption Algorithms

OAM SAML 2.0 supports the following encryption modes:

- <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>

OAM SAML 2.0 also supports the following AES-GCM encryption modes:



This content applies only to OAM Bundle Patch 12.2.1.4.210920 and later releases.

- <http://www.w3.org/2009/xmlenc11#aes128-gcm>
- <http://www.w3.org/2009/xmlenc11#aes192-gcm>
- <http://www.w3.org/2009/xmlenc11#aes256-gcm>

28.6.1.6 Changing Default Encryption Algorithm

Perform the following steps to change the default encryption algorithm:

1. Navigate to `$MW_HOME/common/bin/` and run the `wlst.sh` command.
2. Run `connect()` to connect to WebLogic Admin Server. Provide the username and password when prompted.
3. Run `domainRuntime()` in the WLST shell
4. Run one of the following:
 - To set the default encryption algorithm at a Partner Profile level, run the following command:

```
putStringProperty("/fedpartnerprofiles/<PARTNER_PROFILE>/  
defaultencryptionmethod", "<ALGORITHM>")
```

where, `<PARTNER_PROFILE>` is the partner profile name, for example, `saml20-sp-partner-profile` and `<ALGORITHM>` is one of the OAM SAML 2.0 supported encryption algorithms, for example, `http://www.w3.org/2009/xmlenc11#aes256-gcm`

- To set the default encryption algorithm at the Partner level, run the following command:

```
updatePartnerProperty("<PARTNER>", "<PARTNER_TYPE>",  
"defaultencryptionmethod", "<ALGORITHM>", "string")
```

where, `<PARTNER>` is the partner name, for example, `AcmeSP`,

`<PARTNER_TYPE>` is the one of the following type of partner: `sp` or `idp`, and

`<ALGORITHM>` is one of the OAM SAML 2.0 supported encryption algorithms, for example, `http://www.w3.org/2009/xmlenc11#aes256-gcm`

5. Exit the WLST environment by running `exit()`

28.6.1.7 SAML 2.0 Service Details

The SAML 2.0 Metadata for the IdP and SP is contained in a single XML document and can be retrieved using the Oracle Access Management Console.

The Metadata can also be retrieved by accessing either of the following URLs:

```
http://public-oam-host:public-oam-port/oamfed/idp/metadata
http://public-oam-host:public-oam-port/oamfed/sp/metadata
```

The certificates used for signature and encryption operations are published via the SAML 2.0 Metadata. The certificates can be retrieved by using a Service URL that specifies the Key ID of the key/certificate entry as defined in the Keystore Settings. (See [Defining Keystore Settings for Federation](#).) For example,

```
http://public-oam-host:public-oam-port/oamfed/idp/cert?id=osts_signing
```

The Provider ID and the Issuer ID of the IdP and SP profiles are identical and can be retrieved from the applicable Provider Partner profile using the Oracle Access Management Console.

[Table 28-2](#) documents the SAML 2.0 URLs for use when Identity Federation is configured to act as an IdP.

Table 28-2 SAML 2.0 URLs for Identity Federation Acting As Identity Provider

Description	URL
Single Sign On Service URL for HTTP Redirect binding	http://public-oam-host:public-oam-port/oamfed/idp/samlv20
Single Sign On Service URL for HTTP POST binding	http://public-oam-host:public-oam-port/oamfed/idp/samlv20
Single Sign On Service URL for SOAP binding	http://public-oam-host:public-oam-port/oamfed/idp/soap
Artifact Resolution Service URL for SOAP binding	http://public-oam-host:public-oam-port/oamfed/idp/soap
Single Logout Service URL for HTTP Redirect binding	http://public-oam-host:public-oam-port/oamfed/idp/samlv20
Single Logout Service URL for HTTP POST binding	http://public-oam-host:public-oam-port/oamfed/idp/samlv20
Attribute Authority Service URL for SOAP binding	http://public-oam-host:public-oam-port/oamfed/aa/soap

[Table 28-3](#) documents the SAML 2.0 URLs for use when Identity Federation is configured to act as an SP.

Table 28-3 SAML 2.0 URLs for Identity Federation Acting as Service Provider

Description	URL
Assertion Consumer Service URL for Artifact binding	http://public-oam-host:public-oam-port/oam/server/fed/sp/ssso

Table 28-3 (Cont.) SAML 2.0 URLs for Identity Federation Acting as Service Provider

Description	URL
Assertion Consumer Service URL for HTTP POST binding	http://public-oam-host:public-oam-port/oam/server/fed/sp/sso
Single Logout Service URL for HTTP Redirect binding	http://public-oam-host:public-oam-port/oamfed/sp/samlv20
Single Logout Service URL for HTTP POST binding	http://public-oam-host:public-oam-port/oamfed/sp/samlv20

28.6.2 Using SAML 1.1

Although the standards address the same use case, SAML 2.0 and SAML 1.1 get there in different ways. The most important type of SAML 1.1 request is a query.

A SP makes a query directly to an IdP over a secure back channel (using SOAP). Within the SAML 1.1 protocol, Identity Federation supports the features described in the following sections.

- [SAML 1.1 Profiles for Web Browser SSO](#)
- [SAML 1.1 Logout Profile](#)
- [SAML 1.1 NameID Formats](#)
- [About SAML 1.1 Data Security](#)
- [SAML 1.1 Service Details](#)

28.6.2.1 SAML 1.1 Profiles for Web Browser SSO

SAML 1.1 profiles rely on pushing SAML artifacts and assertions to an SP to relay authentication information.

The following profiles are supported.

- The Browser/Artifact Profile passes a SAML assertion from the IdP to the SP by reference (through the browser using HTTP Redirect). This artifact is subsequently dereferenced through a back-channel exchange in which the SP retrieves the assertion from the IdP using SAML over SOAP over HTTP.
- The Browser/POST Profile passes an SSO assertion to an SP through the browser using HTTP POST. We say that the identity provider "pushes" the assertion to the service provider.

28.6.2.2 SAML 1.1 Logout Profile

The SAML 1.1 specifications do not define a logout profile thus Identity Federation is not able to notify remote partners regarding a user logging out.

28.6.2.3 SAML 1.1 NameID Formats

When a principal authenticated to one SP requests access to a second site, the second SP can obtain the name identifier and communicate with the first SP regarding the principal - even though no federation for the principal exists between them.

The SAML 1.1 NameID formats listed in [Table 28-4](#) are supported in both IdP and SP mode.

Table 28-4 Supported SAML 1.1 NameID Formats

NameID Format	Description
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified	SP/IdP will use the applicable user attribute to populate/process the NameID value
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress	SP/IdP will use the applicable user attribute to populate/process the NameID value
urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName	SP/IdP will use the applicable user attribute to populate/process the NameID value
urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName	SP/IdP will use the applicable user attribute to populate/process the NameID value
urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos	SP/IdP will use the applicable user attribute to populate/process the NameID value
custom value	When this NameID format is used, OIF/IdP will use a user attribute to populate the NameID value

28.6.2.4 About SAML 1.1 Data Security

Regarding the security of identity data transported using the SAML 1.1 specifications, the following is true.

- All outgoing Assertions will be signed.
- All outgoing responses containing Assertions will not be signed.
- The signing certificate will not be included in the messages.
- Identity Federation (acting as the IdP) will not require signatures on any messages.
- The hashing algorithm for signatures is SHA-1 by default. Identity Federation can be configured to use SHA-256.

28.6.2.5 SAML 1.1 Service Details

The certificates used for signature and encryption operations can be retrieved by using a Service URL that specifies the Key ID of the key/certificate entry as defined in the Keystore Settings.

For more information, see [Defining Keystore Settings for Federation](#).

For example,

```
http://public-oam-host:public-oam-port/oamfed/idp/cert?id=osts_signing
```

The Provider ID and the Issuer ID of the IdP and SP profiles are identical and can be retrieved from the applicable Provider Partner profile using the Oracle Access Management Console.

[Table 28-5](#) documents the SAML 1.1 URLs for use when Identity Federation is configured to act as an IdP.

Table 28-5 SAML 1.1 URLs for Identity Federation Acting As Identity Provider

Description	URL
Single Sign On Service URL	http://public-oam-host:public-oam-port/oamfed/idp/samlv11sso
Artifact Resolution Service URL	http://public-oam-host:public-oam-port/oamfed/idp/soapv11

[Table 28-6](#) documents the SAML 1.1 URL for use when Identity Federation is configured to act as an SP.

Table 28-6 SAML 1.1 URL for Identity Federation Acting as Service Provider

Description	URL
Assertion Consumer Service URL	http://public-oam-host:public-oam-port/oam/server/fed/sp/sso

28.6.3 Using OpenID 2.0

OpenID 2.0 allows users to create accounts with a preferred OpenID IdP and use the account as the basis for signing on to any website that accepts OpenID authentication.

Identity data is communicated through the exchange of an OpenID identifier (a URL or XRI chosen by the end-user) and the IdP provides OpenID authentication. Within the OpenID protocol, Identity Federation supports the functionality described in the following sections.

- [OpenID 2.0 Authentication/SSO](#)
- [OpenID 2.0 Logout](#)
- [OpenID 2.0 NameID Format](#)
- [About OpenID 2.0 Data Security](#)
- [OpenID 2.0 Extensions](#)
- [OpenID 2.0 Service Details](#)

28.6.3.1 OpenID 2.0 Authentication/SSO

OpenID 2.0 allows a user to sign into a new web site using a special OpenID URL. For example, if you have a blog at myblog.com, you might have created the OpenID URL, yourname.myblog.com. Then if you navigate to a second web site that accepts OpenID logins and click on the OpenID button, you can type in the URL and click to log in. The second SP discovers the OpenID IdP URL with this OpenID identifier. When the OpenID IdP redirects the authenticated user to the SP, it includes the OpenID Assertion which contains the result of the operation, the NameID of the user and (optional) attributes.

28.6.3.2 OpenID 2.0 Logout

The OpenID 2.0 specifications do not define a logout profile thus Identity Federation is not able to notify remote partners regarding a user logging out.

28.6.3.3 OpenID 2.0 NameID Format

OpenID defines the NameID as being a random string thus Identity Federation will use one of the following as the value for the NameID.

- A hashed user attribute (such as DN)
- A generated, random value that will be stored in the Federation Data Store; this mode requires the use of a Federation Data Store

28.6.3.4 About OpenID 2.0 Data Security

Regarding the security of identity data transported using the OpenID 2.0 specifications, the following is true.

- All outgoing Assertions will be signed.
- The default Association Algorithm is HMAC SHA-1.
- The default Session Agreement Algorithm is Diffie-Hellmann SHA-1.

28.6.3.5 OpenID 2.0 Extensions

OpenID is an extensible specification. The following extensions are available when using the integrated Identity Federation.

- Attribute Exchange (AX): If enabled, a SP can request attributes to be included in the OpenID Assertion response. The IdP can include the requested attributes or attributes configured to be in the response. (Default: enabled)
- Provider Authentication Policy Extension (PAPE): If enabled, advanced authentication methods can be defined and specified. This might include, for example, a phishing-resistant authentication method or multi-factor authentication. (Default: disabled)
- GSA Level 1: identifier in the OpenID Assertion indicating if this server is compliant with the <http://www.idmanagement.gov/schema/2009/05/icam/openid-trust-level1.pdf> policy. If enabled and if PAPE is enabled, OIF will include this policy in the OpenID response (Default: disabled)
- Level Of Assurance (LOA): identifier in the OpenID Assertion indicating if this server is compliant with the http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf policy. If enabled, OIF/IdP will use the mapping between Level of Assurance and schemeID to determine the value to use for LOA in the OpenID response (see 2.5.2 for more information) (Default: disabled)
- No Private Identifier Information (NoPII): identifier in the OpenID Assertion indicating if this server is compliant with the <http://www.idmanagement.gov/schema/2009/05/icam/no-pii.pdf> policy. Note, if enabled, OIF will not include attributes in the OpenID Assertion
- Persistent Personal Identifier (PPID): identifier in the OpenID Assertion indicating if this server is compliant with the <http://schemas.xmlsoap.org/ws/2005/05/identity/>

claims/privatepersonalidentifier policy. If enabled and if PAPE is enabled, OIF will include this policy in the OpenID response (Default: disabled)

- Registration (SReg): extension for attributes in the OpenID Assertion. If enabled, the SP can request attributes to be included in the response, and the IdP can include requested attributes or attributes configured to be in the response (Default: disabled)
- UI Extension (UIExt): extension for UI. In OIF/IdP, support for this extension is limited to advertisement in the XRDS metadata (Default: disabled)

28.6.3.6 OpenID 2.0 Service Details

The following URL is the realm of the OpenID 2.0 SP component.

```
http://public-oam-host:public-oam-port
```

[Table 28-7](#) documents the OpenID 2.0 URLs for use when Identity Federation is configured to act as an IdP.

Table 28-7 OpenID 2.0 URLs for Identity Federation Acting As Identity Provider

Description	URL
Single Sign On Service URL	http://public-oam-host:public-oam-port/oamfed/idp/openidv20
Discovery Service URL	http://public-oam-host:public-oam-port/oamfed/idp/openidv20

[Table 28-8](#) documents the OpenID 2.0 URLs for use when Identity Federation is configured to act as an SP.

Table 28-8 OpenID 2.0 URLs for Identity Federation Acting as Service Provider

Description	URL
Single Sign On Service URL	http://public-oam-host:public-oam-port/oam/server/fed/sp/sso
Discovery Service URL	http://public-oam-host:public-oam-port/oamfed/sp/openidv20
Realm URL	http://public-oam-host:public-oam-port

28.6.4 Using WS-Federation 1.1

Access Manager now supports features of the WS-Federation 1.1 protocol.

WS-Federation 1.1 partners can be created using the new `addWSFed11IdPFederationPartner` and `addWSFed11SPFederationPartner` WLST commands. After creating the partners, the profiles can be configured using the existing WLST Identity Federation commands. For details, see the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference for Identity and Access Management*.

 **Note:**

Partial administration of the WS-Federation 1.1 partners is available using the Oracle Access Management Console.

28.7 Administering Identity Federation

Identity Federation integrated with Access Manager can be administered with a combination of configurations using the Oracle Access Management Console and Oracle WebLogic Scripting Tool (WLST) commands.

Use the Oracle Access Management Console to enable the Identity Federation service, manage IdP and SP partner profiles, and work with federated authentication schemes and policies. Use the WLST utilities to manage additional server and partner configuration properties.

 **Note:**

Not all WLST command functionality is duplicated in the Oracle Access Management Console and not all console functionality is duplicated on the command line.

The Oracle Access Management Console enables Administrators to manage configuration related to the federation service and partners. [Table 28-9](#) summarizes the types of information that you can configure for Identity Federation using Oracle Access Management Console.

Table 28-9 Configuring Identity Federation Settings

Configuring ...	Description
Federation Administrators	Administrators who can manage federated partners and related configuration.
Federation Service	Enable and disable the Identity Federation service in Access Manager. See " Enabling Identity Federation ".
Federation Settings	Manage basic Identity Federation service configuration properties. See Managing Settings for Identity Federation .
Providers for Federation	IdP partners are managed within the context of administering Identity Federation as a SP. Conversely, SP partners are managed within the context of administering Identity Federation as an IdP. See Administering Identity Federation As A Service Provider or Administering Identity Federation As An Identity Provider ..
Authentication Schemes and Modules for Federation	Manage federation authentication schemes. See " Using Authentication Schemes and Modules for Identity Federation ".
Policies for Use with Federation	Manage policies for use with federation partners. See " Managing Access Manager Policies for Use with Identity Federation ".

Table 28-10 outlines the tasks required to implement identity federation using the Oracle Access Management Console.

Table 28-10 Implementing Identity Federation

Task	Reference
Enable the Identity Federation service.	Enabling Identity Federation
Configure federation settings.	Managing General Federation Settings
Identity IdP and/or SP partners, and configure attributes for them.	Administering Identity Federation As A Service Provider
Configure an authentication or authorization policy.	Managing Federation Schemes and Policies
Protect a resource with this policy.	Managing Policies to Protect Resources and Enable SSO

28.8 Enabling Identity Federation

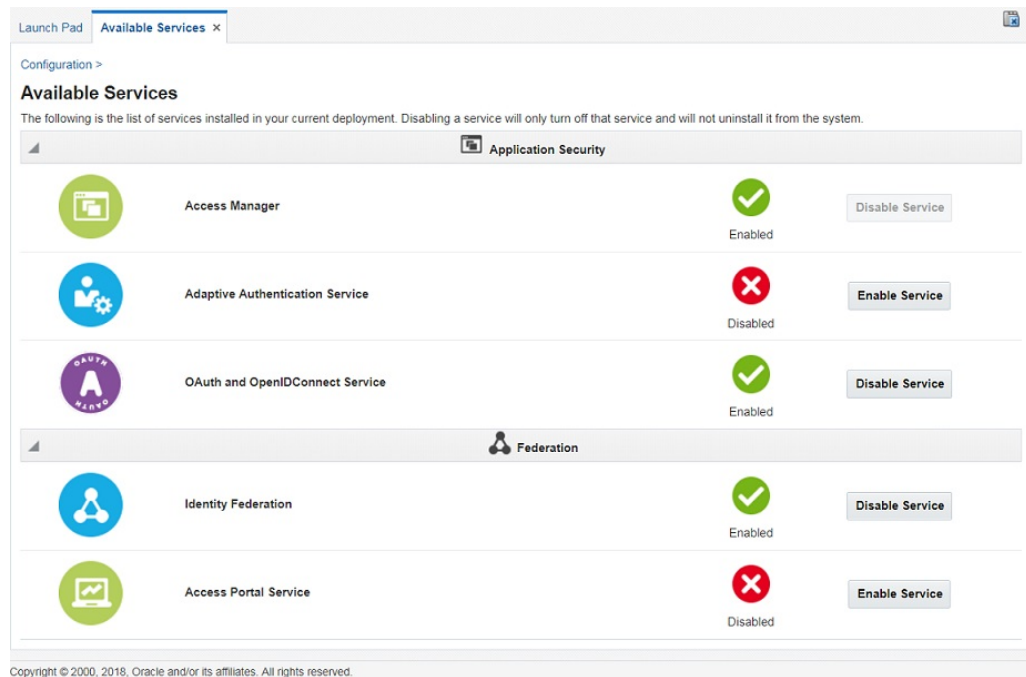
Identity Federation is an authentication module in Oracle Access Management so both the Access Manager service and Identity Federation must be enabled.

Figure 28-1 illustrates the Available Services page in Oracle Access Management Console with the Access Manager service and Identity Federation already enabled. Use this page to enable (or disable) Identity Federation together with the Access Manager service.

Note:

Once enabled, it is possible to enable or disable specific Federation features such as IdP, SP, Attribute Authority and/or Attribute Requester. Use the `configureFederationService()` WLST command as documented in WebLogic Scripting Tool Command Reference for Identity and Access Management

Figure 28-1 Available Services Page



To enable the Identity Federation service with Access Manager

1. Log in to the Oracle Access Management Console.
`https://hostname:port/oamconsole/`
2. From the Welcome page, under **Configuration**, click **Available Services**.
3. **Enable Identity Federation:** Click **Enable** beside Identity Federation (or confirm that the green Status check mark displays).
A Confirmation window is displayed.
4. Click OK.
5. **Enable Access Manager:** Click **Enable** beside Access Manager (or confirm that the green Status check mark displays).
A Confirmation window is displayed.
6. Click OK.

Managing Identity Federation Partners

You need to familiarize yourself with the concept of federation partners (service providers and identity providers) in Oracle Access Management Identity Federation.

The following topics describe how to manage identity federation partners:

- [Understanding Federation And Partners](#)
- [Managing Federation Partners](#)
- [Administering Identity Federation As A Service Provider](#)
- [Administering Identity Federation As An Identity Provider](#)
- [Using Attribute Mapping Profiles](#)
- [Mapping Federation Authentication Methods to Access Manager Authentication Schemes](#)
- [Using the Attribute Sharing Plug-in for the Attribute Query Service](#)
- [Using the Federation Proxy](#)
- [Using WLST for Identity Federation Administration](#)
- [Using WLST for Key Transport Algorithm](#)

29.1 Understanding Federation And Partners

You must have already familiarized with the federation and partner concepts before you start to manage identity federation partners.

You must have completed the following task:

See [Enabling Identity Federation](#).

The integrated Identity Federation server supports the transport and receipt of request and response messages using either the Security Access Markup Language (SAML) 2.0 specifications, SAML 1.1, OpenID 2.0 or WS-Federation 1.1. Thus, Identity Provider (IdP) and Service Provider (SP) partners can be created with any of these protocols defined. SAML and OpenID partners can be defined using the Oracle Access Management Console. WS-Federation partners can be created using WLST commands.

See [Creating Remote Identity Provider Partners](#).

See [Creating Remote Service Provider Partners](#).

See *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference for Identity and Access Management*.

29.2 Managing Federation Partners

This 11g Release 2 (11.1.2.2) of the integrated Identity Federation provides the ability to be configured as a Service Provider (SP) or an Identity Provider (IdP). Following this provider definition, remote providers (whether service or identity) partnered in Federation SSO need to

be managed as well. Towards this end, Identity Federation developed the configuration hierarchy concepts of a *partner* and a *partner profile*.

- A *partner profile* refers to settings specific to a partner type (IdP or SP) or a protocol version (SAML 2.0, SAML 1.1, OpenID 2.0). It is a configuration group that represents a sets of common properties that apply to all partners that reference it. It contains mostly secondary configuration objects such as Authentication Method mappings, cryptographic settings (SHA-1 vs SHA-256) and the like.
- A *partner* refers to the configuration for a specific organization partnered in the Federation SSO process. Each partner is associated with a partner profile. The `partnerprofileid` property in a Partner entry defines the partner profile to which this partner is assigned. If the `partnerprofileid` property is not defined, the default Partner Profile for the Partner (based on the Partner type and the Partner protocol) will be used.

All Partners associated with the same Partner Profile will share its defined settings unless they are specifically overridden for a partner at the Partner configuration level. A Partner configuration overrides a Partner Profile configuration which, in turn, overrides a global configuration.

Partner profiles are only manageable using WLST commands. Each new partner created will be bound to one of the default partner profiles listed in [Table 29-1](#). To assign a new partner profile to a partner, use the `setFedPartnerProfile()` WLST command after creating the partner.

See [Using WLST for Identity Federation Administration](#).

Table 29-1 Default Partner Profiles

Default Partner Profile	Description
saml20-idp-partner-profile	SAML 2.0 Partner Profile for IdP partners
saml20-sp-partner-profile	SAML 2.0 Partner Profile for SP partners
saml11-idp-partner-profile	SAML 1.1 Partner Profile for IdP partners
saml11-sp-partner-profile	SAML 1.1 Partner Profile for SP partners
openid20-idp-partner-profile	OpenID 2.0 Partner Profile for IdP partners
openid20-sp-partner-profile	OpenID 2.0 Partner Profile for SP partners

29.3 Administering Identity Federation As A Service Provider

When the integrated Identity Federation is configured as an SP, you must define any remote IdP partners as trusted by creating and managing profiles that contain details regarding each remote IdP.

To begin administration of the integrated Identity Federation server as an SP, click the Service Provider Administration link under Identity Federation from the Launch Pad in the Oracle Access Management Console. This section provides the following topics.

- [Creating Remote Identity Provider Partners](#)
- [Managing the Remote Identity Provider Partners](#)

29.3.1 Creating Remote Identity Provider Partners

Use the New Identity Provider Page to define an identity provider (IdP) partner record for Access Manager. You can specify service details manually or load them from a metadata file.

Figure 29-1 shows the Create Identity Provider Partner page when service details are configured by loading an XML metadata file.

Figure 29-1 New Identity Provider Page, Service Details Loaded from Metadata

The screenshot shows the 'Create Identity Provider Partner' page with the following sections:

- General:** Includes a text field for 'Name', a text area for 'Description', and two checkboxes: 'Enable Partner' (checked) and 'Default Identity Provider Partner' (checked).
- Service Information:** Includes a 'Protocol' dropdown menu set to 'SAML2.0', 'Service Details' radio buttons for 'Load from provider metadata' (selected) and 'Enter Manually', and a 'Metadata File' section with a 'Browse...' button and the text 'No file selected'.
- Mapping Options:**
 - User Mapping:** Includes a 'User Identity Store' dropdown, a 'User Search Base DN' text field, and three radio button options:
 - 'Map assertion Name ID to User ID Store attribute' (selected): Includes a 'Map assertion Name ID to User ID Store attribute' text field with the value 'mail'.
 - 'Map assertion attribute to User ID Store attribute': Includes 'Assertion Attribute' and 'User ID Store Attribute' text fields.
 - 'Map assertion to user record using LDAP query': Includes an 'LDAP Query' text area.
 - Attribute Mapping:** Includes an 'Attribute Profile' text field with the value 'idp-attribute-profile' and search and add icons.

Figure 29-2 shows the Create Identity Provider Partner page when service details are configured by entering values manually.

Figure 29-2 New Identity Provider Page, Service Details entered Manually

Create Identity Provider Partner Identity Provider Partner Save

General

* Name

Description

Enable Partner

Default Identity Provider Partner

Service Information

Protocol

Service Details Load from provider metadata Enter Manually

* Provider ID

Succinct ID

* SSO Service URL

SOAP Service URL

Logout Request Service URL

Logout Response Service URL

Signing Certificate

* Load Signing Certificate No file selected.

Load Encryption Certificate No file selected.

Mapping Options

User Mapping

User Identity Store

User Search Base DN

Map assertion Name ID to User ID Store attribute

* Map assertion Name ID to User ID Store attribute

Map assertion attribute to User ID Store attribute

Assertion Attribute

User ID Store Attribute

Map assertion to user record using LDAP query

LDAP Query

Attribute Mapping

Attribute Profile

Table 29-2 describes each element on the New Identity Provider page.

Table 29-2 Identity Provider Partner Settings

Element	Description
Name	This is the provider name.
Description	This is a brief description of the provider. (Optional).
Protocol	This is the provider protocol (SAML 1.1, SAML 2.0 and so on).
Service Details	This drop-down enables you to choose whether to enter service details manually or load from metadata.
Metadata File	This field appears if loading metadata from a file. Click Browse to select a file to use. Applies to SAML 2.0 only.
Issuer ID	This is the issuer ID of the provider. Applies to SAML 2.0 and SAML 1.1 only.

Table 29-2 (Cont.) Identity Provider Partner Settings

Element	Description
Succinct ID	This is the succinct ID of the provider. This element is required if using the artifact profile. Applies to SAML 2.0 and SAML 1.1 only.
SSO Service URL	This is the URL address to which SSO requests are sent.
SOAP Service URL	This is the URL address to which a SOAP service request is sent. This element is required if using artifact profile.
Logout Request Service URL	This is the URL address to which a logout request is sent by the provider. This element is required if using the logout feature. Applies to SAML 2.0 only.
Logout Response Service URL	This is the URL address to which a logout response is sent. This element is required if using the logout feature. Applies to SAML 2.0 only.
Signing Certificate	This is the signing certificate used by the provider. You can specify it in <code>pem</code> and <code>der</code> formats. Applies to SAML 2.0 and SAML 1.1 only.
User Identity Store	This is the identity store in which the IdP's users will be located and mapped. Identity Federation supports multiple identity stores, defined on a per-partner basis. Optionally, if no user identity store is selected, the default Access Manager store is used.
User Search Base DN	This is the base search DN used when looking up user records. (Optional.) If omitted, the default user search base DN configured for the selected user identity store is used.)
Mapping Option	<p>This setting indicates how an incoming assertion is mapped to a user in the identity store. Select one of the following:</p> <ul style="list-style-type: none"> • Map Assertion Name ID to User ID Store Attribute Enter the identity store attribute to which the assertion NameID will be mapped. • Map Assertion Attribute to User ID Store Attribute Enter assertion attribute and the identity store attribute to which it will be mapped. • Map Assertion to User Record Using LDAP Query Enter an LDAP query with placeholders for incoming data. You may use: <ul style="list-style-type: none"> - an attribute from the SAML assertion's <code>AttributeStatement</code> element, referenced by its name prefixed and suffixed with the <code>%</code> character - the SAML assertion subject's NameID, referenced by <code>%fed.nameidvalue%</code> - the identity provider's partner name, referenced by <code>%fed.partner%</code>. <p>For example, an LDAP query to map an incoming assertion based on two assertion attributes (lastname and email) would be <code>(&(sn=%lastname%)(mail=%email%))</code>.</p>
Enable Basic HTTP Authentication	Check this box to accept HTTP basic credentials. (Advanced element, available only in provider Edit mode.)
Attribute Mapping Profile	Indicates the attribute profile to which the partner is bound.

Table 29-2 (Cont.) Identity Provider Partner Settings

Element	Description
Service Details	Indicates which of the following options Identity Federation (the RP) uses to perform Federation SSO with the IdP. Applies to OpenID 2.0 only. <ul style="list-style-type: none"> By discovering the IdP SSO URLs via the IdP XRDS metadata available at the Discovery Service URL. By using the specified static OpenID login endpoint which is the IDP SSO service URL.
Discovery URL	Defines the location where the IdP publishes its XRDS metadata. Applies to OpenID 2.0 only.
Endpoint URL	Defines the IdP SSO Service location. Applies to OpenID 2.0 only.
Enable Global Logout	Indicates whether or not Identity Federation should notify the remote partner when the user is signing off during the logout flow. Applies to SAML 2.0 only.
HTTP POST SSO Response Binding	Indicates whether the SAML Assertion should be sent back from the IdP using the HTTP POST Binding or the Artifact Binding. Applies to SAML 2.0 only.
Authentication Request NameID Format	Indicates the NameID format that Identity Federation will request from the IdP during the Federation SSO operation. If none is selected, a NameID format is not specified in the request. Applies to SAML 2.0 only.

29.3.1.1 Defining a New SAML 2.0 Identity Provider for Federation

You can define a new SAML 2.0 identity provider (IdP) for federation.

To create a new identity provider:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, select **Create Identity Provider Partner** from the **Create (+)** drop-down list in the **Federation** section.
3. In the **Service Details** field, select **Load from provider metadata**. (SAML 2.0 is typically configured with metadata.)
4. A new field named **Metadata File** appears. Click **Browse**.
5. Select the metadata file of interest.
6. The metadata is loaded from the file.
7. Click **Save** to create the Identity Provider definition.

29.3.1.2 Defining a New SAML 1.1 Identity Provider for Federation

You can define a new SAML 1.1 identity provider (IdP) for federation.

To create a new identity provider:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.

2. In the Federation console, select **Create Identity Provider Partner** from the **Create (+)** drop-down list in the **Federation** section.
3. In the **Service Details** field, select **Enter Manually**.
4. Fill in the New Identity Provider page using values for your environment (). The information you provide depends on the protocol chosen for the provider and other factors.
See [Table 29-2](#).
5. Click **Save** to create the identity provider definition.

 **Note:**

Some SAML 1.1 configuration parameters are not exposed through the Oracle Access Management Console. The values of these parameters can be modified using the `updatePartnerProperty WLST` command.

See *updatePartnerPropertyWLST Command Reference for WebLogic Server* guide.

29.3.1.3 Defining a New OpenID 2.0 Identity Providers for Federation

From 11g Release 2 (11.1.2.3) the Identity Federation supports OpenID, and acts as an OpenID RP/SP. OpenID Providers can be registered as IdP partners.

Authentication schemes created using these OpenID partners protect Access Manager resources using authentication services provided by the OpenID identity providers.

To define a new OpenID 2.0 identity provider (IdP) for federation:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, select **Create Identity Provider Partner** from the **Create (+)** drop-down list in the **Federation** section.
3. Fill in the values appropriate for your environment either manually or by uploading a metadata file.

The information you provide depends on the protocol chosen for the provider and other factors.

4. Click **Save** to create the identity provider definition.

Google IdP Partners

To add Google as an OpenID 2.0 IdP.

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, select **Create Identity Provider Partner** from the **Create (+)** drop-down list in the **Federation** section.
3. From the Launch Pad, click Service Provider Administration under Identity Federation.
4. Select OpenID 2.0 from the **Protocol** drop down menu.
5. Select **Google provider default settings** from the **Service Details** drop down menu.
6. Click **Save** to create the identity provider definition.

The partner is configured so that the SP requests the assertion attributes from the Google IdP and maps them to the corresponding session attribute names:

See [Table 29-3](#).

Table 29-3 Attributes for Google OpenID Partner

Assertion Attribute Name	Session Attribute Name
http://axschema.org/contact/country/home	country
http://axschema.org/contact/email	email
http://axschema.org/namePerson/first	firstname
http://axschema.org/pref/language	language
http://axschema.org/namePerson/last	lastname

The Google partner uses `mail` as the user mapping attribute, so that an incoming `http://axschema.org/contact/email` attribute should match the `mail` attribute of the user in the user identity store.

Yahoo IdP Partners

To add Yahoo as an OpenID 2.0 IdP:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, select **Create Identity Provider Partner** from the **Create (+)** drop-down list in the **Federation** section.
3. Select OpenID 2.0 from the **Protocol** drop down menu.
4. Select **Yahoo provider default settings** from the **Service Details** drop down menu.
5. Click **Save** to create the identity provider definition.

The partner is configured so that the SP requests the assertion attributes from the Yahoo IdP and maps them to the corresponding session attribute names:

See [Table 29-4](#).

Table 29-4 Attributes for Yahoo OpenID Partner

Assertion Attribute Name	Session Attribute Name
http://axschema.org/contact/country/home	country
http://axschema.org/contact/email	email
http://axschema.org/namePerson/first	firstname
http://axschema.org/pref/language	language
http://axschema.org/namePerson/last	lastname

The yahoo partner uses `mail` as the user mapping attribute, so that an incoming `http://axschema.org/contact/email` attribute should match the `mail` attribute of the user in the user identity store.

29.3.1.4 Enabling OpenID Simple Registration

By default, Identity federation uses the Attribute Exchange extension to obtain user identity attributes from an OpenID IdP.

However, if you need to use the older Simple Registration (SREG) extension, you can enable it by running the following WLST commands:

```
putBooleanProperty("/spglobal/openid20axenabled", "false")
putBooleanProperty("/spglobal/openid20sregenabled", "true")
```

29.3.1.5 Disabling OpenID Simple Registration

You can disable Simple Registration to Attribute Exchange extension.

To switch from the Simple Registration (SREG) extension to the Attribute Exchange extension to obtain user identity attributes from an OpenID IdP:

```
putBooleanProperty("/spglobal/openid20axenabled", "true")
putBooleanProperty("/spglobal/openid20sregenabled", "false")
```

29.3.2 Managing the Remote Identity Provider Partners

The following topics describe how to manage an existing IdP for Identity Federation.

- [Searching for Existing Identity Providers](#)
- [Updating Identity Providers for Federation](#)

29.3.2.1 Searching for Existing Identity Providers

You can search for existing identity providers from the Federation console.

To search:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, click **Identity Provider Management** in the **Federation** section.
3. In the Search section of the page, enter appropriate search criteria for identity provider(s). The characters "*" (asterisk) and "." (period) are supported as search wildcards.

See [Table 29-5](#) for details about the search parameters.

4. Click **Search**.
5. The search results are displayed in a table.

Table 29-5 Elements Used for IdP Provider Search

Element	Description
Partner Name	Searches for a specific partner name.
Provider ID	Searches by provider ID.
Status	Searches providers matching a status.

Table 29-5 (Cont.) Elements Used for IdP Provider Search

Element	Description
Description	Searches by provider description.
Protocol	Searches for providers that use a specified protocol.

Table 29-5 describes the parameters by which providers can be searched.

Figure 29-3 Searching for Identity Providers

Use the search tool to find your Service Provider partner or register a new partner using the Create Service Provider Partner button.

[+ Create Service Provider Partner](#)

Search

Partner Name Provider ID

Status Protocol

Description

[Search](#) [Reset](#)

Search Results

Actions View [+ Create](#) [Duplicate](#) [Edit](#) [Delete](#) [Detach](#)

Row	Partner Name	Status	Provider ID	Protocol	Description
No data to display.					
					Number of Rows

29.3.2.2 Updating Identity Providers for Federation

You can search for Identity Providers for Federation and update providers information.

To update Identity Providers for Federation:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, click **Identity Provider Management** in the **Federation** section.
3. Search for the provider you wish to update.
See [Searching for Existing Identity Providers](#).
4. Select the provider of interest from the search results table.
5. Click the pencil icon to display the provider update page. The page is divided into sections for: Service Information, Signing Certificates, User Mapping, and Advanced.
6. Update the provider information.
See [Table 29-2](#) for details.
7. Click **Save** to update the Identity Provider definition.

29.4 Administering Identity Federation As An Identity Provider

When the integrated Identity Federation is configured as an IdP, you must define any remote SP partners as trusted by creating and managing profiles that contain details regarding each remote SP.

This section provides the following topics.

- [Creating Remote Service Provider Partners](#)
- [Managing the Remote Service Provider Partners](#)

29.4.1 Creating Remote Service Provider Partners

Use the Service Provider Partner page to define a partner profile when Identity Federation is configured as an IdP. You can specify service details manually or load them from a metadata file.

To create remote service provider partners:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, select **Create Service Provider Partner** from the **Create (+)** drop-down list in the **Federation** section.
3. Enter values for the parameters.

[Table 29-6](#) describes each element on the Create Service Provider page.

Table 29-6 Service Provider Partner Settings

Element	Description
Name	This is the provider name.
Enable Partner	Select whether this partner is currently participating in the federation.
Description	This is a brief description of the provider. (Optional).
Protocol	This is the provider protocol (SAML 1.1, SAML 2.0 or OpenID 2.0).
Service Details	Select whether to enter service details manually or load from metadata. If selecting the latter, browse for the metadata file. Applies to SAML 2.0 only.
Metadata File	This field appears if loading metadata from a file. Click Browse to select a file to use. Applies to SAML 2.0 only.
Provider ID	The provider ID or issuer ID of the remote Service Provider. Applies to SAML 2.0 and SAML 1.1 only.
Assertion Consumer URL	A URL to which Assertion responses are sent. Applies to SAML 2.0 and SAML 1.1 only.
Load Signing Certificate	Upload the signing certificate used by this SP. Only visible when Enter Manually is selected. Applies to SAML 2.0 and SAML 1.1 only.
Logout Request URL	A URL to which logout requests are sent. Applies to SAML 2.0 only.
Logout Response URL	A URL to which responses to logout requests are sent. Applies to SAML 2.0 only.
Load Encryption Certificate	Upload the encryption certificate used by this SP. Only visible when Enter Manually is selected. Applies to SAML 2.0 only.

Table 29-6 (Cont.) Service Provider Partner Settings

Element	Description
NameID Format	Indicates which NameID format should be used for this SP. Applies to SAML 2.0 and SAML 1.1 only. See Using SAML 2.0 . See Using SAML 1.1 .
NameID Value	Indicates how to populate the NameID Value. Applies to SAML 2.0 and SAML 1.1 only. <ul style="list-style-type: none"> • If User ID Store Attribute is selected, specify the user attribute to be used. • If Expression is specified, enter the expression to be used
Attribute Mapping Profile	Indicates the attribute mapping profile to which the partner is bound. Applies to SAML 2.0 and SAML 1.1 only.
User Identity Store	This is the identity store in which the IdP's users will be located and mapped. Identity Federation supports multiple identity stores, defined on a per-partner basis. If no user identity store is selected, the default store defined for Access Manager is used.
User Search Base DN	This is the base search DN used when looking up user records. (Optional. If omitted, the default user search base DN configured for the selected user identity store is used.)
Enable Global Logout	Indicates whether or not OIF should notify the remote partner when the user is signing off, during the logout flow. Applies to SAML 2.0 only.
SSO Response Binding	Indicates whether the SAML Assertion should be sent back from the IdP using the HTTP POST Binding or the Artifact Binding, Applies to SAML 2.0 and SAML 1.1 only.
Encrypt Assertion	Indicates whether or not the Assertion should be encrypted for this partner. Applies to SAML 2.0 only.
Realm	The URL identifying an OpenID SP. Applies to OpenID 2.0 only.
Endpoint URL	The URL to which the IdP will redirect the user with the OpenID Assertion. Applies to OpenID 2.0 only.

4. Click Save to create the remote SP partner profile.

29.4.2 Managing the Remote Service Provider Partners

You can edit and manage the profiles of remote SP partners, search for the profile and make changes to the attribute values.

To search for existing service provider partner profiles:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, click **Service Provider Management** in the **Federation** section.
3. In the **Search** section of the page, enter appropriate search criteria for identity provider(s). The characters "*" (asterisk) and "." (period) are supported as search wildcards.

See [Table 29-5](#) for details about the search parameters.

4. Click **Search**.

5. Select the appropriate partner in the Search Results table and click **Edit** in the toolbar. A new tab is activated that displays the partner's attributes. In addition to the attributes. See [Table 29-6](#) for more information about advanced attributes that you can modify.
 - Enable Global Logout
 - Encrypt Assertion
 - SSO Response Binding (HTTP POST or Artifact)
6. Click **Save** to keep the changes.

**Note:**

If using SAML 1.1, you can include a certificate in the signature.
See `addCertificateRequest`.

29.5 Using Attribute Mapping Profiles

Identity Federation (when configured as an SP) supports the capability to request attributes from an IdP during the Federation process.

To configure for this, map the name of an attribute from the incoming Assertion to a local attribute that will be available in the Access Manager session (`$session.attr.fed.attr.ATTR_NAME`, for example). An IdP Attribute Mapping Profile contains these mappings.

Similarly, Identity Federation (when configured as an IdP) supports including attributes in an SSO Assertion or allowing SP partners to request that attributes be placed in the SSO Assertion. Configuring Identity Federation as an IdP involves setting up an SP Attribute Mapping profile that defines the name of the attribute in the SSO assertion, the expression to be used to populate the attribute value, and whether or not to always send the attribute in the SSO Assertion.

**Note:**

The protocol used by the provider must support the feature; for example, OpenID 2.0.

Each partner type (IdP or SP) references an Attribute Mapping Profile that defines the applicable mappings. It indicates how to map attributes for that partner to attributes defined in the Identity Federation server. If a partner does not have an Attribute Mapping Profile defined, the default Attribute Mapping Profile (based on the partner type) will be used. There is a default Attribute Mapping Profile for each provider type.

- SP Attribute Mapping Profile: Each SP partner profile will reference an SP Attribute Mapping Profile. A default SP Attribute Mapping Profile will be used if none is configured. See [Using the SP Attribute Mapping Profile](#).

- IdP Attribute Mapping Profile: Each IdP partner profile will reference an IdP Attribute Mapping Profile. A default IdP Attribute Mapping Profile will be used if none is configured.
See [Using the IdP Attribute Mapping Profile](#).

29.5.1 Using the SP Attribute Mapping Profile

When the Identity Federation instance is configured as an IdP, the SP Attribute Mapping Profile allows the administrator to define which message attributes (included in an incoming or outgoing Identity Federation message) map to which Access Manager session attributes.

An expression is used to find the value for the Access Manager attribute when including it in an Assertion or outgoing message. [Table 29-7](#) documents some sample SP attribute mappings.

Table 29-7 Sample SP Attribute Mappings

Message Attribute	Access Manager Session Attribute	Always Send
mail	\$user.attr.mail	
firstname	\$user.attr.givenname	true
lastname	\$user.attr.sn	true
authn-level	\$session.authn_level	true

Always Send indicates if the attribute should be sent even when it has not been specifically requested. If an attribute has to be included in an outgoing Assertion irrespective of whether it has been requested, Always Send should be set to true. If Always Send is false, this attribute will not be included in the Assertion unless requested. When an SP sends a request, message attributes are looked up and the mapping value for this message attribute is calculated by evaluating its expression.

Note:

The Value expression will use the OAM Policy Expression Language. More than one message attribute can have the same value expression.

See [Introduction to Policy Responses for SSO](#).

When you create or modify an SP partner profile, the available Attribute Mapping Profiles are displayed in a drop-down list. `sp-attribute-profile` is the default profile.

See [Creating Remote Service Provider Partners](#).

Select the default or click the green plus sign to create a custom mapping profile. When creating a new Attribute Mapping for an SP partner, the expressions can be embedded in the value string of the attribute. These expressions are then replaced by their runtime values.

[Table 29-8](#) lists the expressions for the attribute mapping values.

Table 29-8 Attribute Mapping Value Expressions

Value Type	Accepted Values	Expression
request	httpheader.HTTP_HEADER_NAME	HTTP_HEADER_NAME being the name of an HTTP Header stored as \$request.httpheader.HTTP_HEADER_NAME
request	cookie.COOKIE_NAME	COOKIE_NAME being the name of a cookie stored as \$request.cookie.COOKIE_NAME
request	client_ip	stored as \$request.client_ip
session	authn_level	stored as \$session.authn_level
session	authn_scheme	stored as \$session.authn_scheme
session	count	stored as \$session.count
session	creation	stored as \$session.creation
session	expiration	stored as \$session.expiration
session	attr.ATTR_NAME	ATTR_NAME being the name of an Access Manager Session Attribute stored as \$session.attr.ATTR_NAME
user	userid	stored as \$user.userid
user	id_domain	stored as \$user.id_domain
user	guid	stored as \$user.guid
user	groups	stored as \$user.groups
user	attr.ATTR_NAME	ATTR_NAME being the name of an LDAP User Attribute stored as \$user.attr.ATTR_NAME
expression (Based on the identifiers defined above and qualified with the type of data)	- request: <ul style="list-style-type: none"> \$request.httpheader.HTTP_HEADER_NAME \$request.cookie.COOKIE_NAME \$request.client_ip 	- <ul style="list-style-type: none"> HTTP_HEADER_NAME being the name of an HTTP Header COOKIE_NAME being the name of a cookie
expression (Based on the identifiers defined above and qualified with the type of data)	- session: <ul style="list-style-type: none"> \$session.authn_level \$session.authn_scheme \$session.count \$session.creation \$session.expiration \$session.attr.ATTR_NAME 	- <ul style="list-style-type: none"> ATTR_NAME being the name of an Access Manager Session Attribute

Table 29-8 (Cont.) Attribute Mapping Value Expressions

Value Type	Accepted Values	Expression
expression (Based on the identifiers defined above and qualified with the type of data)	<ul style="list-style-type: none"> - from user: • \$user.userid • \$user.id_domain • \$user.guid • \$user.groups • \$user.attr.ATTR_NAME • can be any string, with '.' (dot) characters, spaces characters (like "\$user.userid" or "\$user.attr.givenname \$user.attr.sn" or "This is the number of sessions: \$session.count") 	<ul style="list-style-type: none"> - • • • • • ATTR_NAME being the name of an LDAP User Attribute •

29.5.1.1 AWS Role Mapping Attribute in SAML Response

OAM provides a new function that can be configured in the SP Attribute Profile for supporting the AWS Role mapping attribute in SAML response.

Use the following expression in the SP Attribute Profile:

```
$func.aws_assertion_role_attr_mapping("$base_expression", "account-number", "saml-provider-name")
```

where, `$base_expression` is an expression that evaluates to user groups, `account-number` is the AWS account-number, and `saml-provider-name` is the AWS saml-provider-name.

For example, the following expression:

```
$func.aws_assertion_role_attr_mapping("$user.groups", "123456789", "OAM")
```

with attribute name `https://aws.amazon.com/SAML/Attributes/Role` and user belonging to groups `OAMSSORole` and `EC2SSORole` returns the following SAML Response:

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/Role">
<AttributeValue>arn:aws:iam::123456789:role/
OAMSSORole,arn:aws:iam::123456789:saml-provider/OAM</AttributeValue>
<AttributeValue>arn:aws:iam::123456789:role/
EC2SSORole,arn:aws:iam::123456789:saml-provider/OAM</AttributeValue>
</Attribute>
```

29.5.2 Using Attribute Value Mapping and Filtering

Topics

- [About Attribute Value Mapping](#)

- [Configuring Attribute Value Mapping](#)
- [About Attribute Value Filtering](#)
- [Configuring Attribute Value Filtering](#)

29.5.2.1 About Attribute Value Mapping

Attribute value mapping allows you to specify the value that needs to be assigned to the local attribute in a SAML message when sending or receiving messages.

Attribute value mapping has the following characteristics:

- A value mapping consists of a combination, or duet, of a local value and the corresponding external value.
- Value mappings can be defined for any local attributes. Multiple value mappings can be defined for each local attribute.
- Different external values can be mapped to the same local value using value mappings. A default attribute is used to determine which external value will be used in outgoing mode.
- Different local values can be mapped to the same external value by means of value mappings. A default attribute is used to determine which local value to use in incoming mode when mapping external values into local values.

Attribute value mapping can be configured either through the OAM console or using WLST commands.

For more information about configuring attribute value mapping through OAM console, see [Configuring Attribute Value Mapping](#).

For details about the WLST commands available for configuration, see `setSPPartnerAttributeValueMapping`, `deleteSPPartnerAttributeValueMapping`, `displaySPPartnerAttributeValueMapping`, `setIdPPartnerAttributeValueMapping`, `deleteIdPPartnerAttributeValueMapping`, and `displayIdPPartnerAttributeValueMapping` under Identity Federation Commands in *WebLogic Scripting Tool Command Reference for Identity and Access Management*.

29.5.2.2 Configuring Attribute Value Mapping

Follow the steps to define attribute value mappings through OAM Console.

On the IdP-side

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. Click **Identity Provider Management** in the Federation section.
3. Select the **Service Provider Attribute Profiles** tab and click **Search**.
4. Select the required service attribute profile from the search results, for edit.
5. In Attribute Mapping table, click **Create** and create attribute name mapping for each attribute, for which values need to be mapped.
6. In Attribute Value Mapping table, click **Create** and populate the following fields in the Attribute Value Mappings window:
 - **Message Attribute Name:** Select the required attribute name that you had created in the previous step.

- **Send Unmapped Values:** Select Send Unmapped Values to allow OAM Identity Federation to send values, for which a mapping is not defined.
- Click **Create** and populate the following fields to create a value mapping:
 - **Local Value:** The local value of the attribute
 - **External Value:** The corresponding value to be sent in external messages
 - **Ignore Case:** If selected, indicates that the string comparison must be case-sensitive when matching attribute values.
 - **Local Null:** If selected, indicates that the local value equals a null string (this is different from an empty string "").
 - **External Null:** If selected, indicates that the external value equals a null string (this is different from an empty string "").
 - **Default:** If selected, indicates that this local value will be used in case an incoming external value is mapped to several local values.

On the SP-side

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. Click **Service Provider Management** in the Federation section.
3. Select the **Identity Provider Attribute Profiles** tab and click **Search**.
4. Select the required identity attribute profile from the search results, for edit.
5. In Attribute Mapping table, click **Create** and create attribute name mapping for each attribute, for which values need to be mapped.
6. In Attribute Value Mapping table, click **Create** and populate the following fields in the Attribute Value Mappings window:
 - **OAM Session Attribute Name:** Select the required attribute name that you had created in the previous step.
 - **Receive Unmapped Values:** Select Receive Unmapped Values to allow OAM Identity Federation to receive values, for which a mapping is not defined.
 - Click **Create** and populate the following fields to create a value mapping:
 - **Received Value:** The value received by the SP provider.
 - **External Value:** The corresponding value to save in the OAM session.
 - **Ignore Case:** If selected, indicates that the string comparison must be case-sensitive when matching attribute values.
 - **Received Null:** If selected, indicates that the received value equals a null string (this is different from an empty string "").
 - **External Null:** If selected, indicates that the external value equals a null string (this is different from an empty string "").
 - **Default:** If selected, indicates that this external value will be used in case a received value is mapped to several external values.

Example

The following example shows the value mappings configuration for the attribute `title` and the corresponding result.

Sample Configuration

- Attribute Name: `title`
- Unmapped Values:
 - Send: Selected
 - Receive: Selected
- Value Mappings:

Local Value	External Value	Ignore Case	Local Null	External Null	Default
Senior Member of Technical Staff	smts	selected			selected
Principal Member of Technical Staff	pmts	selected			
	none		selected		
Senior Member of Technical Staff	srmts	selected			
Consulting Member of Technical Staff	cmts	selected			

Sample Results

External Value	maps to Local Value
Consulting Member of Technical Staff	cmts
PRINCIPAL MEMBER OF TECHNICAL STAFF	pmts
Principal Member of Technical Staff	pmts
Senior Member of Technical Staff	smts
Vice President	Vice President

Local Value	maps to External Value
NULL	none
smts	Senior Member of Technical Staff
srmts	Senior Member of Technical Staff
CEO	CEO

Note the following:

- As value mappings was defined as case-insensitive, both "PRINCIPAL MEMBER OF TECHNICAL STAFF" and "Principal Member of Technical Staff" gets mapped to `pmts`.
- As Unmapped Values: Send is selected and there is no rule defined for value "Vice President", it is mapped to itself.
- As `smts` is defined as default local value for "Senior Member of Technical Staff", "Senior Member of Technical Staff" gets mapped to `smts` even though `srmts` also maps to "Senior Member of Technical Staff".

- A local value of NULL gets mapped to the string `none`.
- Both `smts` and `srmts` map to "Senior Member of Technical Staff"
- As Unmapped Values: Receive is selected and there is no rule defined for "CEO", it is mapped to itself.

29.5.2.3 About Attribute Value Filtering

Attribute value filtering lets you specify the local values that can be allowed when sending a SAML message.

Attribute value filtering has the following characteristics:

- Filter rules can be defined for any local attributes. A filter rule evaluates each attribute value to determine if it can be sent. If the evaluation is positive, the value is sent; otherwise, it is removed from the list of attribute values to be sent.
- Multiple filter rules can be defined for each local attribute. When sending a value, OAM Identity Federation can be setup to either:
 - send only after all filters evaluate successfully
 - send if at least one filter evaluates successfully
- You can define a filtering rule by specifying the type of comparison and the string value to compare.
- OAM Identity Federation supports the following comparison types when comparing the attribute value to a string:
 - equals
 - not equals
 - starts with
 - ends with
 - contains
 - does not contain
 - equals null
 - not equals null
- In addition to these comparison types, filtering supports regular expressions, allowing you to match the attribute value against a regular expression. The filtering rules allow you to specify whether the comparison will be case-sensitive.

Attribute value filtering can be configured either through the OAM console or using WLST commands.

For more information about configuring attribute value filtering through OAM console, see [Configuring Attribute Value Filtering](#).

For details about the WLST commands available for configuration, see `setSPPartnerAttributeValueFilter`, `deleteSPPartnerAttributeValueFilter`, and `displaySPPartnerAttributeValueFilter` under **Identity Federation Commands** in *WebLogic Scripting Tool Command Reference for Identity and Access Management*.

29.5.2.4 Configuring Attribute Value Filtering

Follow the steps to define attribute value filtering through OAM Console.

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. Click **Identity Provider Management** in the Federation section.
3. Select the **Service Provider Attribute Profiles** tab and click **Search**.
4. Select the required service attribute profile from the search results, for edit.
5. In Attribute Mapping table, click **Create** and create attribute name mapping for each attribute, for which values need to be filtered.
6. In Attribute Value Filter table, click **Create** and populate the following fields in the Attribute Value Filter window:
 - **Message Attribute Name:** Select the required attribute name that you had created in the previous step.
 - **Condition Operator:** Select one of the following
 - Select **AND** to indicate that all conditions need to be met.
 - Select **OR** to indicate that meeting one condition is sufficient for an attribute to be sent.
 - Click **Create** and populate the following fields:
 - **Condition:** The condition that will be used to evaluate the attribute value. For more details about the conditions supported, see [Table 29-9](#)
 - **Expression:** The value or regular expression that will be used to evaluate the attribute value.
 - **Ignore Case:** If selected, indicates that the string comparison must be case-sensitive when matching attribute values.

OAM provides the following filtering conditions. The rules are used to determine the allowed values. Consequently, if a rule evaluates to `true`, this means that it is permissible to send the value.

Table 29-9 Attribute Value Filtering Conditions

Condition	Description
<code>equals</code>	The filtering rule returns <code>true</code> if the expression value is equal to the outgoing attribute value.
<code>does not equal</code>	Filtering rule returns <code>true</code> if the expression value is different from the outgoing attribute value.
<code>starts with</code>	Filtering rule returns <code>true</code> if the outgoing attribute value begins with the expression value.
<code>ends with</code>	Filtering rule returns <code>true</code> if the outgoing attribute value ends with the expression value.
<code>contains</code>	Filtering rule returns <code>true</code> if the outgoing attribute value contains the expression value.
<code>does not contain</code>	Filtering rule returns <code>true</code> if the outgoing attribute value does not contain the expression value.

Table 29-9 Attribute Value Filtering Conditions

Condition	Description
<code>equals null</code>	Filtering rule returns <code>true</code> if the outgoing attribute value is null.
<code>does not equal null</code>	Filtering rule returns <code>true</code> if the outgoing attribute value is not null.
<code>regex</code>	Filtering rule returns <code>true</code> if the outgoing attribute value matches the regular expression, which is defined in the expression value.

 **Note:**

- The expression value must be a standard Unix regular expression.
- The `ignoreCase` flag is disregarded during attribute value processing because regular expressions already support case-insensitivity.

Example: Regular Expression	Description
<code>.*rector</code>	any string that ends with <code>rector</code>
<code>[^abc]</code>	any character except <code>a</code> , <code>b</code> , or <code>c</code> (negation)
<code>user\d</code>	<code>user0</code> , <code>user1</code> , ..., <code>user9</code>
<code>a*b</code>	any string that begins with <code>a</code> characters and ends with a letter <code>b</code> (for example, <code>aaaaab</code>)

Example 1

The following example shows the value filter configuration for the attribute `title` and the corresponding result.

Sample Configuration

- Attribute Name: `title`

- Condition Operator: and
- Value Filters:

Condition	Expression	Ignore Case
does not equal	Vice-President	selected
contains	President	selected

Sample Result

Value	Send Value?
Vice-President	no
President	yes
Vice-President	no
Senior Vice-President	yes

Example 2

Suppose attribute value mappings are as defined in the: [Configuring Attribute Value Mapping Example](#)

The value filter configuration for the attribute `title` provides the result as shown:

Sample Configuration

- Attribute Name: `title`
- Condition Operator: and
- Value Filters:

Condition	Expression	Ignore Case
does not equal	<code>mngr</code>	selected
ends with	<code>mts</code>	not selected

Sample Result

Value	Send Value?	Value Sent
<code>mngr</code>	no	
<code>cmts</code>	yes	Consulting Member of Technical Staff

Note:

- For a value to be sent, it must not equal `mngr`, so the value `mngr` is not sent.
- `cmts` can be sent (as all filter conditions evaluate to true), and it is mapped to Consulting Member of Technical Staff.

The same results apply for the following value filters, as well:

Condition	Expression	Ignore Case
does not equal	mngr	selected
regexp	*mts	N/A

29.5.3 Using the IdP Attribute Mapping Profile

When the Identity Federation instance is configured as an SP, the IdP Attribute Mapping Profile allows the administrator to define which attributes (included in an incoming or outgoing Identity Federation message) map to which Access Manager session attributes.

The profile allows for the inclusion of the following data:

- **Message Attribute:** the name of the attribute in the incoming/outgoing Federation messages.
- **Access Manager Session Attribute:** the name by which the attribute is known to the local Access Manager server.
- **Request From Partner:** Indicates if this attribute is sent in the Request made to the IdP (a value for this attribute is requested by the SP).

[Table 29-10](#) provides sample IdP attribute mappings.

Table 29-10 Sample IdP Attribute Mappings

Message Attribute	Access Manager Session Attribute	Request for Inclusion
mail	email	true
givenname		true
sn	surname	
uid	uid	

In a protocol where a SP can specify which attributes are required in a response from the IdP, a Message Attribute name is sent in the request to the IdP. In cases when the SP receives an assertion or response from an IdP, the Attributes from the assertion are stored in the Access Manager session. If no Access Manager value is specified, the Message Attribute is stored.

When creating or modifying an IdP partner profile, the Attribute Mapping Profile is displayed with a drop-down list. The `idp-attribute-profile` is the default profile. Select the default or click the green plus sign to create a custom mapping profile.

See [Creating Remote Identity Provider Partners](#).

The Ignore Unmapped Attributes checkbox (in the configuration screen) indicates how to deal with Assertion Attributes not present (or that are present but have no value in the Access Manager Session Attribute column). If this checkbox is not checked, all Assertion Attributes that are not present in the table (or don't have a value mapped to Access Manager) will be stored in the Access Manager session with the same attribute name it had in the Assertion. If checked, any Assertion Attribute not present in the table (or with no value mapped to Access Manager) will be ignored and not added to the Access Manager session.



Note:

When the Identity Federation instance is configured as an SP it can request attributes only if the federation protocol used supports it. OpenID 2.0 supports this feature; SAML 2.0 and SAML 1.1 do not.

29.6 Mapping Federation Authentication Methods to Access Manager Authentication Schemes

A Federation Authentication Method (FAM) is an identifier representing an authentication mechanism in Federation messages.

This identifier can either be well known (such as the identifiers defined in the SAML specifications like

`urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport` or `urn:oasis:names:tc:SAML:1.0:am:password`) or it can be an arbitrary identifier agreed upon between the two communicating partners.

In its responsibilities as an IdP, Identity Federation generates an Assertion (SAML or OpenID) that might contain information on how the user was authenticated. During the Assertion generation process, the IdP will retrieve the Authentication Scheme with which the user was authenticated and attempt to map it to a FAM. If such a mapping exists, the IdP will include the FAM in the outgoing Assertion. If no mapping exists, the IdP will include the defined Authentication Scheme as the FAM in the Assertion.



Note:

Session attributes can be used in proxy mode when a mapping is not defined. Identity Federation (when acting as an IdP) can use session attributes for the FAM value when creating the assertion, if both protocols are equivalent.

Table 29-11 lists the default, out-of-the-box mappings between FAMs and Access Manager Authentication Schemes.

Table 29-11 Default Federation Authentication Method and Access Manager Authentication Scheme Mappings

Protocol	Mapping
saml20-sp-partner-profile	urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport to: <ul style="list-style-type: none"> • LDAPScheme (scheme used if the SP Partner requests urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport) • FAAuthScheme • BasicScheme • BasicFAScheme

Table 29-11 (Cont.) Default Federation Authentication Method and Access Manager Authentication Scheme Mappings

Protocol	Mapping
saml11-sp-partner-profile	urn:oasis:names:tc:SAML:1.0:am:password to: <ul style="list-style-type: none"> • LDAPScheme (scheme used if the SP Partner requests urn:oasis:names:tc:SAML:1.0:am:password) • FAAuthScheme • BasicScheme • BasicFAScheme

More details are included in the following topics:

- [Understanding Federation SSO As An IdP](#)
- [Understanding Federation SSO As An SP](#)
- [Configuring an Alternate Authentication Scheme](#)
- [Using WLST For Mapping Administration](#)

29.6.1 Understanding Federation SSO As An IdP

When Identity Federation acts as an IdP, it processes incoming Authentication Request messages sent by SP partners.

These messages might specify a FAM with which the user should be challenged by Access Manager (the IdP). If the Authentication Request contains a FAM, the IdP will attempt to map it to an Access Manager Authentication Scheme. If such a mapping is defined, Access Manager will authenticate the user using that scheme - only if the user needs to be challenged. The user would need to be challenged if, for example, the session timed out or does not exist or, the authentication level of the current session is lower than the level of the mapped Authentication Scheme or, the user has not yet been authenticated by Access Manager. If no mapping is defined, the IdP will return an error to the SP indicating that the FAM is unknown.

When the IdP Authentication Module invokes Access Manager to challenge the user, it will determine the Authentication Scheme to be used for the operation in one of the following ways:

- The SP requests a specific means to authenticate the user with a Federation Authentication Request.
- The SP settings in the IdP configuration that define a default scheme. The Partner configuration is checked first, followed by the Partner Profile configuration and finally the global default Authentication Scheme defined in the IdP configuration (LDAPScheme).

Note:

By default, the Partner and Partner Profile configurations do not define a default Authentication Scheme. As such, the global default Authentication Scheme is in effect: LDAPScheme.

After authentication, the IdP creates an Assertion and maps the Access Manager Authentication Scheme (and appropriate level) to a FAM, if such a mapping exists. The FAM is set as the Authentication Context. If no mapping exists, Identity Federation sends the default Access Manager Authentication Scheme as the Authentication Context. Following this process, the user is redirected back to Identity Federation.

29.6.2 Understanding Federation SSO As An SP

When acting as an SP in a Federation SSO process, Identity Federation processes an incoming Assertion generated by an IdP partner.

This process results in the creation of an Access Manager session for the user and the mapping of the FAM contained in the Assertion to the default SchemeID/Access Manager authentication scheme. Identity Federation provides the authentication level, if set, that should be used when Access Manager creates the user session. (By default, the Authentication Level of the Access Manager session will be set to the Authentication Level of the defined FederationScheme.) The FAM will be saved as a session attribute.

The administrator can define a mapping where the SP will create an Access Manager session with a level set to the mapped Authentication Level for the FAM contained in the Assertion. This provides a way to reflect the strength of the mechanism with which the user was originally authenticated by the IdP.

29.6.3 Configuring an Alternate Authentication Scheme

An alternate Authentication Scheme is only configurable using WLST commands and not the Oracle Access Management Console.

During a Federation SSO operation, the IdP invokes the Access Manager Authentication Module to challenge the user when required; for example, if the user is not authenticated in Access Manager, has an Access Manager session that has been inactive too long or timed out or, if the Service Provider indicates (with a Federation Authentication Request) that the IdP must re-challenge the user. For certain clients, an IdP might be required to use another Authentication Scheme to challenge a user besides the default one. This is especially true for mobile phones when an administrator might want to challenge a user with an Authentication Scheme that is different than the one used for computer-based browsers; for example, instead of an HTTP Basic Authentication Scheme, a scheme designed for mobile clients would be used.

Identity Federation (when working as an IdP) can be configured to evaluate whether an alternate Authentication Scheme should be used instead of the configured one by examining the HTTP Header sent by the user's browser. Identity Federation evaluates based on the following configurable settings:

- A setting indicating which HTTP Header attribute is sent by the user's browser.
- A setting containing a regular expression that will evaluate the value of the above HTTP Header attribute.
- A setting containing the alternate Authentication Scheme to use.

Note:

If the SP requested a specific Authentication Scheme, evaluation does not apply.

For information on the `setSPPartnerAlternateScheme` and `setSPPartnerProfileAlternateScheme` WLST commands that are used to configure alternate Authentication Scheme, see *WLST Command Reference for WebLogic Server*.

29.6.4 Using WLST For Mapping Administration

All Authentication Method/Scheme/Level mappings are configured using the WLST commands.

This can be done either at the partner level or, if not defined at the partner level, at the partner profile level.

See [Using WLST for Identity Federation Administration](#).

29.7 Using the Attribute Sharing Plug-in for the Attribute Query Service

Identity Federation provides an attribute sharing plug-in to enable Access Manager to request user attributes from an IdP.

In this interaction, the SP is an `<AttributeQuery>` requestor and the IdP is an `<AttributeQuery>` responder. The Attribute Sharing Plug-in depends on the Attribute Query Service, a request/response protocol transported using SOAP.

Note:

The Attribute Sharing Plug-in leverages the `AttributeQuery` requestor service to implement (a superset of) the X.509 Authentication Based Attribute Sharing Profile (XASP) in the context of Access Manager authentication flows.

Identity Federation (when configured as an SP) can send a SAML 2.0 `<AttributeQuery>` to the IdP in response to a SOAP call. The plug-in can be configured as a step in an Authentication Scheme. It can be invoked after authentication (by another plug-in) to fetch attributes for the authenticated user and set them into the Access Manager session. The following sections contain additional details.

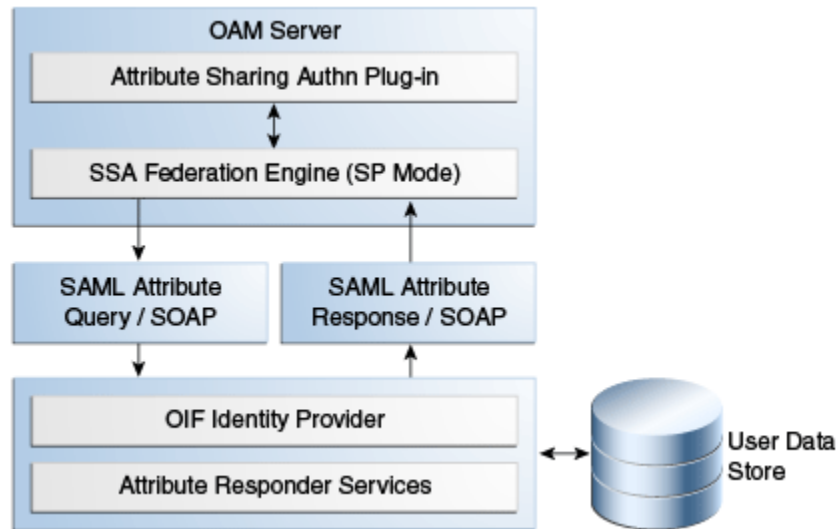
- [Understanding the Plug-in and Query Service Design](#)
- [Configuring for Attribute Sharing](#)

29.7.1 Understanding the Plug-in and Query Service Design

Identity Federation must be configured as an SP to request user attributes from a remote IdP.

[Figure 29-4](#) illustrates the design of the Attribute Sharing plug-in from a high level.

Figure 29-4 Attribute Sharing Plug-in Design



The Attribute Sharing plug-in can be part of an Access Manager Custom Authentication Module and is invoked after a user has been authenticated. The Attribute Sharing plug-in will fetch the user attributes by invoking the Identity Federation Java API, setting the attributes into the Access Manager session and transforming the Java arguments into an Attribute Request that can be processed by the SP. The Identity Federation SP receives the Attribute Request (at an exposed SOAP endpoint), determines the attributes being requested and sends an (optionally) signed and encrypted SAML 2.0 <AttributeQuery> using the requested attribute names over a SOAP/HTTP/SSL channel to the IdP's Attribute Responder Service.

 **Note:**

When invoking the Attribute Sharing plug-in, the framework will provide the following for inclusion in the <AttributeQuery>:

- User ID of the authenticated user or SubjectDN if available
- Partner ID user session attribute (available only if the Federation Authentication plug-in was used to authenticate user)
- Tenant Name
- IdP Name if the plug-in was created specifically for an IdP

The Attribute Responder Service (at the remote IdP) receives the <AttributeQuery>, decrypts it (verifying the signature if necessary) and determines (from its local policy) if the SP is authorized to request the attributes. If so, it retrieves the attributes from a user repository, constructs and (optionally) signs and encrypts an <Assertion> (with an <AttributeStatement> containing the attribute values) and returns a <Response> with the assertion to the SP. On receiving the <Response>, the SP decrypts the assertion, verifies (if necessary) its signature, extracts the attributes from the assertion and set the information in the Access Manager session. The following sections contain more details.

- [Using the SP Attribute Requester](#)

- [Using the IdP Attribute Responder](#)
- [Using the SOAP Endpoint](#)

29.7.1.1 Using the SP Attribute Requester

The Attribute Requester Service processes the SOAP Attribute Request and returns a SOAP Attribute Response.

See [Using the SOAP Endpoint](#).

The Attribute Request includes a SubjectDN and a list of other requested attributes and their values. The Attribute Requester Service identifies the IdP from which to fetch attributes by extracting one of the following (searched for in the order listed) from the request.

To use the SP Attribute Requester

1. The partner/IdP name if the request comes from the Federation engine.
2. The IdP configured in the plug-in used for authentication.
3. The request's Subject DN to determine which IdP will get the query from the configured SubjectDN-IdP map. Map the SubjectDN from most specific (cn=Joe User,ou=Finance,o=Company,c=US) to least specific (c=US).
4. The default IdP.

Following this discovery, the Attribute Requester Service retrieves the SOAP Attribute Responder Service endpoint URL from the IdP's metadata and creates a list of attributes to fetch by processing the attributes in the request through the Attribute Mapping profile.



Note:

Attribute Mapping profile specified for the target IdP will be used to change any incoming attribute names as well as add any attributes that are configured as `send-with-sso` (always requested) in the Attribute Mapping for this IdP.

A SAML Attribute Query is generated with the attribute list and sent to the IdP's SOAP endpoint. Once a response is received, the subject is verified and the each attribute is extracted from the Assertion, its value is found and both attribute and value are cached. Finally, an Attribute Response SOAP message is constructed and returned to the caller.

The following example represents a sample SOAP Attribute Request.

Sample SOAP Attribute Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <attrreq:AttributeRequest TargetIDP="adc.example.com"
      xmlns:attrreq="http://www.example.com/fed/ar/10gR3">
      <attrreq:Subject
        Format="oracle:security:nameid:format:emailaddress">alice@example.com
      </attrreq:Subject>
      <attrreq:Attribute Name="cn">
```



```

        </attrreq:Attribute>
    </attrreq:AttributeRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

The following example represents a sample SOAP attribute response.

Sample SOAP Attribute Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <soap:Envelope
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:ns2="http://
www.w3.org/2005/08/addressing" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/
policy" xmlns:enc="http://www.w3.org/2001/04/xmlenc#"
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:wst14="http://docs.oasis-
open.org/ws-sx/ws-trust/200802" xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:ic="http://schemas.xmlsoap.org/ws/2005/05/identity"
xmlns:mdext="urn:oasis:names:tc:SAML:metadata:extension" xmlns:wst="http://
schemas.xmlsoap.org/ws/2005/02/trust" xmlns:ns11="http://docs.oasis-open.org/ws-sx/ws-
trust/200512" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:ns14="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:xrds="xri://$xrds"
xmlns:xrd="xri://$xrd*($v*2.0)" xmlns:tns="http://schemas.xmlsoap.org/ws/2005/07/
securitypolicy" xmlns:ns18="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/
200702" xmlns:ns19="urn:oasis:names:tc:SAML:1.0:protocol" xmlns:ns20="http://
www.w3.org/2003/05/soap-envelope" xmlns:wssell="http://docs.oasis-open.org/wss/
oasis-wss-wssecurity-secext-1.1.xsd" xmlns:dsig="http://www.w3.org/2000/09/
xmldsig#" xmlns:query="urn:oasis:names:tc:SAML:metadata:ext:query" xmlns:wsu="http://
docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
xmlns:orafed-arxs="http://www.oracle.com/fed/ar/10gR3"
xmlns:mdattr="urn:oasis:names:tc:SAML:metadata:attribute"
xmlns:ns31="urn:oasis:names:tc:SAML:profiles:vlmetadata"
xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp">
<soap:Body><orafed-arxs:AttributeResponse CacheFor="899"> <orafed-arxs:Status>Success</
orafed-arxs:Status> <orafed-arxs:Subject
Format="oracle:security:nameid:format:emailaddress">
    alice@example.com</orafed-arxs:Subject> <orafed-arxs:Attribute Name="cn"> <orafed-
arxs:Value>alice</orafed-arxs:Value> </orafed-arxs:Attribute> </orafed-
arxs:AttributeResponse></soap:Body></soap:Envelope>

```

29.7.1.2 Using the IdP Attribute Responder

The Identity Federation IdP Attribute Responder receives the SAML Attribute Query and returns a SAML response with an Attribute Statement that contains values for the requested attributes. The IdP first identifies the requester as an SP partner and then confirms that the user is in the user data store by searching on the NameId or SubjectDN value. It then uses the Attribute Mapping profile of the SP partner to retrieve values for each of the requested attributes. Finally, it constructs and returns a SAML response containing an Attribute Statement with attribute values.

 **Note:**

The Attribute Responder uses the SP partner's Attribute Mapping profile to retrieve values. An empty value is returned for an attribute if there is no mapping present in the Attribute Mapping profile. If the value expression contains variables in the namespace of a session or request, this also evaluates to an empty string. Value Expressions in the Attribute Mapping Profile can only use variables in the namespace of `user.attr` to be evaluated correctly.

29.7.1.3 Using the SOAP Endpoint

The Attribute Requester Service on the SP exposes a SOAP interface for client requests. The SOAP service is available on the SP at the following URL:

```
http://<SP-managed-server>:<SP-port>/oamfed/ar/soap
```

29.7.2 Configuring for Attribute Sharing

Attribute Sharing Plug-in can optionally be provided with the configuration parameters.

[Table 29-12](#) lists the Attribute Sharing Plug-in.

Table 29-12 Configuration Parameters for Attribute Sharing Plug-in

Parameter	Description
NameIDValueAttribute	The name of the session attribute from which the user's nameID can be retrieved.
NameIDFormatAttribute	The name of the attribute that contains the value to be used as the nameID format.
AttributeAuthorityAttribute	The name of the attribute that contains the value used as the IdP to which the SP will send the <AttributeQuery>.
RequestedAttributes	This parameter can be used to specify attributes to be requested in the URL query format; for example, <code>attr1&attr2&attr3=value1</code> . In this case, <code>attr1</code> and <code>attr2</code> will be fetched but <code>attr3</code> will be present in the response ONLY if one of its values is <code>value1</code> .
DefaultNameIDFormat	The nameID format to be used if it is undetermined from the other parameters and session attributes.
DefaultAttributeAuthority	The default IdP partner from whom to request the user's attributes.

The Attribute Sharing Plug-in can also access the attributes documented in [Table 29-13](#). These attributes may be present in the Access Manager session during its operation.

Table 29-13 Session Attributes Accessible To Attribute Sharing Plug-in

Attribute	Description
fed.partner	If Federation was used to authenticate the user, this value is used to determine the IdP used. The same IdP would then be used for Attribute Sharing.
fed.nameidformat	If Federation was used to authenticate the user, the value of this attribute is used to determine the NameID format.
fed.nameidvalue	If Federation was used to authenticate the user, the value of this attribute is used to determine the NameID of the user. If present in the session, it will be used as the DN to locate the user in the SP's identity store.
KEY_USERNAME_DN	If this value is present, it will be used as the DN to locate the user in the SP's identity store.

The following sections have additional details on parameters and how they determine how the Attribute Sharing process.

- [NameID](#)
- [NameID Format](#)
- [IdP](#)
- [RequestedAttributes](#)

29.7.2.1 NameID

This is the name identifier of the user for whom the SP is requesting attributes.

To determine the NameID, the following searches will be conducted in order.

In the Attribute Sharing plug-in

1. If the NameIDValueAttribute is specified, retrieve the value of the specified attribute from the session and use it as the NameID.
2. If NameIDValueAttribute is not specified, use the value of `fed.nameidvalue` for the NameID.
3. If undetermined by the above, the Attribute Sharing Plug-in will invoke the Federation Engine with a null/empty NameID and the UserID (specified in the `KEY_USERNAME_DN` session attribute) is sent to the SP Attribute Requester.

In the Attribute Requester (SP)

1. If the NameID is in the Request, use its value for the user's nameID.
2. If a NameID is undetermined but a UserID is present (which occurs when invoking the Authentication Plug-in), retrieve the value of the `defaultattrrequestnameiduserattribute` attribute (found in the SP configuration for this IdP) and use it as the NameID.
3. When using SAML 2.0 only: If a NameID is not determined and SSO is configured for Simple NameID mapping, use the `nameiduserattribute` attribute (found in the SP configuration for this IdP). For example, if the value of this attribute is `$user.attr.mail`, extract the name of the user from this attribute and use it as the NameID.
4. If a NameID is still undetermined, an error is thrown.

29.7.2.2 NameID Format

This is the format of the user's NameID.

To determine the NameID format, the following searches will be conducted in order. In the Attribute Sharing plug-in

1. If the `NameIDFormatAttribute` parameter is specified, retrieve the value of the specified attribute and use it as the NameID format.
See [Table 29-12](#).
2. Use the value of the `fed.nameidformat` attribute as the NameID format.
See [Table 29-13](#).
3. Use the value of the `DefaultNameIDFormat` as the NameID format.
See [Table 29-12](#).
4. If NameID Format is still undetermined, the Attribute Sharing plug-in will invoke Federation with a null/empty NameID Format.

In the Attribute Requester (SP)

1. Use the NameID Format specified in the request.
2. Use the value of the `defaultattrrequestnameidformat` attribute (found in the SP configuration for this IdP).
3. When using SAML 2.0 only: If the NameID Format is still undetermined, use the value of the `defaultauthnrequestnameidformat` attribute (found in the SP configuration for this IdP).
4. If a NameID Format is still undetermined, an error is thrown.

29.7.2.3 IdP

This is the IdP partner to which the attribute request should be sent.

To determine the IdP partner, the following searches will be conducted in order. In the Attribute Sharing plug-in

1. If the `AttributeAuthorityAttribute` as specified, retrieve its value and use it as the IdP name.
See [Table 29-12](#).
2. Use the value of the `fed.partner` attribute as the IdP name.
See [Table 29-13](#).
3. Use the value of the `DefaultAttributeAuthority` parameter as the IdP name.
See [Table 29-12](#).
4. If the IdP is still undetermined, the Attribute Sharing plug-in will invoke Federation with a null/empty NameID Format.

In the Attribute Requester (SP)

1. Use the IdP name included with the request sent to the Attribute Sharing plugin.

2. When using x509 only: look up the dn-idp mapping to determine the IdP for this user DN.
3. Use the value of the `defaultattrauthority` attribute (found in the SP configuration).
4. Use the value of the `defaultssoidp` attribute (found in the SP configuration).
5. If an IdP name is still undetermined, an error is thrown.

29.7.2.4 RequestedAttributes

These are the attributes to be requested from the Attribute Authority.

To determine the attributes, the following searches will be conducted in order. In the Attribute Sharing plug-in

If the `RequestedAttributes` parameter is defined, use the attributes specified. If none are specified, no attributes are sent.

See [Table 29-12](#).

In the Attribute Requester (SP)

1. If the `RequestedAttributes` parameter is defined, use the attributes specified.
See [Table 29-12](#).
2. Append (or add) attributes to the `request from partner(send-with-sso)` attribute in the IdP partner profile.

In the Attribute Responder (IdP)

1. If the `<AttributeQuery>` from the SP contains requests for specific attribute values, return values for those attributes.
2. If no attribute values are requested, return any attributes specified as `Always Send (send-with-sso)` in the SP attribute profile configuration.

29.8 Using the Federation Proxy

When configured as an IdP, Identity Federation can enable the Federation Proxy to receive an Authentication Request from a remote SP partner.

Rather than authenticating the user locally, the IdP begins a second Federation SSO flow (SP2) with a second, remote IdP (IdP2). IdP2 then authenticates the user, creates an Assertion and redirects the user back to the Federation Proxy (IdP/SP2). The proxy validates the Assertion, identifies the user and resumes the first Federation SSO flow by creating a second Assertion and redirecting the user back to the original SP. With Federation Proxy, the first IdP is proxying the authentication to the second IdP.

Note:

The Federation Proxy does not refer to the HTTP Proxy settings listed under Federation Settings. That is used by Identity Federation to connect to remote servers when a firewall is present.

To use Federation IdP Proxy, the administrator configures Identity Federation to use `FederationScheme` for authentication rather than a local scheme (like `LDAPScheme` or

BasicScheme). At runtime, if the user needs to be authenticated using the FederationScheme, Identity Federation will act as an SP and start the Federation SSO flow with a remote IdP.

 **Note:**

There is an option to include the proxied Federation authentication method used by the second IdP in the Assertion created for the first SP. This is only possible if the Federation SSO operation between SP2 and IdP2 use the same protocol as the one used between SP1 and IdP1.

29.9 Using WLST for Identity Federation Administration

Identity Federation uses WLST commands for administration.

There are commands for managing authentication mappings, partner profiles, and SAML 1.1 that do not have applicable administrative fields for configuration in the Oracle Access Management Console. For information on these and other WLST commands, familiarize yourself with the following document:

See the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference for Identity and Access Management*.

29.10 Configuring OAM (IDP) for SAML Holder-of-Key (HoK) Profile with OCI Government Regions (SP)

This section provides information about configuring OAM 12c Identity Provider (IDP) for SAML Holder-of-Key (HOK) Profile with OCI-GOV Service Provider (SP) Partners.

OAM supports SAML Holder-of-Key (HoK) Profile, while acting as IDP.

 **Note:**

This configuration is applicable only for setting up Oracle OCI-GOV. HoK is enabled by default for OCI-GOV instances. The configuration is not supported with any other Service Providers (SP).

1. Stop all the Administration and Managed Servers.
2. Edit the `$OAM_DOMAIN_HOME/bin/setDomainEnv.sh` and set the following system properties, as shown:

```
EXTRA_JAVA_PROPERTIES="-Dosdt.useLineBreaks=false  
-  
Dcom.sun.xml.ws.spi.db.BindingContextFactory=com.sun.xml.ws.db.glass  
fish.JAXB RIContextFactory  
-Djavax.xml.bind.JAXBContext=com.sun.xml.bind.v2.ContextFactory $  
{EXTRA_JAVA_PROPERTIES}"  
export EXTRA_JAVA_PROPERTIES
```

3. Start the Administration and Managed Servers.
4. OCI-GOV instances have HoK enabled by default. Configure OAM federation with OCI-GOV. For setting up OCI-GOV with Identity Providers, using federation, see [Federating with Identity Providers](#)
5. Set up X.509 Authentication for the SP Partner by configuring the out-of-the-box X509AuthScheme to use the X509Plugin. Alternatively, you can use any Custom Plugin that is capable of an X509 challenge. Optionally, you can also configure OCSP validation. For details, see [Native X.509 Authentication Module](#)
6. (Optional, but Recommended) Set a default Authentication Scheme for the SP and IDP partners using the following WLST command:

```
setSPPartnerDefaultScheme(partner="<OCI_GOV_SP_PARTNER_NAME>", authnScheme=
"<X509SchemeName>")
setIdPDefaultScheme("<X509SchemeName>")
```

7. Enable SSO HoK profile using the following WLST command. This results in the SAML metadata being enabled.

```
putBooleanProperty("/fedserverconfig/hokprofileenabled", "true")
```

8. Exchange metadata with OCI-GOV and establish a federation agreement.
9. Map the incoming SAML Authentication Context Class to the appropriate OAM Authentication Scheme for the SP Partner.

```
addSPPartnerAuthnMethod("<OCI_GOV_SP_PARTNER_NAME>",
"urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient", "<X509SchemeName>")
```

10. Configure the SP Partner to respond using the SAML HoK profile.

```
updatePartnerProperty(partnerName="<OCI_GOV_SP_PARTNER_NAME>",
partnerType="sp", propName="hokenabled", propValue="true", type="boolean")
```

11. Configure OAM to use X509Scheme for the SP partner.

```
updatePartnerProperty(partnerName="<OCI_GOV_SP_PARTNER_NAME>",
partnerType="sp", propName="hokauthenticationscheme",
propValue="<X509SchemeName>", type="string")
```

12. Launch a browser and access the OCI-GOV console to confirm the setup is working. You are challenged by OAM for your X.509 certificate, after which you should be able to see the OCI-GOV Console.

30

Managing Settings for Identity Federation

Introduction to the settings that you must configure for use by Oracle Access Management Identity Federation.

This chapter includes the following sections:

- [Prerequisites for Settings in Federation Identity](#)
- [About Federation Settings](#)
- [Managing General Federation Settings](#)
- [Managing Proxy Settings for Federation](#)
- [Defining Keystore Settings for Federation](#)
- [Exporting Metadata](#)

30.1 Prerequisites for Settings in Federation Identity

The following topics presume that you have performed tasks in [Managing Identity Federation Partners](#).

30.2 About Federation Settings

This topic introduces the federation settings that must be configured to enable the Identity Federation functionality available from the Oracle Access Management Console.

[Figure 30-1](#) shows the Federations Settings page as it appears in the Oracle Access Management Console. This page is the same whether you choose Identity Federation Service Settings from the Welcome page, Configuration panel, or you display the Federation section of the System Configuration tab and choose Federation Settings.

Figure 30-1 Identity Federation Service Settings Page

Federation Settings
The following settings must be configured to enable the Identity Federation functionality available from the Oracle Access Management Console.

General

Provider Id: Encryption Key:

Succinct Id: Custom Trust Anchor File:

Signing Key:

Proxy

Enable Proxy

Host: Username:

Port: Password:

Non-Proxy Hosts:

Keystore

Keystore Location: /scratch/prate/mwhome/user_projects/domains/base_domain/config/fmwconfig/oamkeystore

Row	Key ID	Alias	password	Description
1	osts_encryption	<input type="text" value="stprivatekeyalias"/>	*****	<input type="text"/>
2	osts_signing	<input type="text" value="stprivatekeyalias"/>	*****	<input type="text"/>

Table 30-1 outlines the types of federation settings you can configure.

Table 30-1 Federation Settings in the Console

Elements	Description
General	General federation settings include basic information about the provider and the keys used to send assertions. See Also: Managing General Federation Settings
Proxy	Proxy settings enable you to set up a proxy server for federation. See Also: Managing Proxy Settings for Federation
Keystore	Keystore settings enable you to create aliases (a short hand notation) for keys in the keystore. See Also: Defining Keystore Settings for Federation

30.3 Managing General Federation Settings

The following topics describe how to manage general Federation Settings:

- [About Managing General Federation Settings](#)
- [Managing General Federation Settings](#)

30.3.1 About Managing General Federation Settings

You view and manage general federation properties on the Federation Settings page of the console.

Figure 30-1 shows the General section of the Federation Settings page.

[Table 30-2](#) describes each element on the General section of the Federation Settings page.

Table 30-2 General Federation Settings

Element	Description
Provider ID	This is the provider ID of this federation server. For example, <code>http://foo.example.com/fed</code> .
Signing Key	This key is used to sign assertions.
Encryption Key	This key is used to decrypt incoming messages.
Custom Trust Anchor File	Specifies a keystore that contains trusted root certificates use in federation. The default trust store is <code>\$DOMAIN_HOME/config/fmwconfig/amtruststore</code> . In most cases, the default trust anchor should be enough. If necessary, specify the location of an alternate keystore to use. <i>Note:</i> When you use a custom trust anchor keystore, it will not be replicated automatically across the cluster. You must manage replication of this keystore.
Export SAML 2.0 Metadata	After changes to the General settings, you must export the metadata for use by federation partners. See Also: Exporting Metadata

30.3.2 Managing General Federation Settings

General settings include basic information about a provider.

30.3.2.1 Prerequisites for General Federation Settings

None.

30.3.2.2 Setting or Modifying General Settings for Federation

You can set or modify General settings for Federation.

To set or modify:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, Select **Federation** from the drop-down list in the **Settings** section.
3. On the Federation Settings page, enter General Settings values for your ([Table 30-2](#)).
4. Click **Apply** to save your changes.
5. Proceed to "[Managing Proxy Settings for Federation](#)".

30.4 Managing Proxy Settings for Federation

This topic is organized in the following sections.

- [About Proxy Settings for Federation](#)
- [Managing Proxy Settings for Identity Federation](#)

30.4.1 About Proxy Settings for Federation

A proxy may be required when Identity Federation needs to directly connect to the federation partner, such as in a SAML artifact SSO operation. You view and manage a proxy configured for use with federation partners on the Federation Settings page of the console.

[Figure 30-1](#) illustrates the Federation Proxy Settings section of the Federation Settings page. [Table 30-3](#) describes each element on the Federation Proxy Settings section of the Federation Settings page.

Table 30-3 Federation Proxy Settings

Element	Description
Enable Proxy	Checking the box enables the proxy server. When the box is unchecked, the Proxy function is disabled and related fields are inaccessible for editing.
Host	This element specifies the proxy hostname.
Port	This element specifies the proxy port number.
Non-proxy Hosts	This is a list of hosts for which the proxy should not be used. Use ',' to separate multiple hosts.
Username	This is the proxy user name to use when connecting to the proxy.
Password	This is the proxy password to use when connecting to the proxy.

30.4.2 Managing Proxy Settings for Identity Federation

Skip Step 1 if you are viewing the Federation Settings page.

30.4.2.1 Prerequisites for Proxy Settings for Identity Federation

None.

30.4.2.2 Setting or Modifying Proxy Settings for Federation

You can set or modify Proxy settings for Federation.

To set or modify:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, Select **Federation** from the drop-down list in the **Settings** section.
3. On the Federation Settings page, evaluate current proxy settings values against those needed for your environment.
4. Fill in the Proxy settings using values for your environment ([Table 30-3](#)).
5. Click **Apply** to save your changes.

- Proceed to "Defining Keystore Settings for Federation".

30.5 Defining Keystore Settings for Federation

The following topics describe how to define Keystore settings for Federation:

- [About Managing Keystore Settings for Identity Federation](#)
- [Managing Identity Federation Encryption/Signing Keys](#)

30.5.1 About Managing Keystore Settings for Identity Federation

You view and manage keystores configured for use with federation partners on the Federation Settings page of the console.

Figure 30-2 illustrates the expanded Federation Proxy Settings section of the Federation Settings page.

Figure 30-2 Keystore Settings

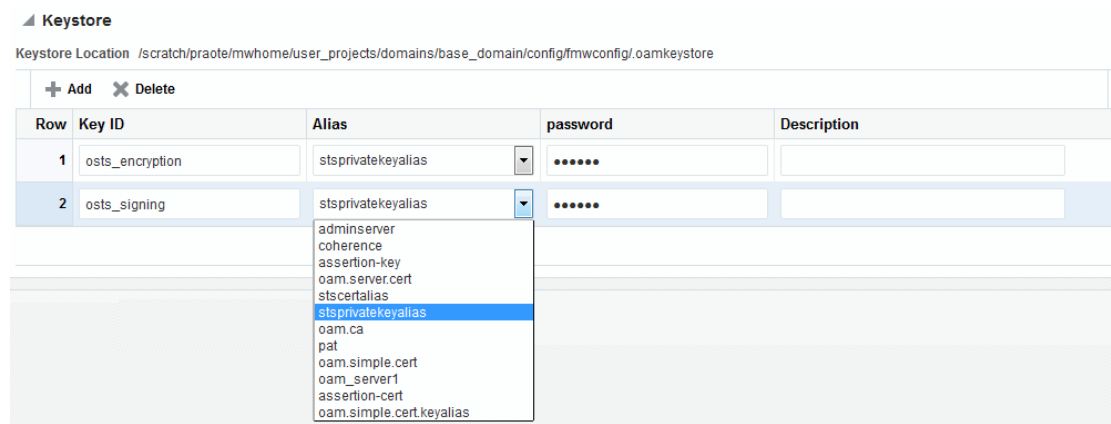


Table 30-4 describes each element on the Keystore Settings section of the Federation Settings page.

Table 30-4 Keystore Settings for Federation

Element	Description
Keystore Location	This element specifies the keystore path.
Key ID	This is the unique key ID.
Description	This element provides a brief description of the key, such as its usage type.
Alias	This element specifies the key alias. <i>Note:</i> You can choose one of the aliases that is available in the keystore using the drop-down.
Password	This element specifies the key password.

30.5.2 Managing Identity Federation Encryption/Signing Keys

As described in [Managing Data Sources](#), Identity Federation uses keys in the following keystore to store encryption and signing certificates:

```
$DOMAIN_HOME/config/fmwconfig/.oamkeystore
```

30.5.2.1 Task Overview: Managing Identity Federation Encryption/Signing Keys

- [Resetting the System \(.oamkeystore\) and Trust \(amtruststore\) Keystore Password](#)
- [Adding a New Key Entry to the System Keystore \(.oamkeystore\)](#)



Note:

AM denotes Access Manager and IF denotes Identity Federation in this discussion.

30.5.2.2 Resetting the System (.oamkeystore) and Trust (amtruststore) Keystore Password

You can reset the password that protects the keystores as well as the key entries which use the same password as the keystore.

The keystores have been created and configured by the IM/OAMAM installer, and the password and the key entries password were randomly generated. The WLST `resetKeystorePassword` method allows you to set the `.oamkeystore` password and any key entries with a password identical to the `.oamkeystore` password to a new value. The command updates the:

- `.oamkeystore` password
- Key entries in the `.oamkeystore` which had the same password as the keystore
- OAMAM/IF configuration to reflect the change
- `amtruststore` password if the keystore is protected by the same password as the `.oamkeystore` (default)

To set the system keystore (`.oamkeystore`) password:

1. Enter the WLST scripting environment.
2. Connect to the WebLogic Server AdminServer, using the `connect()` command.
3. Navigate to the domain runtime tree: `domainRuntime()`.
4. Execute the following command:

```
resetKeystorePassword()
```
5. Enter and confirm the password.

30.5.2.3 Adding a New Key Entry to the System Keystore (.oamkeystore)

You can 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 Identity Federation settings configuration screen so that it can be used to sign assertions and decrypt incoming messages.

30.5.2.3.1 Task Overview: Adding a New Key Entry to the System Keystore (.oamkeystore)

The following topics describe how to add a new entry to the system keystore to sign SAML assertions or decrypt XML-encrypted data not covered by WSS:

- [Adding a New Entry in the .oamkeystore](#)
- [Adding a New Entry in the Identity Federation Settings](#)
- [Configuring the Signing and Encryption Key](#)

30.5.2.3.2 Adding a New Entry in the .oamkeystore

There are no prerequisites for this task. The system keystore (.oamkeystore) password has been reset.

To add a new entry in the .oamkeystore:

1. Locate `keytool`.
2. Use `keytool` to:
 - Generate a self-signed certificate, or
 - Generate a certificate request, export the request to a remote Certificate Authority (CA), and finally import the certificate issued by the CA.

30.5.2.3.3 Adding a New Entry in the Identity Federation Settings

In the Identity Federation settings, you can add a new row to the Keystore table.

To add a new entry in the Identity Federation settings:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, Select **Federation** from the drop-down list in the **Settings** section.
3. On the Federation Settings page, navigate to the Keystore table.
4. Add a row.
5. Enter a key ID that will be used to reference this key when configuring Identity Federation.
6. Select the alias of the key entry stored in .oamkeystore.
7. Enter the key password.
8. Click **Apply**.

30.5.2.3.4 Configuring the Signing and Encryption Key

Once the key has been added to the keystore table, you can configure Identity Federation to use the key.

To configure the signing and encryption key:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, Select **Federation** from the drop-down list in the **Settings** section.
3. Navigate to the General section.
4. Select the Signing Key from the list of available key entries that were defined in the keystore table.
5. Select the encryption key from the list of available key entries that were defined in the keystore table.
6. Click **Apply**.

Identity Federation will now use those keys to sign and decrypt messages.

30.5.2.3.5 Using WLST for Key Transport Algorithm

Oracle Identity Federation supports RSA 1.5 as the key transport algorithm by default. The key transport algorithm can be changed from RSA 1.5 to RSA-OAEP based on the requirement, by adding a new property, `defaultkeytransportmethod` to **oam-config.xml** using the WLST commands.

You can configure the `defaultkeytransportmethod` parameter in **oam-config.xml** as follows:

```
<Setting Name="defaultkeytransportmethod" Type="xsd: xsd">  
http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgflp  
</Setting>
```

For example:

- To update the key transport algorithm for a specific partner only (in this example, OIFSP), use the following WLST command:

```
updatePartnerProperty(partnerName="OIFSP", partnerType="SP",  
propName="defaultkeytransportmethod", propValue="http://www.w3.org/2001/04/  
xmlenc#rsa-oaep-mgflp", type="string")
```

- To update the key transport algorithm for all partners that use a specific partner profile (in this example, `saml20-sp-partner-profile`), use the following WLST command:

```
putStringProperty("/fedpartnerprofiles/saml20-sp-partner-profile/  
defaultkeytransportmethod", "http://www.w3.org/2001/04/xmlenc#rsa-oaep-  
mgflp");
```

- To update the key transport algorithm for all defined SP partners, use the following WLST command:

```
putStringProperty("/idpglobal/defaultkeytransportmethod", "http://www.w3.org/  
2001/04/xmlenc#rsa-oaep-mgflp")
```

 **Note:**

This is a global change.

30.5.2.3.6 Configuring RSA OAEP Key Transport Digest and MGF Digest

The properties described in this section only apply to IdP partner configurations where the `defaultkeytransportmethod` property is set to <http://www.w3.org/2009/xmlenc11#rsa-oaep>.

A new property `defaultkeytransportdigest` is added to configure the digest used by RSA OAEP key transport algorithm. The possible values are:

- <http://www.w3.org/2000/09/xmlsig#sha1> (default)
- <http://www.w3.org/2001/04/xmlsig-more#sha384>
- <http://www.w3.org/2001/04/xmlenc#sha256>
- <http://www.w3.org/2001/04/xmlenc#sha512>

A new property `defaultkeytransportmgfdigest` is added to configure the MGF digest used by RSA OAEP key transport algorithm. The possible values are:

- <http://www.w3.org/2009/xmlenc11#mgf1sha1> (default)
- <http://www.w3.org/2009/xmlenc11#mgf1sha256>
- <http://www.w3.org/2009/xmlenc11#mgf1sha384>
- <http://www.w3.org/2009/xmlenc11#mgf1sha512>

For example:

To use the RSA OAEP key transport digest with the SHA256 digest and SHA256 MGF digest for a specific IdP partner (in this example, OIFIDP), use the following WLST commands:

```
updatePartnerProperty(partnerName=" OIFIDP ",
partnerType="idp",propName="defaultkeytransportmethod",propValue=http://
www.w3.org/2009/xmlenc11#rsa-oaep, type="string")
```

```
updatePartnerProperty(partnerName=" OIFIDP ",
partnerType="idp",propName="defaultkeytransportdigest",propValue=http://
www.w3.org/2001/04/xmlenc#sha256, type="string")
```

```
updatePartnerProperty(partnerName=" OIFIDP ",
partnerType="idp",propName="defaultkeytransportmgfdigest",propValue=http://
www.w3.org/2009/xmlenc11#mgf1sha256, type="string")
```

30.5.2.3.7 Configuring Signature Algorithm

Oracle Identity Federation supports <http://www.w3.org/2000/09/xmlsig#rsa-sha1> as the signature algorithm by default. The signature algorithm can be changed based on the requirement by using the `signedigestalgorithm` property. This property can take the following values:

- SHA-1 to use <http://www.w3.org/2000/09/xmldsig#rsa-sha1> signature algorithm (default)
- SHA-256 to use <http://www.w3.org/2001/04/xmldsig-more#rsa-sha256> signature algorithm
- PSS-SHA-1 to use <http://www.w3.org/2007/05/xmldsig-more#sha1-rsa-MGF1> signature algorithm
- PSS-SHA-256 to use <http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1> signature algorithm
- PSS-SHA-384 to use <http://www.w3.org/2007/05/xmldsig-more#sha384-rsa-MGF1> signature algorithm
- PSS-SHA-512 to use <http://www.w3.org/2007/05/xmldsig-more#sha512-rsa-MGF1> signature algorithm

The following prerequisites are required when using one of the PSS-SHA algorithms:

- Java version must be greater than 8u251
- OWSM patch 34566592 must be installed in the middleware home

For example:

To use the <http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1> signature algorithm for a specific partner (in this example, OIFSP), use the following WLST command:

```
updatePartnerProperty(partnerName="OIFSP",  
partnerType="SP",propName="signaturedigestalgorithm", propValue="PSS-  
SHA-256", type="string")
```

30.6 Exporting Metadata

After you have made changes to the general settings, you can export the metadata for use by federation partners.

To export SAML 2.0 metadata:

1. In the Oracle Access Management Console, click **Federation** at the top of the window.
2. In the Federation console, Select **Federation** from the drop-down list in the **Settings** section.
3. On the Federation Settings page, click **Export SAML 2.0 Metadata**.
4. A dialog box appears where you must specify the file for the exported metadata.
5. Click **Save** to save your new metadata file.

31

Managing Federation Schemes and Policies

If you want to enable Oracle Access Management Access Manager to work with federation providers, you define one or more authentication schemes. The defined schemes authenticate users that request access to resources protected by Access Manager.

The following topics introduce authentication schemes and policies that you can configure for Oracle Access Management Identity Federation:

- [Use of Identity Federation and Access Manager Together](#)
- [Using Authentication Schemes and Modules for Identity Federation](#)
- [Using Authentication Schemes and Modules for Oracle Identity Federation](#)
- [Managing Access Manager Policies for Use with Identity Federation](#)
- [Testing Identity Federation Configuration](#)
- [Using the Default Identity Provisioning Plug-in](#)
- [Configuring the Identity Provider Discovery Service](#)
- [Integrating OAM Identity Provider With Microsoft Office 365 Service Provider](#)

31.1 Use of Identity Federation and Access Manager Together

The use of federation features in Access Manager varies depending on the release.

When you integrate with Identity Federation:

- 11g Release 1 (11.1.1) sites, and those upgrading from 11g Release 1 (11.1.1) to 11g Release 2 (11.1.2), can use the integration.
See [Integrating Access Manager 11gR2 with Identity Federation 11gR1 in the Oracle Fusion Middleware Integration Guide for Oracle Identity Management Suite](#).
- Sites with new 11g Release 2 (11.1.2) installations can leverage federation features using the Oracle Access Management Console.
See [Deploying Identity Federation with Oracle Access Management](#).

31.2 Using Authentication Schemes and Modules for Identity Federation

The following topics describe how to use authentication schemes and modules for Identity Federation:

- [About the FederationScheme Authentication Scheme](#)
- [About the FederationMTScheme](#)
- [About the FederationPlugin Authentication Module](#)
- [Managing Authentication with Identity Federation in 11g Release 2](#)

31.2.1 About the FederationScheme Authentication Scheme

FederationScheme is a general-purpose scheme for use with Identity Federation 11g Release 2 (11.1.2.2).

Figure 31-1 shows the Access Console page for FederationScheme:

Figure 31-1 FederationScheme

FederationScheme Authentication Scheme
An Authentication Scheme defines the challenge mechanism required to authenticate a user.

* Name

Description

* Authentication Level ^ v

Default

* Challenge Method v

Challenge Redirect URL

* Authentication Module v

* Challenge URL

* Context Type v

* Context Value

Challenge Parameters

Table 31-1 describes the FederationScheme.

Table 31-1 FederationScheme Element Definitions

Element	Description
Name	This is the scheme name.
Description	This is a brief description of the scheme.

Table 31-1 (Cont.) FederationScheme Element Definitions

Element	Description
Authentication Level	This is the trust level of the authentication scheme.
Default	This is a non-editable box that is checked when the Set as Default button is clicked.
Challenge Method	You may select a challenge method from those available in the drop-down box.
Challenge Redirect URL	This is the URL of another server to which user requests must be redirected for processing.
Authentication Module	This is the authentication module to use with the scheme.
Challenge URL	This is the URL to which the credential collector will redirect for credential collection. Not used by the federation plug-in.
Context Type	This element is used to build the final URL for the credential collector.
Context Value	This element is used to build the final URL for the credential collector. The value depends on the context type.
Challenge Parameters	This is the list of parameters, if any, to use with the challenge.

[Table 22-23](#) lists the specifications for `FederationScheme`.

31.2.2 About the FederationMTScheme

The `FederationMTScheme` authentication scheme is a scheme that is designed for use in multi-tenancy environments.

31.2.3 About the FederationPlugin Authentication Module

The `FederationPlugin` provides a custom authentication module.

[Figure 31-2](#) displays the module's Console page.

Figure 31-2 FederationPlugin Steps

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View + × Detach

Step Name	Description	Plug-in Name
FedAuthnRequestPlugin	Federation Authn Reque...	FedAuthnRequestPlugin
AssertionProcessing	Assertion Processing	FedUserAuthenticationPlugin

Step Details Save Cancel

Step Name FedAuthnRequestPlugin

Description Federation Authn Request to IdP

Plug-in Name FedAuthnRequestPlugin

FedForceAuthn

FedSSOIdP

FedPassive

Table 31-2 describes the attributes that you need to configure the FederationPlugin.

Table 31-2 FederationPlugin Steps

Element	Description
Step Name	This is the name of the step within the module.
Description	This element contains a brief description of the step.
Plugin Name	This element specifies the plugin associated with the step.

The value of FedSSOIdP is the IDP to be picked up by the authentication plugin.

Orchestration enables you to specify the order of the steps within the plugin, and what to do if each of those steps succeeds or fails.

Figure 31-3 illustrates the orchestration of the FederationPlugin.

See Table 22-13 for a similar orchestration.

Figure 31-3 FederationPlugin Orchestration

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General Steps **Steps Orchestration**

You can specify the initial step

* Initial Step FedAuthnRequestPlugin ▼

Name	Description	On Success	On Failure	On Error
FedAuthnRequestPlugin		success ▼	AssertionProcessing ▼	failure ▼
AssertionProcessing		success ▼	failure ▼	failure ▼

[Table 31-3](#) describes the attributes for the orchestration of the `FederationPlugin`.

Table 31-3 Orchestration of `FederationPlugin`

Element	Description
Name	This is the step name. The steps appear in this column in order of execution, which can be modified with the Initial Step drop-down.
Description	This is a brief description of the step.
On Success	This is the action to take upon successful completion of the step, such as execution of next step in the orchestration.
On Error	This is the action to take upon error, such as taking the specified failure action.
On Failure	This is the action to take upon step failure.

31.2.4 Managing Authentication with Identity Federation in 11g Release 2

When you manage authentication with Identity Federation in 11g Release 2, you work with the `FederationScheme` and the `FederationPlugin` plug-in, a custom authentication module.

The following topics introduce authentication with Identity Federation in 11g Release 2:

- [Prerequisites for the Authentication with Identity Federation in 11g Release 2](#)
- [Viewing or Modifying `FederationScheme`](#)
- [Viewing or Modifying `FederationPlugin`](#)
- [Adding an Authentication Policy with `FederationScheme`](#)

31.2.4.1 Prerequisites for the Authentication with Identity Federation in 11g Release 2

None.

31.2.4.2 Viewing or Modifying `FederationScheme`

You can view or modify `FederationScheme` authentication scheme.

To view or modify `FederationScheme`:

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Authentication Schemes** in the **Access Manager** section.
3. Search for and open the `FederationScheme` authentication scheme.
4. Review `FederationScheme` details to ensure these are desired for your deployment.
[Table 31-1](#) describes field details.
5. Click **Save**.

31.2.4.3 Viewing or Modifying FederationPlugin

You can view or modify `FederationPlugin` authentication plug-in.

To view or modify `FederationPlugin`:

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Authentication Plug-ins** in the **Plug-ins** section.
3. Search for and open the `FederationPlugin` authentication plug-in.
4. Review `FederationPlugin` details to ensure these are desired for your deployment.

[Table 31-2](#) provides plugin step details.

5. Use the icons above the step table to add a step (+) or delete a step (x).
6. Modify the order of steps as needed using the Steps Orchestration tab.

[Table 31-3](#) provides orchestration details.

7. Click **Save**.

31.2.4.4 Adding an Authentication Policy with FederationScheme

A *Prerequisite* represents any resource to be added to a policy that you must define in the same Application Domain as the policy. You can add an authentication policy with `FederationScheme` to associate a resource that is protected by this policy.

To add an authentication policy with `FederationScheme` to associate a resource that is protected by this policy:

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Application Domains** in the **Access Manager** section.
3. Search for and open the target application domain.
4. In the application domain configuration page, click the **Authentication Policies** tab.
5. Click **Create** and enter the following General Policy Details.

[Table 25-9](#).

- Name
 - Authentication Scheme
6. Add these Global Policy Elements and Specifications:
 - Description (optional)
 - Success URL
 - Failure URL
 7. To add resources:

- a. Click the Resources tab on the Authentication Policy page.
 - b. Click the **Add** button on the tab.
 - c. Choose a URL from the list.
 - d. Repeat these steps as needed to add more resources.
8. Click **Apply** to save changes and close the confirmation window.
 9. **Responses:**
See [Introduction to Policy Responses for SSO](#).
See [Adding and Managing Policy Responses for SSO](#).

Figure 31-4 shows the console page to define the authentication policy and associate the policy to the resources.

Figure 31-4 Setting Up the Authentication Policy with FederationScheme

Create Authentication Policy Authentication Policy Apply

Authentication Policy defines the type of verification that must be performed to provide a sufficient level of trust for Access Manager to grant access to the user making the request. A single policy can be defined to protect one or more resources in the Application Domain.

* Name: Success URL:

Description:

* Authentication Scheme: Failure URL:

Resources Responses Advanced Rules

Resources + Add ✕ Delete

Resource Type	Host Identifier	Resource URL	Query String
This Policy does not protect any Resources			

31.3 Using Authentication Schemes and Modules for Oracle Identity Federation

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.

The following topics describe the authentication schemes and modules that are available for use with the Oracle Identity Federation server in Oracle Fusion Middleware Release 11g R1 (11.1.1).

See [Using Authentication Schemes and Modules for Identity Federation](#) about any schemes that are used for Identity Federation in 11g Release 2 (11.1.2.3).

See [Managing Authentication Schemes](#) for additional information about schemes.

- [About Scheme OIFScheme](#)
- [About the OIFMTLDAPPlugin Authentication Module](#)
- [Managing Authentication with Oracle Identity Federation Release 11gR1](#)

31.3.1 About Scheme OIFScheme

OIFScheme and OIFMTScheme are used for integration with Oracle Identity Federation 11g Release 1 (11.1.1).

See [Using Authentication Schemes and Modules for Identity Federation](#) for the schemes available with Identity Federation 11g Release 2 (11.1.2.3).

Figure 31-5 OIFScheme

[Table 31-4](#) describes the scheme OIFScheme.

Table 31-4 OIFScheme Definition

Element	Description
Name	This is the scheme name.
Description	This is a brief description of the scheme.
Authentication Level	This is the trust level of the authentication scheme.
Default	This is a non-editable box that is checked when the Set as Default button is clicked.
Challenge Method	Use to select a challenge method from those available in the drop-down box.
Challenge Redirect URL	This is the URL of another server to which user requests must be redirected for processing.
Authentication Module	This is the authentication module to use with the scheme.
Challenge URL	This is the URL the credential collector will redirect to for credential collection.
Context Type	Use this element to build the final URL for the credential collector.
Challenge Parameters	This is the list of parameters, if any, to use with the challenge.

[Table 22-23](#) for OIFScheme specifications.

31.3.2 About the OIFMTLDAPPlugin Authentication Module

The `OIFMTLDAPPlugin` module authenticates federated tenants through Identity Federation and non-federated tenants with the identity store associated with Access Manager.

Figure 31-6 OIFMTLDAPPlugin

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View + × Detach

Step Name	Description	Plug-in Name
TAPAuthentication	TAP response authentica...	TAPUserAuthenticationPlugin
TenantDisambiguation	Plugin to disambiguate t...	TenantDisambiguationPlugin
LocalUserIdentification	Plugin to identify local user	UserIdentificationPlugin
LocalUserAuthentication	Plugin to authenticate loc...	UserAuthenticationPlugin
FederatedTAPRequestPI...	Plugin to send TAP reque...	TAPRequestPlugin

Step Details Save Cancel

Step Name TAPAuthentication

Description TAP response authentication plugin

Plug-in Name TAPUserAuthenticationPlugin

KEY_CHECK_TOKEN_EXPIRY

KEY_IDENTITY_STORE_REF

KEY_USERNAME_ATTRIBUTE

Table 31-5 lists the steps for `OIFMTLDAPPlugin`.

Table 31-5 IFMTLDAPPlugin Steps

Element	Description
Step Name	This is the name of the step within the module.
Description	This element contains a brief description of this step.
Plugin Name	This element specifies the plugin associated with this step.
Plugin Parameters	This element lists the parameters, if any, needed for plugin execution. The parameter list varies with the plugin.

31.3.3 Managing Authentication with Oracle Identity Federation Release 11gR1

When you manage authentication with Oracle Identity Federation Release 11gR1, you work with `OIFScheme` and `OIFMTLDAPPlugin`, a custom authentication module for Identity Federation 11g Release 1 (11.1.1).

The following topics explain how to manage authentication with Oracle Identity Federation Release 11gR1:

- [Prerequisites for Authentication with Oracle Identity Federation Release 11gR1](#)
- [Viewing or Modifying the OIFScheme Authentication Scheme](#)
- [Prerequisites for Viewing or Modifying the OIFMTLDAPPlugin Authentication](#)
- [Viewing or Modifying the OIFMTLDAPPlugin Authentication](#)
- [Adding an Authentication Policy with OIFScheme](#)

31.3.3.1 Prerequisites for Authentication with Oracle Identity Federation Release 11gR1

None

31.3.3.2 Viewing or Modifying the OIFScheme Authentication Scheme

You can search for the OIFScheme Authentication Scheme and modify the Scheme details as desired.

To view or modify the Authentication Scheme:

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Authentication Schemes** in the **Access Manager** section.
3. Search for and open the OIFScheme authentication scheme.
4. Review OIFscheme details to ensure these are desired for your deployment.
See [Table 31-4](#) for field details.
5. Click **Save**.

31.3.3.3 Prerequisites for Viewing or Modifying the OIFMTLDAPPlugin Authentication

None.

31.3.3.4 Viewing or Modifying the OIFMTLDAPPlugin Authentication

You can search for the OIFMTLDAPPlugin Authentication and modify module details as desired.

To view or modify the OIFMTLDAPPlugin Authentication:

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the Application Security console, click **Authentication Modules** in the **Plug-ins** section.
3. Search for and open the OIFMTLDAPPlugin authentication module.
4. Review OIFMTLDAPPlugin details to ensure these are configured as desired for your deployment.
See [Table 31-5](#) for details.

5. Click **Save**.

31.3.3.5 Adding an Authentication Policy with OIFScheme

The procedure for this task is the same as described in the following topics:

See "[Adding an Authentication Policy with FederationScheme](#)".

31.4 Managing Access Manager Policies for Use with Identity Federation

The following topics explain how to use policy responses in Access Manager in the context of federation policies:

- [About Policy Responses with Assertion Attributes for Identity Federation](#)
- [Defining Policy Responses with Assertion Attributes for Identity Federation](#)

31.4.1 About Policy Responses with Assertion Attributes for Identity Federation

A policy can optionally include one or more authentication responses, or authorization responses, or both. You can configure the use of assertion attributes when setting up Access Manager policy responses with Identity Federation.

You use assertion attributes as follows:

- Authorization policy conditions
- Response attributes as HTTP headers
- Response attributes for identity context

[Figure 31-7](#) shows the Response configuration tab for an authorization policy:

Figure 31-7 Authorization Policy Response Tab

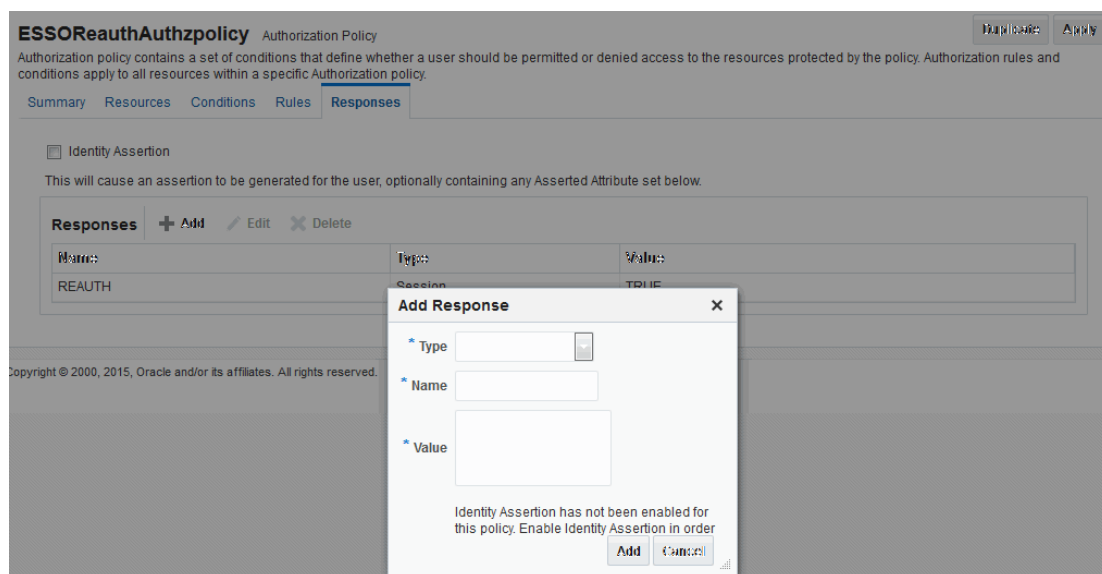


Table 31-6 describes the elements for a policy response.

Table 31-6 Policy Response Elements

Element	Description
Name	This is a unique name to distinguish this response from other responses that use the same mechanism (type).
Type	This is the mechanism used to convey the response form of the action to be taken with the value string. Select Assertion Attribute.
Value	This is the response expression, set as a variable. To provide the federation data as response attributes in the authentication or authorization policy, the values can reference: <ul style="list-style-type: none"> • <code>\$session.attr.fed.nameidvalue</code> for the name ID value • <code>\$session.attr.fed.attr.AttributeName</code> for any other assertion attribute

31.4.2 Defining Policy Responses with Assertion Attributes for Identity Federation

You can use the Oracle Access Management Console to configure policy responses with assertion attributes.

- [Background on Conditions and Responses for Identity Federation](#)
- [Prerequisites for Viewing and Configuring Policy Responses with Assertion Attributes](#)
- [Viewing or Configuring Responses with Assertion Attributes](#)

31.4.2.1 Background on Conditions and Responses for Identity Federation

Identity Federation conditions and responses must be specified separately because they are used for different tasks.

For example, if the identity provider sends a role assertion and the service provider wanted to only allow people who had a role of `sales` to gain access to the resource, you add a condition as follows:

- The Condition Namespace is "Session".
- The Name is "fed.attr.role".
- The Operator is set to EQUALS.
- Value is "sales".

A condition is used to control access to a resource within Access Manager.

 **Note:**

- Replace the role in this example to the actual SAML asserted attribute.
- If you want to use the standard SAML NameID value as the condition, then the value is "attr.fed.nameidvalue".

A response, on the other hand, enables you to pass an asserted attribute to the application. For example, if you wanted to pass the asserted attribute `role` to a back-end application in an HTTP header, you would:

- Go to the **Response** tab.
- Add a Header, name `role` (this is the name of the HTTP header).
- The value would be `$session.attr.fed.attr.role`.

Then replace the role in this example to correspond to the SAML asserted attribute.

31.4.2.2 Prerequisites for Viewing and Configuring Policy Responses with Assertion Attributes

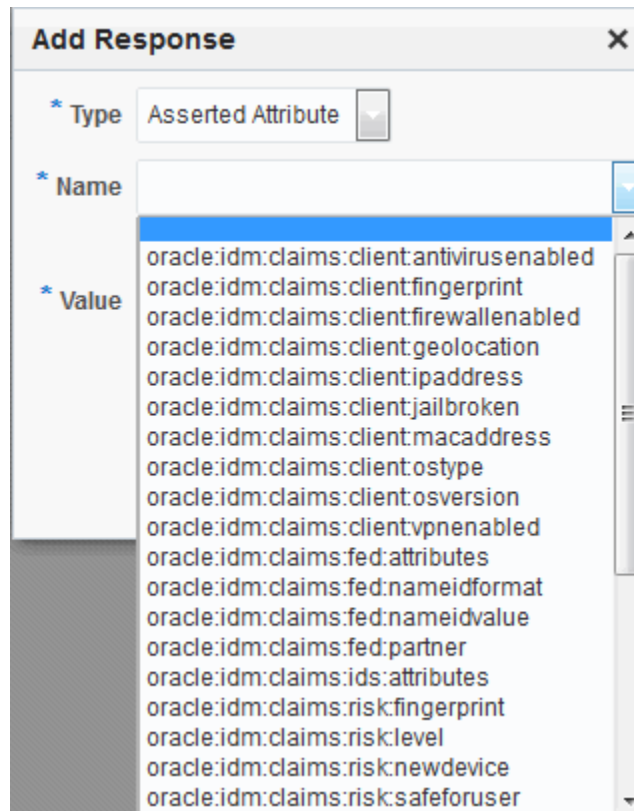
None.

31.4.2.3 Viewing or Configuring Responses with Assertion Attributes

To view or configure responses with assertion attributes:

1. Using the Oracle Access Management Console, search for the desired application domain and open the desired policy to view or configure a response.
2. Select the **Responses** tab.
3. Click the relevant icon to add, delete or update a response.
4. When updating, review the response details to ensure these are desired for your deployment.
See [Table 31-6](#) for details.
5. Click **Save**.

[Figure 31-8](#) shows an example of federation response attribute configuration.

Figure 31-8 Adding a Federation Response Attribute to an AuthZ Policy

31.5 Testing Identity Federation Configuration

After performing the procedure that is described in the previous topic, you have completed all the steps to configure federation in SP mode.

To recap, these steps are:

1. Enabling the Identity Federation service using Oracle Access Management Console.
2. Creating an IdP partner or using an existing IdP partner.
3. Ensuring that IdP setup including SAML attributes, global logout, and nameID format are configured.
4. Configuring an authentication/authorization policy that uses `FederationScheme` with federation response attributes; and
5. Protecting a resource with this policy.

To test this configuration, access the resource that is protected by the authentication policy and verify that access is granted or denied according to the policy.

31.5.1 Test SP Module

Identity Federation provides a Test SP module that enables you to Test Federation SSO with an IdP Partner and view the result of the Federation SSO operation as well as the assertion sent by the Identity Provider.

31.5.1.1 Enabling or Disabling the Test SP Module

You can enable or disable the Test SP Module.

1. Enter the WLST environment:

```
$OH/common/bin/wlst.sh
```

2. Connect to the Admin Server:

```
connect ()
```

3. Move to the domain runtime location:

```
domainRuntime ()
```

4. Execute the following WLST command to enable the Test SP Module:

```
configureTestSPEngine("true")
```

5. Execute the following WLST command to disable the Test SP Module:

```
configureTestSPEngine("false")
```

31.5.2 Accessing the Test SP Module and Performing a Federation SSO Operation

You can access the Test SP module and perform a federation SSO operation with an IdP partner.

1. Access the following service:

```
http(s)://oam-hostname:oam-port/oamfed/user/testspssso
```

2. Select the IdP with which to perform a federation SSO (*note*: only enabled IdP partners are listed).
3. Start the federation SSO operation. The browser will be redirected to the IdP Partner for authentication and redirected back to Identity Federation with a federation response.
4. Identity Federation will process the federation assertion and the Test SP module will display the result of the processing (*note*: no Access Manager session will be created as a result of the operation).

31.5.3 Troubleshooting Errors During Federation Configuration After an Upgrade

IAM Suite is the OOTB Application Domain created when OAM 11.1.2 is installed. This Application Domain can be renamed after installation but when upgrading OAM to 11.1.2.2.0,

it must be renamed back to IAM Suite otherwise the upgrade operation will fail with the following error seen in the WLS admin logs.

```
java.lang.NullPointerException
at
oracle.security.am.common.policy.tools.upgrade.r2ps2.bootstrap.FedR2PS2BootstrapHandler.createFedAuthnResource(FedR2PS2BootstrapHandler.java:505)
at
oracle.security.am.common.policy.tools.upgrade.r2ps2.bootstrap.FedR2PS2BootstrapHandler.doBootstrap(FedR2PS2BootstrapHandler.java:151)
at
oracle.security.am.common.policy.tools.upgrade.r2ps2.bootstrap.R2PS2BootstrapHelper.doBootstrap(R2PS2BootstrapHelper.java:70)
at
oracle.security.am.common.policy.tools.PolicyComponentLifecycle.initialize(PolicyComponentLifecycle.java:99)
```

If the IAM Suite Application Domain has been renamed after installation, it is required to rename it back to its original IAM Suite name prior to beginning the upgrade process. After the upgrade process is complete, the name can be changed back to a custom name.

31.6 Using the Default Identity Provisioning Plug-in

11g Release 2 (11.1.2.3) features a plug-in that you can optionally use to provision a missing identity during a federated SSO operation.

The following topics describe how to use a provisioning plug-in:

- [Why Use a Provisioning Plug-in?](#)
- [About the Default Provisioning Plug-in](#)
- [Using the Default Provisioning Plug-in](#)
- [Switching to a Custom Provisioning Plug-in](#)

31.6.1 Why Use a Provisioning Plug-in?

When a federated SSO transaction is initiated, the processing flows as follows:

1. The IdP authenticates a user and sends an assertion to Oracle Access Management Identity Federation.
2. Acting as SP, Identity Federation maps the user to the local identity store.
3. If the user does not exist in the local store, the mapping fails.

Resolving this issue requires you to provision the user so the transaction can continue.

31.6.2 About the Default Provisioning Plug-in

To handle the identity mapping failure, Identity Federation supports the ability to set up a plug-in, known as the default provisioning plug-in, to provision the missing user in the identity store and enable the federated single sign-on to proceed.

The user is provisioned in the identity store associated with the IdP partner. You can specify a list of attributes to use in provisioning the plug-in, as explained in the next section.

31.6.3 Using the Default Provisioning Plug-in

You can enable this default provisioning plug-in from the plug-in configuration interface.

To use the default provisioning plug-in:

1. From the plug-in configuration interface select `FedUserProvisioningPlugin`.
2. In the configuration parameters tab, set the following parameters:
 - `KEY_USER_RECORD_ATTRIBUTE_LIST` - This is the list of attributes with which the user should be provisioned. These attributes are available as part of the assertion, for example: `mail, givenname`. (optional)
 - `KEY_PROVIDERID_ATTRIBUTE_NAME` – This is the tenant ID attribute name in the identity store which Identity Federation populates at run-time with the tenant name. (optional)
 - `KEY_USERID_ATTRIBUTE_NAME` – This is the attribute name to use for the `userid` value from the assertion attributes. (optional)
3. Enable user provisioning with the default plug-in by executing the WLST command:

```
putBooleanProperty("/fedserverconfig/userprovisioningenabled", "true")
```

31.6.4 Switching to a Custom Provisioning Plug-in

A custom provisioning plug-in is also available with Identity Federation.

To switch from the default plug-in to the custom plug-in, follow the guidelines in *Developing a Custom User Provisioning Plug-in* chapter of the *Developing Applications with Oracle Access Management*.

When you use the custom plug-in, set the plug-in name with the WLST command:

```
putStringProperty("/fedserverconfig/userprovisioningplugin", "CustomPlugin")
```

31.7 Configuring the Identity Provider Discovery Service

Identity provider discovery is a service that selects an identity provider (possibly through interaction with the user) to use during SSO.

While Identity Federation does not provide an identity provider discovery service, it provides support for using such a service to select an IdP, if one is not passed in the authentication request to the SP during SP-initiated SSO.

See the following specifications about IdP discovery at:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery-cs-01.pdf>

When acting as a service provider, Identity Federation can be configured so that if an SSO operation is initiated without the provider ID of the partner IdP, the user is redirected to an IdP discovery service to select the identity provider with which to perform SSO.

After the user selects an identity provider, the custom page resubmits the SSO request with the chosen IdP to Identity Federation.

See the following topics for details:

- [Configuring the Bundled IdP Discovery Service](#)
- [Configuring Identity Federation with a Custom IdP Discovery Service](#)
- [Disabling the use of an IdP Discovery Service](#)

31.7.1 Configuring the Bundled IdP Discovery Service

Identity Federation provides a simple Identity Provider Discovery Service that can be used to determine the Federation IdP Partner to be used at runtime during a Federation SSO operation.

To configure the bundled IdP discovery service:

1. Enter the WLST environment:

```
$OH/common/bin/wlst.sh
```

2. Connect to the Admin Server:

```
connect ()
```

3. Move to the domain runtime location:

```
domainRuntime ()
```

4. Execute the following WLST command to configure Identity Federation to use an IdP Discovery Service:

```
putBooleanProperty("/spglobal/idpdiscoveryserviceenabled", "true")
```

5. Execute the following WLST command to configure Identity Federation to use the default out-of-the-box IdP Discovery Service:

```
putBooleanProperty("/spglobal/idpdiscoveryservicepageenabled",  
"true")putStringProperty("/spglobal/idpdiscoveryserviceurl", "/oamfed/  
discovery.jsp")
```

31.7.2 Configuring Identity Federation with a Custom IdP Discovery Service

You can configure Identity Federation to interact with a custom IdP Discovery Service that is deployed remotely.

To configure Identity Federation with a custom IdP Discovery Service:

1. Enter the WLST environment:

```
$OH/common/bin/wlst.sh
```

2. Connect to the Admin Server:

```
connect ()
```

3. Move to the domain runtime location:

```
domainRuntime ()
```

4. Execute the following WLST command to configure Identity Federation to use an IdP Discovery Service:

```
putBooleanProperty("/spglobal/idpdiscoveryserviceenabled", "true")
```

- Execute the following WLST command to configure Identity Federation to use a custom IdP Discovery Service (replace *IDP_DISCOVERY_SERVICE_URL* with the fully qualified URL of the Discovery Service):

```
putBooleanProperty("/spglobal/idpdiscoveryservicepageenabled", "false")
putStringProperty("/spglobal/idpdiscoveryserviceurl",
  "IDP_DISCOVERY_SERVICE_URL")
```

At runtime, Identity Federation redirects to the IdP Discovery Service page with the following parameters:

- `return`: This is the URL to which the page should send the new request containing the chosen IdP provider ID to Identity Federation.
- `returnIDParam`: This is the name of the parameter to use to specify the chosen IdP provider ID in the request sent to Identity Federation.

The discovery service receives the values of these parameters, displays a list of IdPs, and then sends a new request to Identity Federation specifying the chosen IdP Provider ID.



Note:

Make sure that the URL query parameter values are correctly URL-encoded.

Example of an IdP Discovery Service Page

The following example represents an IdP discovery service page that enable a user to select an identity provider (from the list of provider IDs: <http://idp1.com>, <http://idp2.com>, <http://idp3.com>), and submit the chosen provider ID to Identity Federation to continue the SSO flow.

```
<%@ page buffer="5kb" autoFlush="true" session="false"%>
<%@ page language="java" import="java.util.*, java.net.*"%>

<%
// Set the Expires and Cache Control Headers
response.setHeader("Cache-Control", "no-cache");
response.setHeader("Pragma", "no-cache");
response.setHeader("Expires", "Thu, 29 Oct 1969 17:04:19 GMT");

// Set request and response type
request.setCharacterEncoding("UTF-8");
response.setContentType("text/html; charset=UTF-8");
String submitURL = request.getParameter("return");
String returnIDParam = request.getParameter("returnIDParam");

List idps = new ArrayList();
idps.add("http://idp1.com");
idps.add("http://idp2.com");
idps.add("http://idp3.com");

%>

<html>
  <title>
    Select an Identity Provider
  </title>
  <body bgcolor="#FFFFFF"><form method="POST" action="<%=submitURL%>" id="PageForm"
name="PageForm" autocomplete="off">
```

```

<center>
  <table cellpadding="5" cellspacing="2" border="0" width="500">
    <tr><td colspan="2" align="center">
      Select an Identity Provider
    </td></tr>
  </tr>
  <tr>
    <td align="right">Provider ID</td>
    <td>
      <select size="1" name="<%=returnIDParam%>">
<%
Iterator idpIT = idps.iterator();
while(idpIT.hasNext())
{
    String idp = (String)idpIT.next();
%>
                                <option value="<%=idp%>"><%=idp%></option>
<%
}
%>
                                </select>
      </td>
    </tr>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <input type="submit" value="Continue"/>
    </td>
  </tr>
</table>
</center>
</form>
</body>
</html>

```

31.7.3 Disabling the use of an IdP Discovery Service

To disable the use of an IdP Discovery Service:

1. Enter the WLST environment:

```
$OH/common/bin/wlst.sh
```

2. Connect to the Admin Server:

```
connect ()
```

3. Move to the domain runtime location:

```
domainRuntime ()
```

4. Execute the following WLST command to configure Identity Federation to stop using an IdP Discovery Service:

```

putBooleanProperty("/spglobal/idpdiscoveryseviceenabled", "false")
putBooleanProperty("/spglobal/idpdiscoverysevicepageenabled", "false")
putStringProperty("/spglobal/idpdiscoveryseviceurl", "/oamfed/
discovery.jsp")

```

31.8 Integrating OAM Identity Provider With Microsoft Office 365 Service Provider

The following topics describe how to administer OAM Identity Federation 11g R2PS2 (11.1.2.2.0) as an IdP for integration with Microsoft Office 365 when the latter is configured as an SP leveraging the SAML 2.0 standard. After the integration implementation, you can use an account in the Identity Repository to access all web clients (including Office rich client apps connecting to SharePoint Online) and email-rich clients that use basic authentication and a supported Exchange access method such as IMAP, POP, Active Sync or MAPI. (The Enhanced Client Protocol end point is required to be deployed).

The deployment assumes that:

1. OAM 11gR2PS2 has been installed and configured using SSL.
2. An account has been created using the Oracle Access Management Console that defines the Administrator role for Office 365.
3. Windows PowerShell 2.0 and Microsoft Online Services Module have been installed.
4. An available domain name can be used as the federated domain in Office 365. Generally, this domain needs to be purchased.



Note:

For non Web-based client integration:

- The OAM IdP endpoint must be accessible from the public network.
- A trusted SSL certificate issued by a well known entity must be used.

The following topics provide configuration details:

- [Configuring Microsoft Office 365 for OAM Integration](#)
- [Configuring OAM for Microsoft Office 365 Integration](#)
- [Configuring Microsoft Office 365 for OAM Integration](#)

31.8.1 Configuring Microsoft Office 365 for OAM Integration

To configure Microsoft Office 365 for OAM integration:

1. Add the domain name (for example, test.com) and verify it using the Office 365 Web administration center.
2. Define the authentication scheme for the domain as Federated by running the `Set-MsolDomainAuthentication` PowerShell command.

```
$dom("<domain name>")
$url="https://server_host:port/oamfed/idp/samlv20"
$uri("<entityID>")
$secpUrl=https:// server_host:port/oamfed/idp/soap
$logourl="https://server_host:port/oamfed/idp/samlv20"
$cert="MIIB/DCCAWWgAwIBAgI....."
```

```
Set-MsolDomainAuthentication -FederationBrandName $dom
-Authentication Federated -ActiveLogUri $ecpUrl -PassiveLogOnUri $url
-SigningCertificate $cert -IssuerUri $uri
-LogOffUri $logouturl -PreferredAuthenticationProtocol SAML
```

 **Note:**

The values for some of these parameters can be found in the OAM Identity Provider metadata.

3. Create a user in the Federated domain by running the `New-MsolUser` PowerShell command.

```
New-MsolUser -DisplayName <name> -UserPrincipalName
<name@domain_name> -UsageLocation <location>
-BlockCredential $false -ImmutableId <immutableid>
```

Values for `UserPrincipalName` and `ImmutableId` are required by Office 365 for Federation. In the SAML assertion, the value of `ImmutableId` will be stored in the SAML Subject using the "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" NameID format. The `UserPrincipalName` will be stored in the SAML Attribute using the attribute name `IDPEmail`. In the OAM User Identity Store, the user entry must use the same attributes to store the values of `UserPrincipalName` and `ImmutableId`. Use the following:

- `mail=<name@domain_name (UserPrincipalName)>`
- `uid=<immutableid>`

 **Note:**

If Office 365 has been before this integration, you can use an existing user for testing. You must know the values of the `UserPrincipalName` and `ImmutableId` attributes for the existing user.

4. Assign a license to the user to make the applications provided by Office 365 available to the user.

31.8.2 Configuring OAM for Microsoft Office 365 Integration

The following topics describe how to configure OAM for integration with Microsoft Office 365:

- [Configuring for Web and Non-Web Clients](#)
- [Additional Configurations for Non-Web Clients](#)

See *Identity Federation WLST Commands* for details on how to use the WLST commands.

31.8.2.1 Configuring for Web and Non-Web Clients

To configure for Web and non-Web clients:

1. Log in to the Oracle Access Management Console.

2. Navigate to Available Services and enable the Identity Federation service.
3. Navigate to Identity Provider Administration.
4. Create a Service Provider Attribute Profile mapping.

Table 31-7 Message Attribute Mapping

Message Attribute Name	Value	Always Send
IDPEmail	\$user.attr.mail	true

5. Create a Service Provider Partner for Office 365 using the attributes and values. See [Table 31-8](#) for details.

Table 31-8 Office 365 Service Provider Attribute Values

Provider Attribute	Value
Name	Office365
Protocol	SAML 2.0
Service Details	Load from provider metadata
Metadata File	Can be downloaded from: https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml For customers in China using the China-specific instance of Office 365 download from: https://nexus.partner.microsoftonline-p.cn/federationmetadata/saml20/federationmetadata.xml
NameID Format	persistent
NameID Value	User ID Store Attribute + uid
Attribute Mapping Profile	The profile created in step 2
User Identity Store	Identity Store used
User Search Base DN	The base DN for User search
SSO Response Binding	HTTP POST

6. Optionally, set the default Authentication Scheme for the service provider partner using the `setSPPartnerDefaultScheme WLST` command.

By default, OAM uses `LDAPScheme` for user authentication. To use another scheme, run the following command:

```
setSPPartnerDefaultScheme(<partner>, <authnScheme>)
```

See [Additional Configurations for Non-Web Clients](#) if you use non-Web clients.

31.8.2.2 Additional Configurations for Non-Web Clients

Perform these additional configurations if using non-Web clients. These steps will not impact Web-based integration.

1. Use the `setSPPartnerAlternateScheme WLST` command to set an alternative Authentication Scheme for the Service Provider partner to handle HTTP Basic authentication. For example:


```
setSPPartnerAlternateScheme(<partner>, "true",
    httpHeaderName="X-MS-Client-Application", httpHeaderExpression=".*
    Microsoft.Exchange.*", authnScheme="BasicScheme or
    BasicSessionlessScheme")
```

The values of `httpHeaderName` and `httpHeaderExpression` can be determined from the HTTP request sent from Office365 to OAM. If you want to use other values, use rich clients to connect the email account and capture the HTTP request on OAM server side.

 **Note:**

It is recommended to use `BasicSessionlessScheme` because Office 365 only validates user credentials to get an assertion.

2. Use the `updatePartnerProperty WLST` command to update the configuration to send certificates in XML signatures.

```
updatePartnerProperty(<partner>,"sp","includecertinsignature","true","boolean")
```

For Basic Authentication, you may need re-authentication even after the Request is already authenticated.

31.8.3 Verifying Federation Single Sign-On

The following topics explain how to verify Federation SSO:

- [Verifying SP-Initiated SSO](#)
- [Verifying IDP-Initiated SSO](#)
- [Verifying Federation with Non Web-based Clients](#)

31.8.3.1 Verifying SP-Initiated SSO

To verify SP-initiated SSO:

1. Open one of the following URLs.
 - `http://portal.microsoftonline.com`: from login page, input "xxx@test.com" in the user name field, then click the password field; at this time, you should be automatically redirected to the OAM login page.
 - `http://www.outlook.com/test.com`: you should be automatically redirected to the OAM login page.
2. Enter a user name and password in the displayed OAM login page and click Login. If SSO is successful, you are then logged into the Office 365 Web portal.

31.8.3.2 Verifying IDP-Initiated SSO

To verify IDP-initiated SSO:

1. Open `http://host:port/oamfed/idp/initiatesso?providerid=urn:federation:MicrosoftOnline&returnURL=http://portal.microsoftonline.com` in a browser.

2. Enter a user name and password in the displayed OAM login page and click Login.
If SSO is successful, you will be logged into the Office 365 Web portal.

31.8.3.3 Verifying Federation with Non Web-based Clients

To verify federation with non Web-based clients:

1. Add an Email account for an email client.
 - For Desktop Email client like Outlook client, please refer to <http://help.outlook.com/en-ca/140/cc875899.aspx>
 - For Native Email app in Android device, please refer to <http://office.microsoft.com/client/15/help/preview?AssetId=HA102823196&lcid=1033&NS=O365ENTADMIN&Version=15&CTT=5&origin=HA103787372>
 - For IOS device, please refer to <http://office.microsoft.com/client/15/help/preview?AssetId=HA102818554&lcid=1033&NS=O365ENTADMIN&Version=15&CTT=5&origin=HA102828259>

 **Note:**

When adding an email account using the Outlook client, after you input Your Name and Email Address in the User Information area, it auto-fills the User Name value in the Logon Information area with the value of Your Name. It is recommended that you change the value of Your Name to reflect the email address.

2. Check that you can send and receive email successfully.

Identity Federation Use Cases

These topics describe use cases related to the Identity Federation.

- [Using Test SP Application in OIF and SP](#)
- [Using Federation Attributes for OAM Authorization and Protected Web Applications](#)
- [Key and Certificate Management and Rollover in OIF and OSTs](#)
- [Cryptographic Settings in Oracle Identity Federation](#)

32.1 Using Test SP Application in OIF and SP

This chapter describes how to enable and use the Test SP application in Oracle Identity Federation (OIF)/ Service Provider (SP), which is useful when OIF is an SP and Federation agreements are set up. The Test SP application is a web based application that allows you to perform the following tasks.

- Test the Federation SSO flows
- Start Federation SSO with the specified Identity Provider (IdP)
- Authentication at the IdP
- Verify if the mapping rules work
- Find out which attributes are sent by the IdP, how they are named and how they are processed by OIF/SP
- View the Federation token (SAML Assertion or OpenID SSO Response)
- Diagnose issues with the SAML/OpenID flows, before rolling Federation SSO out
- Implement the SP functionality of OIF via a browser without creating any OAM session
- Display the Test SP operation result and SAML Assertion/OpenId SSO response

32.1.1 Enabling Test SP Engine

The Test SP application is disabled by default. To use it, you must enable the application.

**Note:**

The Test SP application must be disabled once you have finished using it.

Use the following OIF WLST commands to enable the Test SP application:

1. Enter the WLST environment:

```
$IAM_ORACLE_HOME/common/bin/wlst.sh
```
2. Connect to the WLS Admin server:

```
connect()
```

3. Navigate to the Domain Runtime branch:

```
doimanRuntime()
```

4. Enable the Test SP Engine:

```
configureTestSPEngine("true")
```

 **Note:**

Use `configureTestSPEngine("false")` command to disable the Test SP engine.

5. Exit the WLST environment:

```
exit()
```

32.1.2 Using the Test SP Engine

The Test SP Engine allows you to perform the following tasks:

- [Starting Federation SSO Flow](#)
- [Displaying Test SP Operation Result](#)
- [Diagnosing Mapping and Response Validation Issues](#)

32.1.2.1 Starting Federation SSO Flow

Starting the Federation SSO flow involves the following steps.

1. Navigating to the Test SP application through a browser
2. Selecting the IdP to perform Federation SSO
3. Starting the operation

To start Federation SSO:

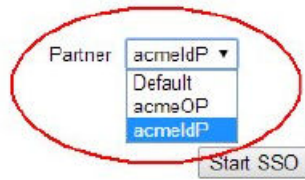
1. Use the following URL to access the Test SP application:

```
https://oam-runtime-host:oam-runtime-port/oamfed/user/testspssso
```

The Test SP application displays a drop-down list with a list of IdPs to perform Federation with.

2. Select an IdP for OIF/SP, or choose **Default** to use the Default SSO IdP.

Initiate Federation SSO



Copyright © 2006, 2014, Oracle and/or its affiliates. All rights reserved.

3. Click **Start SSO** to trigger the Federation SSO with the specified IdP.

You will be redirected to the IdP, similar to normal Federation SSO operation. The IdP:

- Either challenges you for your credentials, and then sends a SAML/OpenID response.
- Or sends an SAML/OpenID response (either because you are already authenticated or an error has occurred).

32.1.2.2 Displaying Test SP Operation Result

When the IdP redirects the user with the SAML Assertion/OpenID Response to OIF/SP, the server validates the response and maps it to an LDAP user record.

Federation SSO Operation Result

SSO Authentication Result Authentication Successful
User Identifier oud-e2e:USER:cn=alice,ou=users,dc=us,dc=oracle,dc=com:alice
Authentication Instant Fri Feb 28 07:09:56 PST 2014
SSO Primary Status Code SUCCESS
SSO Secondary Status Code
SSO Status Message
Partner acmeldP

Attributes from the Assertion

fed.partner [acmeldP]
fed.nameidformat [urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress]
fed.nameidvalue [alice@oracle.com]

Assertion Message

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" Destination="http://adc00pcc.us.GGZfZaUA2Ql6wWtvFkt8EbTgL8lpQ3vLE4NHgDN" IssueInstant="2014-02-28T15:09:56Z" Version="2.0">  
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">http://adc00peq.us.oracle</saml:Issuer>  
<samlp:Status>  
<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>  
</samlp:Status>  
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="id-Y-2GYbdNZ-DorvNnew3q">  
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">http://adc00peq.us.oracle</saml:Issuer>  
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">  
<dsig:SignedInfo>  
<dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></dsig:CanonicalizationMethod>  
<dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sa-sha1"></dsig:SignatureMethod>  
<dsig:Reference URI="#id-Y-2GYbdNZ-DorvNnew3qb3y9plM-"></dsig:Reference>  
<dsig:Transforms>  
<dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"></dsig:Transform>  
<dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></dsig:Transform>  
</dsig:SignedInfo>  
</dsig:Signature>  
</saml:Assertion>  
</samlp:Response>
```

The result of the Test SP operation displays the following information.

- The result of the authentication operation.
- The canonical user ID to which the response was mapped that contains the Identity Store Name, User DN, and User ID.
- The authentication instant.
- The IdP partner name.
- Attributes from the SSO Response that are stored in the OAM session.
- The decrypted/decoded SSO response.

32.1.2.3 Diagnosing Mapping and Response Validation Issues

If the Federation SSO between the IdP and OIF/SP is not working then use the Test SP engine together with the Oracle Access Manager (OAM) to diagnose the problem. You can diagnose the following issues:

- [Mapping Issues](#)

- Response Validation Issues

32.1.2.3.1 Mapping Issues

If the incoming SSO assertion cannot be mapped to a local LDAP user record then the Test SP application displays the following information.

- An error Message
- The NameID/attributes sent by the IdP
- The SSO message sent by the IdP, which contains the NameID/attributes

In the following example, the IdP and OIF/SP administrators agreed to use SAML 2.0 and identified the user through the email address. The issue is that the email address for Alice at the IdP is `alice.appleton@oralce.com`, where as the email address used by OIF/SP in the LDAP directory is `alice@oracle.com`.

Federation SSO Operation Result

```
SSO Authentication Result Authentication Failed
User Identifier
Authentication Instant
SSO Primary Status Code RESPONDER
SSO Secondary Status Code
SSO Status Message No user returned during attribute based authentication using NameID
mapping for name ID: alice.appleton@oracle.com and name ID format
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
Partner acmeldP
```

Attributes from the Assertion

```
fed.partner [acmeldP]
fed.nameidformat [urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress]
fed.nameidvalue [alice.appleton@oracle.com]
```

Assertion Message

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" Destination="http://adc00pcc.us.
WSRYZ0zP" IssueInstant="2014-02-20T15:10:05Z" Version="2.0">
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Format="urn:oasis:names:tc:SAML
<samlp:Status>
<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="id-PoOD-BDUeoiSY4ajPCQ
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">http://adc00peq.us.oracle
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
<dsig:SignedInfo>
<dsig:CanonicalizationMethod 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-PoOD-BDUeoiSY4ajPCQ86yjZ/Wkw-">
```

At the end of the flow, the Test SP application displays the following information in the Federation SSO Operation result.

- The authentication operation failed message.

- The assertion could not be mapped to a local user record message.
- The data extracted from the assertion and the SAML message.

The OIF log files display the following error message and the SAML message.

```
<Feb 28, 2014 7:18:05 AM PST> <Warning>
<oracle.security.fed.eventhandler.fed.profiles.sp.sso.assertion.Saml20A
sassertionProcessor>
<FED-15108> <User was not found during attribute based authentication
using NameID mapping for name identifier: alice.appleton@oracle.com
name identifier format :
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress and message :
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
Destination="http://adc00pcc.us.oracle.com:23002/oam/server/fed/sp/
sso" ID="idaWfL5-
f37nhQWh0WWjHbobsVetM-" InResponseTo="id-hqkZGMV-wEO5-
CulpYxArIvr91Y14dA-WSRYZ8zP" IssueInstant="2014-02-28T15:18:05Z"
Version="2.0"><saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
Format="urn:oasis:names:tc:SAML:2.0:nameidformat:
entity">http://adc00peq.us.oracle.com:7499/fed/idp</saml:Issuer>
<samlp:Status><samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/></
samlp:Status><saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="id-
PoODBDUeoiSY4ajPCQ86yjZWkw-"
IssueInstant="2014-02-28T15:18:05Z" Version="2.0">
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameidformat:
entity">http://adc00peq.us.oracle.com:7499/fed/idp</saml:Issuer>
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/
xmldsig#"><dsig:SignedInfo>
<dsig:CanonicalizationMethod 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-PoOD-
BDUeoiSY4ajPCQ86yjZWkw-"><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:DigestMethod Algorithm="http://www.w3.org/
2000/09
/xmldsig#sha1"/><dsig:DigestValue>X5ojFxrpbOS4klosM5jcBOF8Bqg=
</dsig:DigestValue></dsig:Reference></dsig:SignedInfo>
<dsig:SignatureValue>VJKJOB0owHZ41VkJHjX4w2YHi+0ZAa4ez
/
+D+ketAQcOxxtwOZPcBYERwkMgazudMh0XEMbIkwsBTVwb4tX+uV327Gjlp1hXc0uYnm2n8
mZfen9Ppru6jTES4N7PoD3mOpCfFEHBUJg118DihWGLfzBWw7LMLaN2A
</dsig:SignatureValue></dsig:Signature><saml:Subject><saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameidformat:
emailAddress">alice.appleton@oracle.com</saml:NameID>
<saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
<saml:SubjectConfirmationData InResponseTo="id-hqkZGMV-wEO5-
CulpYxArIvr91Y14dA-WSRYZ8zP" NotOnOrAfter="2014-02-28T15:23:05Z"
```



```
Recipient="http://adc00pcc.us.oracle.com:23002/oam/server/fed/sp/sso"/>
</saml:SubjectConVrmation></saml:Subject><saml:Conditions
NotBefore="2014-02-28T15:18:05Z" NotOnOrAfter="2014-02-28T15:23:05Z">
<saml:AudienceRestriction><saml:Audience>http://adc00pcc.us.oracle.com:23002
/oam/fed</saml:Audience></saml:AudienceRestriction></saml:Conditions>
<saml:AuthnStatement AuthnInstant="2014-02-28T15:18:05Z" SessionIndex="id-
2i7BYlgGnhukoBSDmrvkBIaG-NI-" SessionNotOnOrAfter="2014-02-28T16:18:05Z">
<saml:AuthnContext>
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordPro
tectedTransport</saml:AuthnContextClassRef>
</saml:AuthnContext></saml:AuthnStatement></saml:Assertion></samlp:Response>
```

32.1.2.3.2 Response Validation Issues

If the incoming SSO assertion cannot be validated the Test SP application displays the following information.

- An error message
- The SSO message sent by the IdP

In the following example, the IdP and OIF/SP administrators agreed to use SAML 2.0 but the IdP is not signing the assertion as required by OIF/SP.

Federation SSO Operation Result

```

SSO Authentication Result Authentication Failed
User Identifier
Authentication Instant
SSO Primary Status Code RESPONDER
SSO Secondary Status Code
SSO Status Message The assertion could not be validated
Partner acmeloP
    
```

Attributes from the Assertion

Assertion Message

```

<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" Destination="http://adc00pcc.us.
De7M27k5CWpBsuGzgxwHgwqV1g-" InResponseTo="id-fX4nHKLCMcA-ZjHvskfCORDZLmDcQMpVYjgm
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Format="urn:oasis:names:tc:SAML
format:entity">http://adc00peq.us.oracle.com:7499/fed/idp</saml:Issuer>
<samlp:Status>
<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="id-EAdQSXj-royYNuuWbaB
28T15:23:05Z" Version="2.0">
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">http://adc00peq.us.oracle
<saml:Subject>
<saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">alice@oracle
<saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
<saml:SubjectConfirmationData InResponseTo="id-fX4nHKLCMcA-ZjHvskfCORDZLmDcQMpVY
28T15:28:05Z" Recipient="http://adc00pcc.us.oracle.com:23002/oam/server/fed/sp/sso"/>
</saml:SubjectConfirmation>
</saml:Subject>
<saml:Conditions NotBefore="2014-02-28T15:23:05Z" NotOnOrAfter="2014-02-28T15:28:05Z">
<saml:AudienceRestriction>
<saml:Audience>http://adc00pcc.us.oracle.com:23002/oam/fed</saml:Audience>
</saml:AudienceRestriction>
    
```

At the end of the flow, the Test SP application displays the following information in the Federation SSO Operation result:

- The authentication operation failed message.
- The assertion could not be validated message.
- The SAML message.

The OIF log files display the following error message and the the SAML message.

```

<Feb 28, 2014 7:23:05 AM PST> <Error>
<oracle.security.fed.eventhandler.fed.profiles.utils.CheckUtils>
<FEDSTS-18003>
<Assertion is not signed.>
<Feb 28, 2014 7:23:05 AM PST> <Error>
<oracle.security.fed.eventhandler.fed.profiles.sp.sso.v20.ProcessResponseEvent
seEventHandler>
<FED-18012> <Assertion cannot be validated: <samlp:Response
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
Destination="http://adc00pcc.us.oracle.com:23002/oam/server/fed/sp/
    
```

```
sso" ID="id-
De7M27k5CWpBsuGzgxwHgwqV1g-" InResponseTo="id-
fx4nHKLCMcAZjHvsKfCORDZLmIDcQMpVYjqmxQb"
IssueInstant="2014-02-28T15:23:05Z"
Version="2.0"><saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
Format="urn:oasis:names:tc:SAML:2.0:nameidformat:
entity">http://adc00peq.us.oracle.com:7499/fed/idp</saml:Issuer>
<samlp:Status><samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/></
samlp:Status><saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="id-
EAdQSXjroyYNuuWbaBWZVdBtu8-"
IssueInstant="2014-02-28T15:23:05Z" Version="2.0">
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameidformat:
entity">http://adc00peq.us.oracle.com:7499/fed/idp</saml:Issuer>
<saml:Subject><saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameidformat:
emailAddress">alice@oracle.com</saml:NameID><saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"><saml:SubjectConfirmationData
InResponseTo="id-fx4nHKLCMcA-ZjHvsKfCORDZLmIDcQMpVYjqmxQb"
NotOnOrAfter="2014-02-28T15:28:05Z"
Recipient="http://adc00pcc.us.oracle.com:23002/oam/server/fed/sp/sso"/>
</saml:SubjectConfirmation></saml:Subject><saml:Conditions
NotBefore="2014-02-28T15:23:05Z" NotOnOrAfter="2014-02-28T15:28:05Z">
<saml:AudienceRestriction><saml:Audience>http://adc00pcc.us.oracle.com:23002
/oam/fed</saml:Audience></saml:AudienceRestriction></saml:Conditions>
<saml:AuthnStatement AuthnInstant="2014-02-28T15:23:05Z" SessionIndex="id--
0QWpaU2AV-L7UpNvLH5Bn7Z5Xk-" SessionNotOnOrAfter="2014-02-28T16:23:05Z">
<saml:AuthnContext>
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordPro
tectedTransport</saml:AuthnContextClassRef>
</saml:AuthnContext></saml:AuthnStatement></saml:Assertion></samlp:Response>
```

32.2 Using Federation Attributes for OAM Authorization and Protected Web Applications

This chapter describes how attributes received in SAML/OpenID SSO messages can be used in the Oracle Access Manager (OAM) authentication process and how they can be provided to protected web applications.

At runtime, when Oracle Identity Federation (OIF)/ Service Provider(SP) successfully processes a SAML/OpenID SSO response message, the server saves some of the information from the response in the OAM session as attributes and can be used in OAM authorization policies:

- As conditions in authorization rules
- As responses to provide the SAML/OpenID attributes to protected web applications

The SAML/OpenID SSO Response information is saved in the OAM session as attributes referenced by the following identifiers:

- The IdP partner name, referenced by `$session.attr.fed.partner`.
- The NameID value from the SSO response, referenced by `$session.attr.fed.nameidvalue`.

- The NameID format from the SSO response for SAML protocols referenced by `$session.attr.fed.nameidformat`.
- Attributes contained either in the SAML Assertion's Attribute Statement or in the OpenID SSO Response are referenced using `$session.attr.fed.attr.ATTR_NAME`, with `ATTR_NAME` being either the local session attribute name, if the IdP Attribute Profile mapping is applied or the attribute name from the SSO response, if no IdP Attribute Profile mapping is applied for this attribute.

32.2.1 Overview of Authenticating User Access to a Protected Resource

Oracle Access Management environment is made of the following components:

- LDAP directory
- OAM admin server, with the OAM admin console
- OAM runtime server
- Web applications
- WebGate agents protecting web applications on HTTP servers (OHS, IIS, and so on)

The WebGate performs the following tasks when an authenticated user requests access to a protected resource. Interprets the call and ensures that the:

- Interprets the call and ensures that the:
 - User is authenticated
 - User is authorized to access the resource by evaluating authorization policies for the resource
- Injects data as cookies or HTTP headers into the HTTP request, and forwards the HTTP request to the protected resource

The following are the various conditions that the OAM Authorization Policies consider when determining whether a user can access a resource:

- **Identity:** Condition based on the user's identity or groups to which the user belongs
- **IP Address:** Condition based on the user's IP address
- **Temporal:** Condition based on time
- **Attributes:** Condition based on attributes (LDAP, HTTP request, or session attributes)

Following are the components based on which the OAM Authorization Responses inject data into the HTTP request to make it available for protected web applications:

- User LDAP attributes
- HTTP request data
- Static strings
- OAM session attributes

Similar to OAM Authization Policies, an administrator can inject federation data into an HTTP request using OAM session attributes

```
($session.attr.fed.partner, $session.attr.fed.attr.ATTR_NAME...)
```

32.2.2 Prerequisites for Setting up Federation SSO

Following are the requirements for setting up Federation SSO:

- OIF acting as a Service Provider
- The IdP (AcmeIdP) sending a SAML assertion with NameID set to userID
- Set the following attributes:
 - email to user's email address
 - fname to user's first name
 - surname to user's last name
 - title to user's last job title
- Configure OIF/SP with an IdP attribute profile to map:
 - fname to firstname
 - surname to lastname
 - leave email as is

Configure two users with the following values:

- User 1: Alice
 - userID: alice
 - email: alice@oracle.com
 - first name: Alice
 - last name: Appleton
 - title: manager
- User 2: Bob
 - userID: bob
 - email: bob@oracle.com
 - first name: Bobby
 - last name: Smith
 - title: engineer

The XML SAML response with the assertion sent back by the IdP must be as follows.

```
<samlp:Response ..>
<saml:Issuer ...>http://acme.com/idp</saml:Issuer>
<samlp:Status>
<samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
<saml:Assertion ...>
<saml:Issuer ...>http://acme.com/idp</saml:Issuer>
```

```
<dsig:Signature ...>
...
</dsig:Signature>
<saml:Subject>
<saml:NameID ...>alice</saml:NameID>
...
</saml:Subject>
<saml:Conditions ...>
...
</saml:Conditions>
<saml:AuthnStatement ...>
...
</saml:AuthnStatement>
<saml:AttributeStatement ...>
<saml:Attribute Name="email" ...>
<saml:AttributeValue...>alice@oracle.com</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute Name="title" ...>
<saml:AttributeValue...>manager</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute Name="surname" ...>
<saml:AttributeValue...>Appleton</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute Name="fname" ...>
<saml:AttributeValue...>Alice</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>
```

The Test SP page displays different results, as the OIF/SP processes the attributes according to the rules where:

- email was not changed
- title was not changed
- fname was mapped to firstname
- surname was mapped to lastname

Federation SSO Operation Result

SSO Authentication Result Authentication Successful
User Identifier oud-e2e:USER:cn=alice,ou=users,dc=us,dc=oracle,dc=com:alice
Authentication Instant Mon Mar 10 14:33:21 PDT 2014
SSO Primary Status Code SUCCESS
SSO Secondary Status Code
SSO Status Message
Partner AcmeldP

Attributes from the Assertion

fed.partner [AcmeldP]
title [manager]
fed.nameidformat [urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified]
email [alice@oracle.com]
fed.nameidvalue [alice]
lastname [Appleton]
firstname [Alice]

Assertion Message

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" Destination="http://adc00pcc  
FWpwDuf9QvOL4BHsJV3zrlFSn0NdAWAIK9IY" IssueInstant="2014-03-10T21:33:21Z" Version="2.0">  
  <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Format="urn:oasis:names:tc:S
```

32.2.3 Prerequisites for Protected Web Application

Ensure that the following components are configured.

- OHS is installed
- A WebGate agent must be configured for the OHS instance
- An OAM Application Domain must be created for the WebGate, which protects all the resources on the OHS server
- Authentication Policy:
 - Name: Protected Resource Policy
 - Authentication Scheme: FederationScheme
- Authorization Policy:
 - Name: Protected Resource Policy
 - Resources linked to 'Protected Resource Policy' of Authentication Policy and 'Protected Resource Policy' of the Authorization Policy

The `/cgi-bin/printenv` resource on OHS prints the following data when processing the HTTP Request sent by the browser:

- HTTP Headers
- Request Data (path, query string)
- Server Data (IP address, port)

An example of a browser accessing the resource without being protected by OAM/ WebGate would result in the following display (in the test, the web application will be protected as listed below):

```
DOCUMENT_ROOT=
GATEWAY_INTERFACE="CGI/1.1"
HTTP_ACCEPT="application/x-ms-application, image/jpeg, application/xaml+xml"
HTTP_ACCEPT_ENCODING="gzip, deflate"
HTTP_ACCEPT_LANGUAGE="en-US"
HTTP_CONNECTION="Keep-Alive"
HTTP_COOKIE="OAMAuthnHintCookie=0@1394486789; OAM_ID=VERSION_4~+wyj1KoQ1"
HTTP_ECID_CONTEXT="1.004x4T2GEHKFg4ALJaK6yf0000px00001t;kajE1ZDLIPGIj3RE"
HTTP_HOST=
HTTP_OAM_IMPERSONATOR_USER=""
HTTP_OAM_LAST_REAUTHENTICATION_TIME="Mon Mar 10 14:26:38 PDT 2014"
HTTP_OAM_REMOTE_USER="Anonymous"
HTTP_USER_AGENT="Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64)"
PATH="/bin:/usr/bin:/usr/ucb:/usr/bsd:/usr/local/bin"
QUERY_STRING=""
REMOTE_ADDR=
REMOTE_PORT="62551"
REQUEST_METHOD="GET"
REQUEST_URI="/cgi-bin/printenv"
SCRIPT_FILENAME=""
SCRIPT_NAME="/cgi-bin/printenv"
SERVER_ADDR=
SERVER_ADMIN="[no address given]"
SERVER_NAME=
SERVER_PORT=
SERVER_PROTOCOL="HTTP/1.1"
SERVER_SIGNATURE=""
SERVER_SOFTWARE="Oracle-Application-Server-11g"
TZ="PST8PDT"
UNIQUE_ID="Ux4uDgrkwrMAAA2FNc0AAAAAS"
```

32.2.4 Constructing Authorization Policy Using Federation Attributes

The following example shows how to construct an Authorization Policy using Federation attributes stored in the OAM session for a resource with the following constraints:

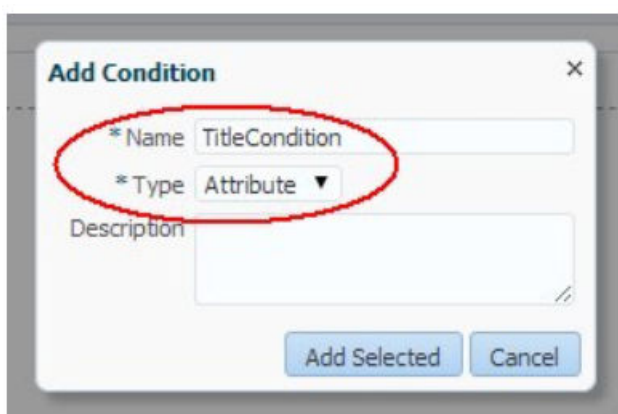
- Users authenticated through Federation SSO (The resource is protected through FederationScheme Authentication Policy).
- IdP provides the job title of the user and is locally known as title (if an IdP sends the job title through a name other than the title then an IdP Attribute Profile must be used to map it to the local title name).
- Only users with the manager title must have access to the resource.

Following are the steps to create an authorization policy:

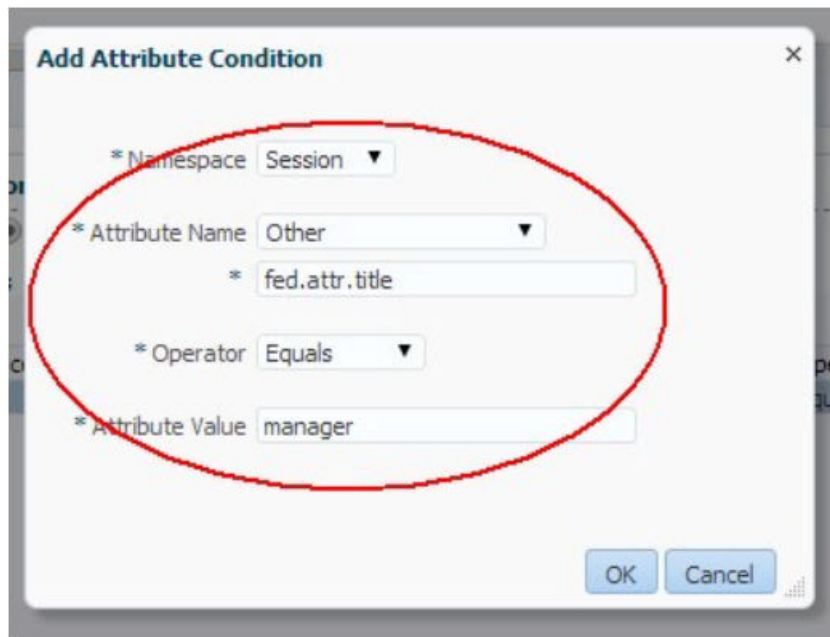
1. Go to the OAM Administration Console:

<https://oam-admin-host:oam-adminport/oamconsole>

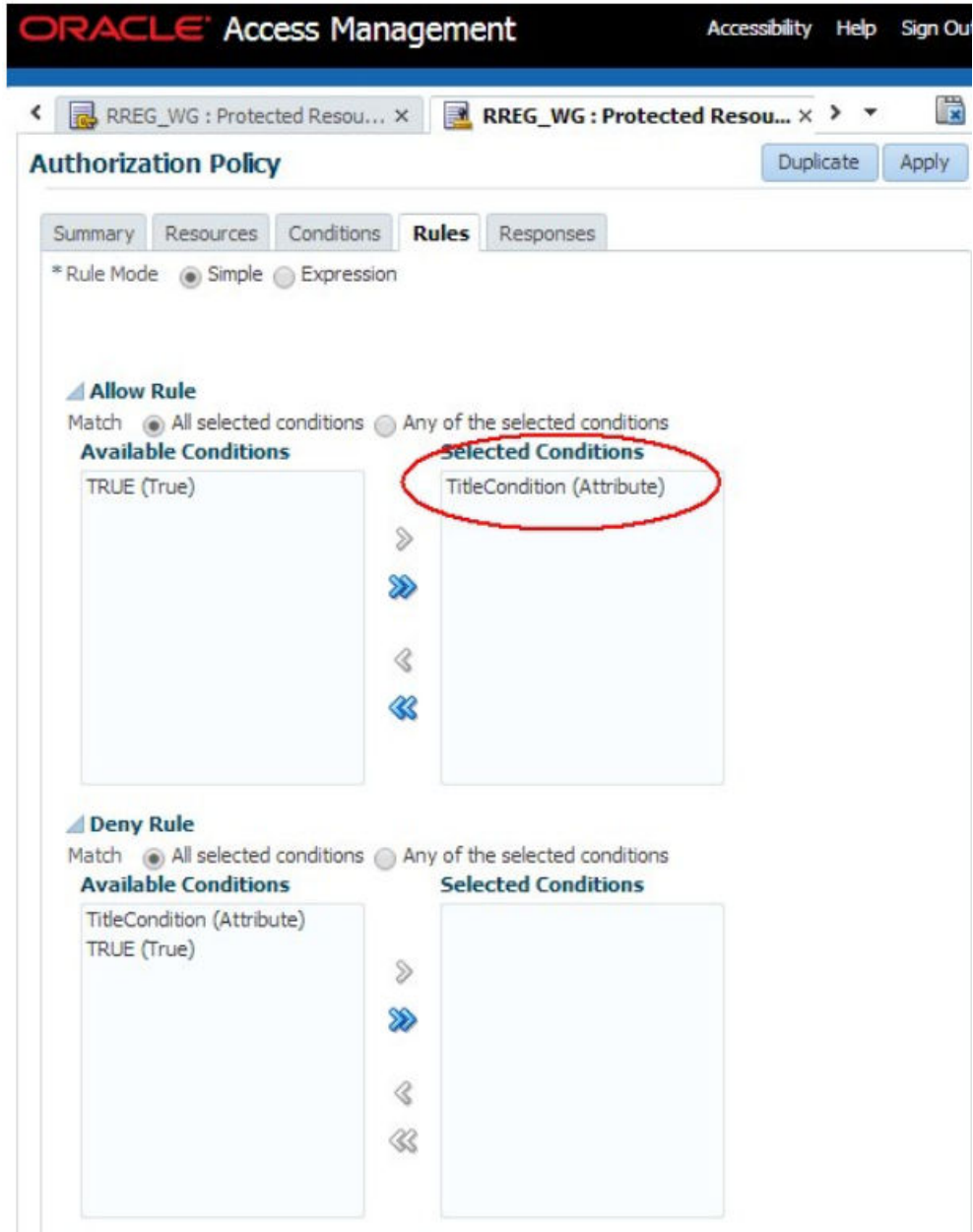
2. Navigate to **Access Manager , Application Domains**.
3. Search and click **Application Domain** for the resource.
4. Click **Authorization Policies**.
5. Open the Authorization Policy protecting the resource (**Protected Resource Policy** in this example)
6. Click **Conditions** tab.
7. Click **Add** to define a new condition and select the following values:
 - Name: TitleCondition
 - Type: Attribute
8. Click **Add Selected**.



9. Select the newly created condition.
10. In the Condition Details window, click **Add** and select the following values.
 - Namespace: Session
 - Attribute Name: Other
 - Enter the attribute name: fed.title
 - Operator: Equals
 - Attribute Value: Manager



11. Click **OK**.
12. Click **Rules** tab.
13. Remove the `TRUE` condition, if present in the **Allow Rule, Selected Conditions**.
14. Add the `TitleCondition` to the **Allow Rule, Selected Conditions**.
15. click **Apply**.



To test, open a new browser and access the protected resource. You will be redirected to the IdP.

If you authenticate at the IdP as `alice` then the browser displays the following information at the end of the flow showing the Remote User HTTP header set to `alice` (Since at IdP the title attribute is set to `manager`, the OAM only allows access to the users with the OAM session attribute `fed.title` set to `manager`).

```

DOCUMENT_ROOT=
GATEWAY_INTERFACE="CGI/1.1"
HTTP_ACCEPT="application/x-ms-application, image/jpeg, application/xam
HTTP_ACCEPT_ENCODING="gzip, deflate"
HTTP_ACCEPT_LANGUAGE="en-US"
HTTP_CACHE_CONTROL="no-cache"
HTTP_CONNECTION="Keep-Alive"
HTTP_COOKIE="OAMAuthnHintCookie=0@1394488467; OAM_JSESSIONID=FXmxTp0JR
HTTP_ECID_CONTEXT="1.004x4Ua0f5TFg4ALJaK6yf00000px00003I;kZjE1ZDLIPGIj3
HTTP_HOST=
HTTP_OAM_IDENTITY_DOMAIN="oud-e2e"
HTTP_OAM_IMPERSONATOR_USER=""
HTTP_OAM_LAST_REAUTHENTICATION_TIME="Mon Mar 10 14:54:31 PDT 2014"
HTTP_OAM_REMOTE_USER="alice"
HTTP_REFERER=
HTTP_USER_AGENT="Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WO
PATH="/bin:/usr/bin:/usr/ucb:/usr/bsd:/usr/local/bin"
QUERY_STRING=""
REMOTE_ADDR=
REMOTE_PORT="62626"
REQUEST_METHOD="GET"
REQUEST_URI="/cgi-bin/printenv"
SCRIPT_FILENAME=
SCRIPT_NAME="/cgi-bin/printenv"
SERVER_ADDR=
SERVER_ADMIN="[no address given]"
SERVER_NAME=
SERVER_PORT=
SERVER_PROTOCOL="HTTP/1.1"
SERVER_SIGNATURE=""
SERVER_SOFTWARE="Oracle-Application-Server-11g"
TZ="PST8PDT"
UNIQUE_ID="Ux401wrkwrMAAA2FNf0AAAAM"

```

If you authenticate at the IdP as bob, the browser displays an error at the end of the flow (since at IdP the title attribute is set to engineer, the OAM only allows access to users with the OAM session attribute `fed.title` set to engineer).

Oracle Access Manager Operation Error

Access to the URL `/cgi-bin/printenv` has been denied for user.

Contact your website administrator to remedy this problem.

32.2.5 Injecting Federation Attributes

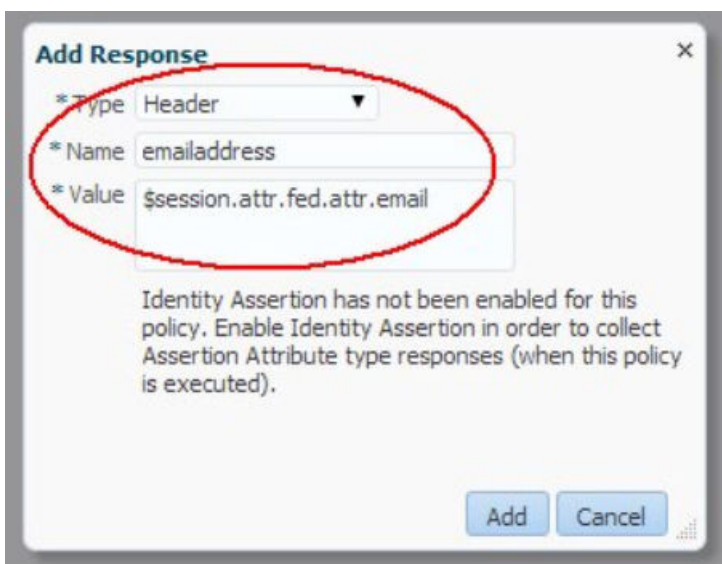
The following example shows how to inject SAML/OpenID attributes collected from the SSO Response as HTTP Headers for the protected Web with the following constraints:

- Users authenticated through Federation SSO (The resource is protected through a FederationScheme Authentication Policy).
- IdP provides the job title of the user and is locally known as title (if an IdP sends the job title through a name other than the title then an IdP Attribute Profile must be used to map it to the local title name).
- OAM/WebGate is configured to inject:

- Email address as emailaddress
- First name as firstname
- Last name as lastname
- The configuration is done through the use of Authorization Response objects in an Authorization Policy definition.

Following are the steps to inject Federation attributes:

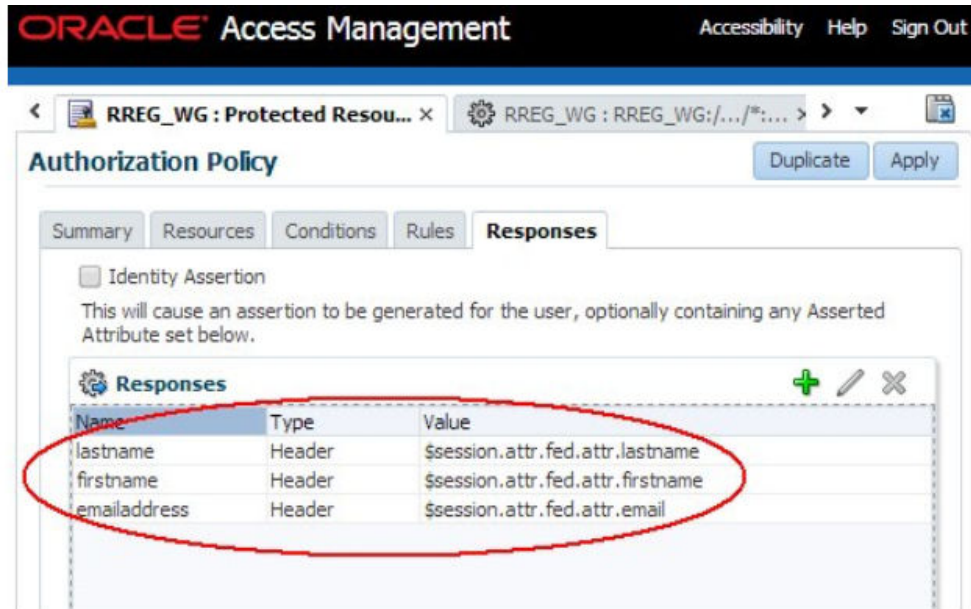
1. Go to the OAM Administration Console:
`http(s)://oam-admin-host:oam-adminport/oamconsole`
2. Navigate to **Access Manager** , **Application Domains**.
3. Search and click **Application Domain** for the resource.
4. Open the Authorization Policy protecting the resource (**Protected Resource Policy** in this example)
5. Click **Responses** tab.
6. Click **Add** to create the entry for the email address:
 - Type: Header
 - Name: emailaddress
 - Value: `$session.attr.fed.attr.email`
7. Click **Add** to add a response.



8. Click **Add** to create the entry for the first name:
 - Type: Header
 - Name: firstname
 - Value: `$session.attr.fed.attr.Lrstname`
9. Click **Add**.
10. Click **Add** to create the entry for the last name:

- Type: Header
- Name: lastname
- Value: `$session.attr.fed.attr.lastname`

11. Click **Add**.
12. Click **Apply** to save the values.



To test, open a new browser and access the protected resource. You will be redirected to the IdP where the authentication occurs.

OAM/WebGate then injects the Authorization Response items based on the OAM Session attributes (received from the IdP) and the protected web application displays them (my test page displays an HTTP header as `HTTP_NAME`, where `NAME` is the name of the HTTP Header).


```

DOCUMENT_ROOT=
GATEWAY_INTERFACE="CGI/1.1"
HTTP_ACCEPT="application/x-ms-application, image/jpeg, application,
HTTP_ACCEPT_ENCODING="gzip, deflate"
HTTP_ACCEPT_LANGUAGE="en-US"
HTTP_CACHE_CONTROL="no-cache"
HTTP_CONNECTION="Keep-Alive"
HTTP_COOKIE="OAMAuthnHintCookie=0@1394489512; OAM_JSESSIONID=CYYzT;
HTTP_COOKIE_CONTEXT=1.004XWAGDumEg4ALJaK6yf0000px00004P;kZjE1ZDLIP(
HTTP_EMAILADDRESS="alice@oracle.com"
HTTP_FIRSTNAME="Alice"
HTTP_HOST=
HTTP_LASTNAME="Appleton"
HTTP_OAM_IDENTITY_DOMAIN="oud-e2e"
HTTP_OAM_IMPERSONATOR_USER=""
HTTP_OAM_LAST_REAUTHENTICATION_TIME="Mon Mar 10 15:11:55 PDT 2014"
HTTP_OAM_REMOTE_USER="alice"
HTTP_REFERER=
HTTP_USER_AGENT="Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1,
PATH="/bin:/usr/bin:/usr/ucb:/usr/bsd:/usr/local/bin"
QUERY_STRING=""
REMOTE_ADDR=
REMOTE_PORT="62669"
REQUEST_METHOD="GET"
REQUEST_URI="/cgi-bin/printenv"
SCRIPT_FILENAME=
SCRIPT_NAME="/cgi-bin/printenv"
SERVER_ADDR=
SERVER_ADMIN="[no address given]"
SERVER_NAME=
SERVER_PORT="23777"
SERVER_PROTOCOL="HTTP/1.1"
SERVER_SIGNATURE=""
SERVER_SOFTWARE="Oracle-Application-Server-11g"
TZ="PST8PDT"
UNIQUE_ID="Ux44qwrkwrMAAA2FNgsAAAAV"

```

32.3 Key and Certificate Management and Rollover in OIF and OSTs

This chapter describes the following concepts in the key and certificate management and rollover.

- Generating new keys and certificates.
- Configuring Oracle Identity Federation (OIF) and Oracle Secure Token Service (OSTS) to use the new keys and certificates.
- Implementing a key rollover on a per partner basis.
- Distributing the new certificates to partners.

32.3.1 Introduction to Key and Certificate Management and Rollover

Oracle Identity Federation (OIF)/Oracle Secure Token Service (OSTS) uses Public Key Infrastructure (PKI) Keys and Certificates as part of the Federation and WS-Trust protocol

interaction for providing non-repudiation and integrity through the use of digital signatures, and confidentiality through digital encryption.

The following occurs during a Federation/WS-Trust exchange:

- OIF/OSTS uses its own PKI Keys and Certificates to perform signature and decryption operations on the SAML messages:
 - Signing outgoing SAML messages and Assertions is done by using XML Digital signatures or Query String signatures.
 - Decryption of incoming SAML Assertions is done by using XML Digital encryption.
- OIF/OSTS uses the partner's signing or encryption certificate to:
 - Verify signatures on incoming SAML messages and Assertions (XML Digital encryption)
 - Optionally encrypt outgoing SAML Assertions (XML Digital encryption)

Following is an example of SAML messages with XML Digital signatures.

```
<samlp:Response ...>
<saml:Issuer>hUps://idp.com</saml:Issuer>
<samlp:Status>...</samlp:Status>
<saml:Assertion ...>
<saml:Issuer>https://idp.com</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
excc14n#" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1" />
<ds:Reference URI="#idhmf9KzAhxleuJ-L3vaVr979Ffa0">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#envelopedsignature" />
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>JGvBqil/NXA6dlMOn5ZhmBb0ie8=</DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>VgOrU79ZJO4rzHiFTCDGnmkb0...Y776QM4vEBBybIpbCCUih7I
0aA==
</ds:SignatureValue>
</ds:Signature>
<saml:Subject>
<saml:NameID>alice@acme.com</saml:NameID>
...
</saml:Subject>
...
</saml:Assertion>
</samlp:Response>
```


Following is an example of SAML message with query string signature (typically used by an SP to send the user to the IdP with an AuthnRequest).

```
https://idp.com
/saml20sso?
SAMLRequest=hZJRT8IwFIX%2FStP3sW5BTW7YEhRREtQFplHeytZBQ9fw3i7Kv3cUTdAHfL09t98
5J3eEvFUWxp3f6o
V47wR68tkqjRAeMto5DYajRNC8FQi%2BguX4YQ7pgIFlpxvKKEom%2FZ7U3EujM7r13iLEsaztoDI
tJVPjKhEQGW24QkHJbJJRWUfPu
LtTFyuuS7bbsLVcpK92OmdN8XYb9SLETsw0eq59RlOWDCOWRikrkytIhsAuV5QU3x6upa6l3pw3vD
6KEO7LsoiKp2VJyYtwGGz3ApqP
DrEhgN1JEee%2F5YjCHbKHqC335%2BWHSZ%2B9CVIQ2ku%2Fp%2F1PaxhKG8UnRo6uLDz2i7NJYZS
s9mSslPm4cYJ7kVHvOvE%2FPBk
kf%2BCdRisq2Uhr0zg%2FQn9fQ%2F4F&RelayState=id--
mAKlwhfUGrvoLqghU2ysXLWSIw-&SigAlg=hUp%3A%2F
%2Fwww.w3.org%2F2000%2F09%2Fxmldsig%23rsa-sha1&
Signature=S5TZ0uwK9SMZUgBfDaipbNhlLqbbSG9t4rgA9n3%2FwxFsK7H66IoK6G%2BDfaIUvc5
bLtTrwmxa2iB2gjF8p5
Q6%2FgH8OtFbT7mKZ7z8FihgxxTKjHJ2FQocOEn%2FrkcRKAaq%2Blig5xVS1R%2BzLq1vkQzIMNO
rfLw%2FM6uk3i%2Fk54EnQ%3D
SAMLRequest: SAML AuthnRequest message
RelayState: SAML 2.0 Relay State parameter
SigAlg: signature algorithm
Signature: signature bytes covering SAMLRequest, RelayState and SigAlg
parameters
```

The OIF/OSTS PKI keys and certificates are stored in the `$(DOMAIN_HOME)/config/fmwconfig/.oamkeystore` Java Keystore file (This keystore is of type JCEKS, not JKS), and the OIF/OSTS settings reference the key entries in the `.oamkeystore`, so that they can be used by the components for SAML operations.

32.3.2 Generating Keys and Certificates

During the installation phase of Oracle Access Manager (OAM), a key pair and self signed certificate gets generated and OIF/OSTS is configured to use it for signing and decryption operations. Along with the key pair and the self signed certificate generation the following changes occur.

- The OAM installer creates the `.oamkeystore` with a random password. For more information on how to reset the password, see [Setting New Key Entries](#) section.
- A new key entry called `stsprivatekeyalias` is created.
- RSA Key Pair is created.
- Self signed certificate is created.
- Subject and Issuer is set to: `cn=<MACHINE_HOSTNAME>`.
- The following two entries are created in the OIF/OSTS configuration:
 - `osts_signing` referencing the `stsprivatekeyalias` key entry in the `.oamkeystore`.
 - `osts_encryption` referencing the `stsprivatekeyalias` key entry in the `.oamkeystore`.
- OIF is set up to use `osts_signing` entry for signature operations and `osts_encryption` for decryption operations.

- OSTS is set up to use `osts_encryption` for decryption operations, and `osts_signing` for signature operations in the SAML Issuance Templates.

32.3.3 Setting New Key Entries

The process of creating new PKI Keys and Certificates before using them in OIF/OSTS is a two step process.

1. Create key entry in `.oamkeystore`
2. Create entry in OIF/OSTS to refer the key entry in `.oamkeystore`

32.3.3.1 Creating New Key Entry in `.oamkeystore`

The password to the `.oamkeystore` Keystore is unknown to the administrator and must be reset to make modifications. Use the following WebLogic Scripting Tool (WLST) commands to reset the password.

1. Enter into the WLST environment by executing:

```
$IAM_ORACLE_HOME/common/bin/wlst.sh
```

2. Connect to the WLS Admin server:

```
connect()
```

3. Navigate to the Domain Runtime branch:

```
domainRuntime()
```

4. Reset the `.oamkeystore` password:

```
resetKeystorePassword()
```

5. Exit WLST environment:

```
exit()
```

One way to create a new key entry in the `.oamkeystore` is to use the JDK's KeyTool application. In this example, two key entries with self signed certificates will be created, one with the alias `samlsigning` and the other with the alias `samlencryption` (replace `$JDK_HOME` and `$DOMAIN_HOME` with the correct path; for the keystore password enter the one you selected during the reset operation):

```
$JDK_HOME/bin/keytool -genkeypair -alias samlsigning -keyalg RSA -  
keysize 2048  
-sigalg sha1withrsa -dname cn="ACME SAML Signing" -validity 1000 -  
keystore  
$DOMAIN_HOME/conEg/fmwconEg/.oamkeystore -storetype JCEKS
```

```
$JDK_HOME/bin/keytool -genkeypair -alias samlencryption -keyalg RSA -  
keysize  
2048 -sigalg sha1withrsa -dname cn="ACME SAML Encryption" -validity
```

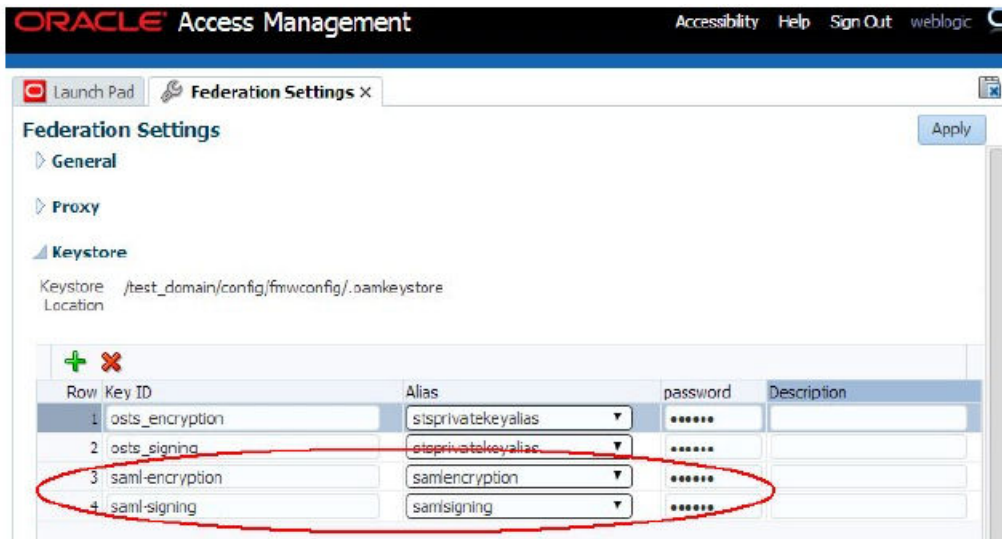
```
1000
-keystore $DOMAIN_HOME/conEg/fmwconEg/.oamkeystore -storetype JCEKS
```

32.3.3.2 Updating OIF and OSTs Settings

To use the new key entries created in the `.oamkeystore` during SAML protocol exchanges, you have to create new SAML key entries in the Oracle Identity Federation (OIF)/Oracle Secure Token Service (OSTS).

Following are the steps to create a new SAML key entry in OIF/OSTS.

1. Login to the OAM Administration Console: `http(s)://oam-admin-host:oam-adminport/oamconsole`.
2. Navigate to **Configuration**, **Federation Settings** or **Security Token Service Settings**.
3. In the Keystore section, click the **+** button.
4. Enter a KeyID for the new entry (for example, `saml-signing`).
5. Select the alias for the new key entry from the dropdown, which lists the key entries in the `.oamkeystore` (for example, `saml signing`).
6. Enter the password for the key entry that you set when creating that key.
7. Repeat the process for other entries, if needed.
8. click **Apply**.



Note:

Different key IDs can be referenced to the same key entry in the OAM Keystore.

32.3.4 Using New Key Entries to Update Global Settings

You must update the following settings to use new keys and certificates to sign and decrypt SAML messages.

- Global Oracle Identity Federation (OIF) Settings
- Global Oracle Secure Token Service (OSTS) Settings
- Oracle Secure Token Service (OSTS) Settings

32.3.4.1 Updating Global OIF Settings

To update global Oracle Identification Federation (OIF) settings perform the following steps:

1. Login to the OAM Administration console: `https://oam-admin-host:oam-adminport/oamconsole`
2. Navigate to **Configuration , Federation Settings**.
3. Select the Signing Key from the dropdown list of key entries (these entries are defined in the Keystore section). For example, select `saml-signing`
4. Select the Encryption Key from the dropdown list of key entries (these entries are defined in the Keystore section). For example, select `saml-encryption`
5. Click **Apply**.

 **Note:**

After applying the settings, you must Redistribute certificates and/or SAML 2.0 metadata to partners.

32.3.4.2 Updating Global OSTS Settings

To update global Oracle Secure Token Service (OSTS) settings perform the following steps:

1. Login to the OAM Administration console: `https://oam-admin-host:oam-adminport/oamconsole`
2. Navigate to **Configuration , Security Token Service Settings**.
3. Select the Default Encryption Template from the dropdown list of key entries (these entries are defined in the Keystore section). For example, select `saml-encryption`
4. Click **Apply**.

 **Note:**

After applying the settings, you must Redistribute certificates to partners.

32.3.4.3 Updating OSTS Settings

To update Oracle Secure Token Service (OSTS) settings perform the following steps:

1. Login to the OAM Administration console: `https://oam-admin-host:oam-adminport/oamconsole`
2. Navigate to **Security Token Service , Token Issuance Templates**.

3. Click **SAML Issuance Template** that you want to update.
4. Click **Security** tab.
5. Select the Signing Keystore Access Template Id from the dropdown list of key entries (these entries are defined in the Keystore section). For example, select `saml-signing`.
6. Click **Apply**.

 **Note:**

After applying the settings, you must redistribute certificates to the partners.

32.3.5 Managing Key Rollover per Partner

When an OIF/OSTS deployment involves a large number of partners, it might be difficult to adjust the global signing/encryption keys/certificates at once, since notifying all of the partners at the same time of the changes and updating their configurations to reflect the new certificates/SAML 2.0 metadata might be difficult.

 **Note:**

Once the keys/certificates in OIF/OSTS have been updated, the Federation/WS-Trust flows will not work with the partners until they upload the new certificates into their systems.

With OIF/OSTS, the administrator can perform key rollovers per partner. This allows them to plan when and how to notify specific partners of key and certificate changes.

Perform the following steps to key rollover for OIF IdP or SP partners:

1. Set up new keys and certificates. For more information on setting up new keys and certificates, see [Setting New Key Entries](#).
2. Update the IdP or SP partner configuration in OIF to use the new keys and certificates.
3. Notify the partner with new SAML 2.0 Metadata generated specifically for those new keys/certificates or new certificates corresponding to the new key entries.

Perform the following steps to key rollover for OSTs Relying Party partners:

1. Set up new keys and certificates. For more information on setting up new keys and certificates, see [Setting New Key Entries](#).
2. If not already one:
 - Create a new Relying Party Profile, which is a copy of the current Relying Party Profile of the Relying Party partner.
 - Create a new SAML Issuance Template, which is a copy of the SAML Issuance Template referenced by the current relying party Profile used by the relying party partner.
 - Update the new Relying Party Profile to use a new SAML Issuance Template instead of the current one.
 - Update the new SAML Issuance Template to use the new keys/certificates.

- Assign the Relying Party partner to the new Relying Party Profile.



Note:

Partner Profiles in the OIF configuration can be used to rollover OIF keys across a group of partners.

32.3.6 Managing OIF Key Rollover

The following steps are involved in performing Oracle Identity Federation (OIF) key rollover for a specific partner.

1. Update the Identity Provider (IdP) or Service Provider (SP) partner configuration in OIF via WebLogic Scripting Tool (WLST) commands.
2. Use the SAML 2.0 metadata or certificate information.

To update IdP or SP partner configuration in OIF via WLST commands perform the following steps.

1. Enter the WLST environment: `$IAM_ORACLE_HOME/common/bin/wlst.sh`
2. Connect to the WLS Admin server: `connect()`
3. Navigate to the Domain Runtime branch: `domainRuntime()`
4. Update the partner configuration to set the Signing Key property (referenced by `signingkeystoreaccesstemplateid`) to the key entry ID defined in the **Federation Settings, Keystore** section.
In this example, `saml-signing` is the key entry ID.

Replace `<PARTNER_NAME>` by the name of the partner in OIF.

Replace `<IDP_OR_SP>` by IDP or SP, the type of partner.

```
updatePartnerProperty("<PARTNER_NAME>", "<IDP_OR_SP>",
    "signingkeystoreaccesstemplateid", "saml-signing", "string")
```

5. Update the partner configuration to set the Encryption Key property (referenced by `encryptionkeystoreaccesstemplateid`) to the key entry ID defined in the **Federation Settings, Keystore** section.
In this example, `saml-encryption` is the key entry ID.

Replace `<PARTNER_NAME>` by the name of the partner in OIF.

Replace `<IDP_OR_SP>` by IDP or SP, the type of partner.

```
updatePartnerProperty("<PARTNER_NAME>", "<IDP_OR_SP>",
    "encryptionkeystoreaccesstemplateid", "saml-encryption", "string")
```

6. Exit the WLST environment: `exit()`

After the partner configuration has been updated, the partner must generate SAML 2.0 metadata or certificate information.

To generate SAML 2.0 metadata for new signing and encryption key perform the following steps.

1. Open a new browser and use the following URL to generate metadata.
`http://oam-runtime-host:oam-runtime-port/oamfed/idp/metadata?signid=<SIGN_KEYENTRY_ID>&encid=<ENC_KEYENTRY_ID>`
 The signid query parameter contains the key entry ID for the signing certificate. Replace `<SIGN_KEYENTRY_ID>`.
 The encid query parameter contains the key entry ID for the encryption certificate. Replace `<SIGN_KEYENTRY_ID>`.
 An example would be: `http://oam.com/oamfed/idp/metadata?signid=saml-signing&encid=samlencryption`

To generate certificate file for the new key perform the following steps.

1. Open a new browser and use the following URL to generate certificate in PEM format.
`http://oam-runtime-host:oam-runtime-port/oamfed/idp/cert?id=<KEYENTRY_ID>`
 The id query parameter contains the key entry ID for the certificate. Replace `<KEYENTRY_ID>`.
 An example would be: `http://oam.com/oamfed/idp/cert?id=saml-signing`



Note:

You can first generate the SAML 2.0 Metadata/Certificate and provide it to the partner before updating the partner configuration.

32.3.7 Managing OSTs Key Rollover

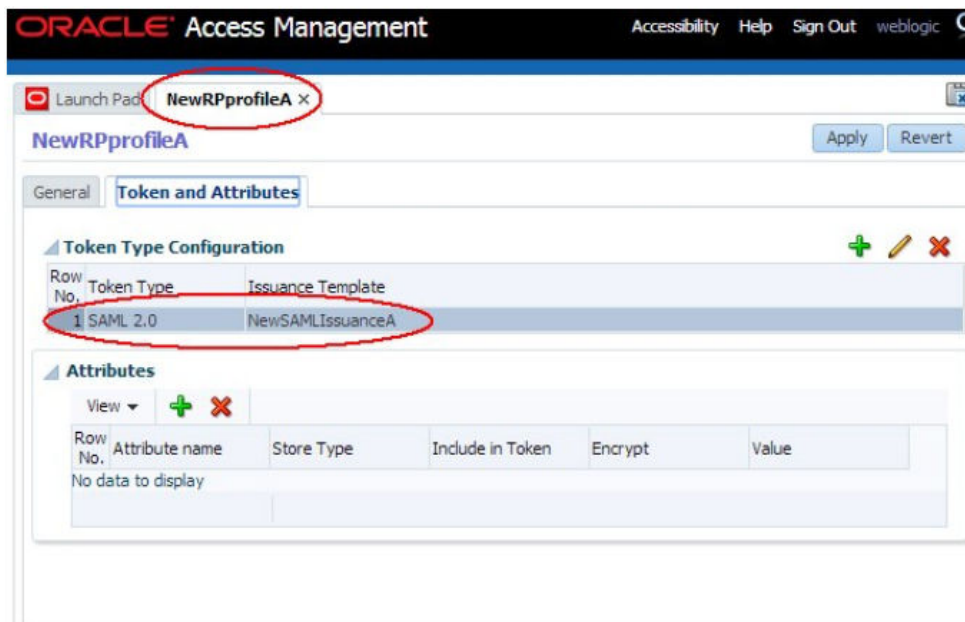
To explain OSTs key rollover consider the following example:

- Three Relying Party Partners: RP1, RP2, and RP3
- Two Relying Party Profiles: RPprofileA and RPprofileB, with RP1 and RP2 using RPprofileA and RP3 using RPprofileB
- Two SAML 2.0 Issuance Templates: SAMLIssuanceA referenced by RPprofileA and SAMLIssuanceB referenced by RPprofileB

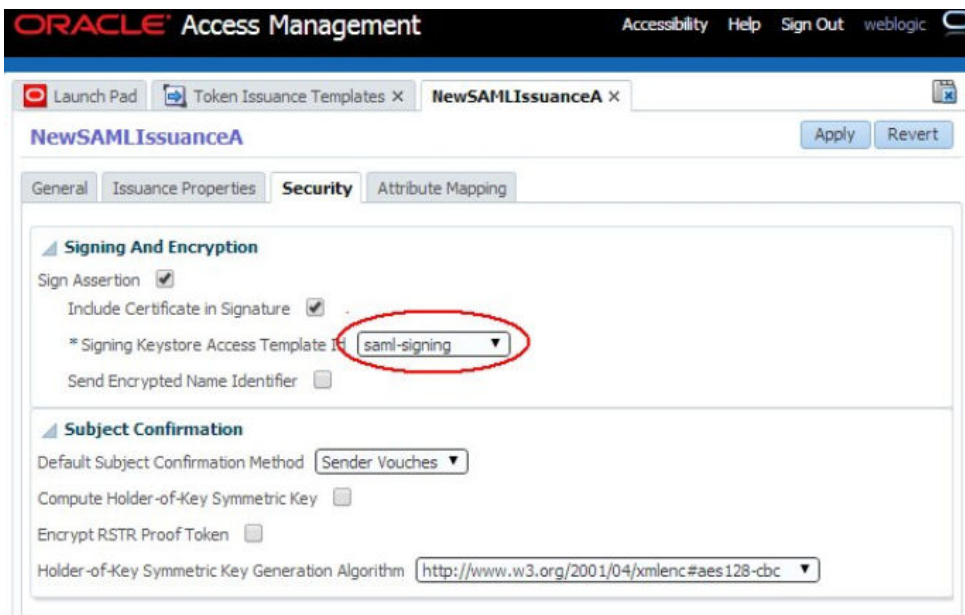
The rollover consists of switching the RP1 first then followed by RP2 and RP3.

To switch RP1 perform the following steps.

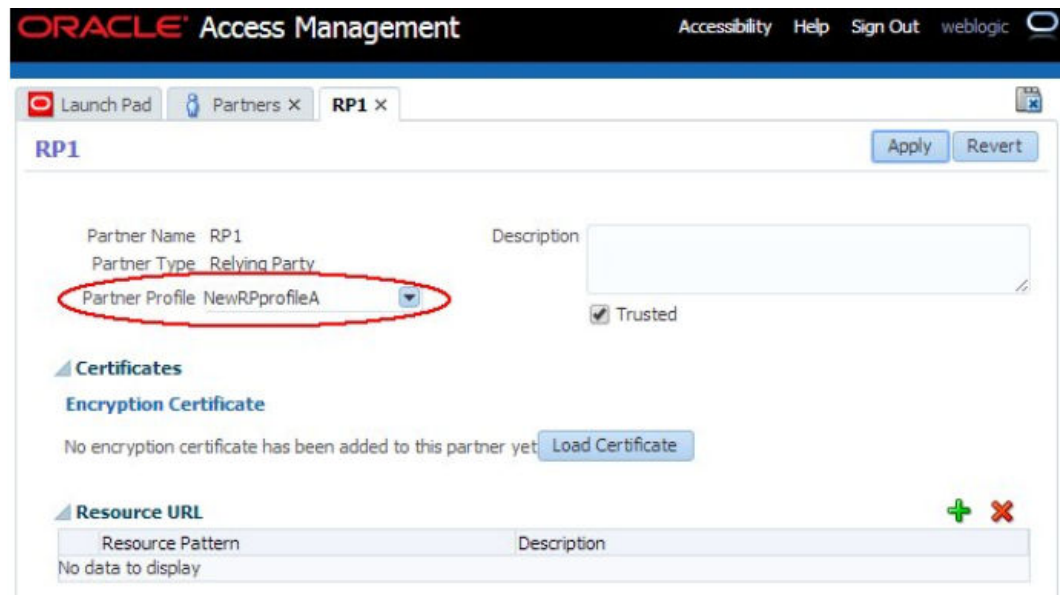
1. Login to the OAM Administration Console: `https://oam-admin-host:oam-adminport/oamconsole`.
2. Navigate to **Security Token Service, Partner Profiles, Relying Party Profiles**.
3. Create a new Relying Party Profile called `NewRPprofileA`, a copy of `RPprofileA`.
4. Navigate to **Security Token Service, Token Issuance Templates**.
5. Create a new SAML Issuance Template called `NewSAMLIssuanceA`, a copy of `SAMLIssuanceA`.
6. Update `NewRPprofileA` to reference `NewSAMLIssuanceA` SAML 2.0 Issuance Template.



7. Update NewSAMLIssuanceA SAML 2.0 Issuance Template in the Security tab to use the new key entry.



8. Navigate to Security **Token Service, Partners, Relying Parties**.
As a result, OSTs uses the new key entry `saml-signing` for signing outgoing SAML 2.0 Assertions for the RP1 Relying Party partner.



- Download the new certificates from OSTs by opening a browser and use the following URL to generate the certificate in PEM format:

```
http://oam-runtime-host:oam-runtime-port/sts/servlet/samlcert?
id=<KEYENTRY_ID>
```

The id query parameter contains the key entry ID for the certificate. Replace <KEYENTRY_ID>.

An example would be: <http://oam.com/sts/servlet/samlcert?id=saml-signing>

- Provide the certificate to the partner.

The process of switching RP2 to a new certificate will be easier since the new Relying Party profile and SAML Issuance Template already exist.

To switch RP2 perform the following steps.

- Login to the OAM Administration Console: <https://oam-admin-host:oam-adminport/oamconsole>.
- Navigate to **Security Token Service, Partners, Relying Parties**.
- Open RP1 and configure it to use the `NewRPprofileA` Relying Party Profile. As a result, OSTs uses the new key entry `saml-signing` for signing outgoing SAML 2.0 Assertions for the RP1 Relying Party partner.
- Provide the certificate to the partner.

Since the new Relying Party profile and SAML Issuance Template for RP3 have not been created yet, switching RP3 to the new certificate requires repeating the previously executed operations for RP1.

 **Note:**

You can first provide the new certificate to the partner before updating the OSTs configuration.

32.4 Cryptographic Settings in Oracle Identity Federation

This chapter describes the crypto configuration properties in OIF that are used to affect the Federation SSO exchanges.

32.4.1 Hashing Algorithms

Oracle Identity Federation (OIF) supports the consumption and issuance of SAML messages signed with the SHA-1 hashing algorithm or SHA-256 hashing algorithm.

By default, OIF uses SHA-1 for signing outgoing messages. Messages are signed differently based on the binding being used.

- For XML Digital signatures HTTP-POST or Artifact bindings are used.
- For Query signatures HTTP-Redirect binding is used.

32.4.2 Examples on SHA-1 Signed Messages

An example of a signed AuthnRequest message sent via the HTTP-Redirect binding would be:

```
https://acme.com/idp/saml20?
SAMLRequest=hVPLbtswEPwVgT1LpB6tY8JyoNZ1q9ZujVgJ3N4Yio5ZSKTMpaz470v5ESQ
B4gA8LWZ2ZnaXo%2BvHuvJ2woDUKkVhQJAnFNe1VA8pui2m%2FhXywdJVsorkaK9AHQ9Hg
Grq4Zmrd2oG7FtBVjPNVJAS5COuLG2oRh3XRd0caDNA44IIZgMsUP1ka%
%2B%2FruIoOPN0IOU8aba1MxeDtpXXKj1AeoOxUq7R%2BMX0py%2Fko5iQiKsWd3rj%2FAz
yfn%2FnJd881CV5Lv37URBuFrGzWTVVaWRgAgL6sq3X0xgln3NaxpBcLjo%2BrLzzH%2BD
w%3D%3D&RelayState=id-AkgTE5PMRAZTaKRcZHT-2rIse-
oPhCxyI00Xycbf&SigAlg=hSp%3A%2F%2Fwww.w3.org%2F2000%2F09%2Fxmldsig%23rs
a-
shal&Signature=rjZFsFuaFKv77JbspdDwT2wGV366iL3zvWc%2B1aybu%2FW%2BpFwLof
TJbtVsKfwJre1nGCU5SEvFD%2FBBURkxOG1KhR3k%2FrOw%2Bj7g7RlHfSahKaA03p6aAGQ
YPCpz%2Fd0%2BKArDAL%2FDNoH46G6Pnf7VWSb6a2COUiTV6118KaPbexrnJtE
```

An example of a SAML 2.0 Assertion sent via the HTTP-POST binding would be:

```
<saml:Response ...<saml:Response ...>
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">...
</saml:Issuer>
<saml>Status><saml:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/></saml:Status>
<saml:Assertion ID="id-BgLUimKUWYyS3JQbf2geeP9EwS-eGKxOPTuPvxgJ" ...>
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">...
</saml:Issuer>
<dsig:Signature>
<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
excc14n#" />
<dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#rsasha1" />
```

```

<dsig:Reference URI="#id-BgLUimKUWYyS3JQbf2geeP9EwS-eGKxOPTuPvxgJ">
<dsig:Transforms>
<dsig:Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#envelopedsignature"/>
<dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>uS85cIFf4z9KcHH/z60fNRPLoyo=</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>NiTyPtOEjyG...SpVjhbKxY=</dsig:SignatureValue>
</dsig:Signature>
<saml:Subject>
...
</saml:Subject>
<saml:Conditions ...>
...
</saml:Conditions>
<saml:AuthnStatement ...>
...
</saml:AuthnStatement>
</saml:Assertion>
</samlp:Response>

```

32.4.3 Examples on SHA-256 Signed Messages

An example of a signed AuthnRequest message sent via the HTTP-Redirect binding would be:

```

https://acme.com/idp/saml20?
SAMLRequest=hVNdb9owFP0rkfec%2BCbAVixClY2hIdGWDui6vhnbfEuOndoOKfv1c%2Fio2kqlkp%
2BuzrnnnHuvB5fPpYq2wjppdI7SBFAkNDNc6sccLRfj%2BAJFz1PNqTJa5GgnHLocDhwtVUWK2m%2F0
b%2FFUC%2Bej0Eg7wp0MxI33FcG4aZqk6STGPuIMADD0c%2F%2FFLDswb9bFN2dNp61G584V4vJ3o4P
JUi7MYTXWaRd0vtKoP%2BAolHYsdTU71nHbJQzgEo8JbULAS1TImGmJN%2B6%2FT5eC441r3A7%2F20
G6HAzZC9lo7GxJfXng7aVEGq9h4ZD8dLv0PCNNGPvpLMOQIYNLv9AX4lebrZ69B1MpoZJdnuUxtprk
63UVKpCs6tcA5FhVKmRelayState=id-
BiQreMi9cMY3oFI9PKMNktuOjpuFS2PrW4R8KKvd&SigAlg=hSp%3A%2F%2Fwww.w3.org%2F2001%2
F04%2Fxmldsig-more%23rsa-
sha256&Signature=PvyMUD%2FKXnCc0dr1n1pvoK171znJkajEHLgtzE4I7YFQIvP4wp3M%2FV7y08
x0Qkv0jwo9K4VBG%2BQUBftXr41ZDp%2BHob7G1maY973n7X2UD1bUbV1rJX%2FqS1GyyNY6MSMc05K
0J7VJcQXf8CvGEcVHr%2FZhpjihnaO2vi%2Bej3fbfgo%3D

```

An example of a SAML 2.0 Assertion sent via the HTTP-POST binding would be:

```

<samlp:Response ...<samlp:Response ...>
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">...
</saml:Issuer>
<samlp:Status><samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/></samlp:Status>
<saml:Assertion ID="id-5B4KZ-PeUzikxtC-Cr9g6uFQ-muwj3ZmC4PUW4wT" ...>
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">...
</saml:Issuer>
<dsig:Signature>

```

```

<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
excc14n#" />
<dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsasha256" />
<dsig:Reference URI="#id-5B4KZ-PeUzikxtC-Cr9g6uFQ-muwj3ZmC4PUW4wT">
<dsig:Transforms>
<dsig:Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#envelopedsignature" />
<dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</dsig:Transforms>
<dsig:DigestMethod Algorithm="http://www.w3.org/2001/04/
xmlenc#sha256" />
<dsig:DigestValue>Ppx/...L9ooHtsvgxvI=</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>G6yppQXy...SzHz2oa+zA=</dsig:SignatureValue>
</dsig:Signature>
<saml:Subject>
...
</saml:Subject>
<saml:Conditions ...>
...
</saml:Conditions>
<saml:AuthnStatement ...>
...
</saml:AuthnStatement>
</saml:Assertion>
</samlp:Response>

```

32.4.4 Configuring OIF to use SHA-1 or SHA-256 Hashing Algorithm

Oracle Identity Federation (OIF) can be configured at the following levels to use SHA-1 or SHA-256 in SAML signatures.

- At a partner level
- At a partner profile level, where all partners referencing this profile will be affected unless they were configured at a partner level for SHA-1/SHA-256 signatures

Use the following WLST commands to configure how OIF should compute a signature.

1. Enter the WLST environment.

```
$IAM_ORACLE_HOME/common/bin/wlst.sh
```

2. Connect to the WLS Admin server.

```
connect ()
```

3. Navigate to the Domain Runtime branch.

```
domainRuntime ()
```

4. Run the `configureFedDigitalSignature()` command.

```
configureFedDigitalSignature(partner="", partnerProfile="",
partnerType="", default="false", algorithm="SHA-256",
displayOnly="false", delete="false")
```

You can set the following parameters:

- `partner`: To configure a specific partner
- `partnerProfile`: To configure a specific partner profile
- `partnerType`: Indicates the type of partner/partner profile (idp or sp)
- `algorithm`: Indicates which hashing algorithm to use (SHA-1 or SHA-256)
- `displayOnly`: Indicates whether or not the command should display the setting on this partner/partner profile instead of setting it. If set to true, this command will not modify the configuration (true or false)
- `delete`: Indicates whether or not the command should delete the setting on this partner/partner profile instead of setting it. If set to true, this command will delete the configuration and the parent configuration (partner profile or global) will be used (true or false)

An example would be:

```
configureFedDigitalSignature(partner="AcmeIdP", partnerType="idp",
algorithm="SHA-256")
```

5. Exit WLST environment.

```
exit()
```

32.4.5 Signing Outgoing Messages

This section provides information on how to configure the following settings:

- Out of the Box (OOTB) Boolean settings for the outgoing SAML messages
- SAML 2.0 AuthnRequest at different levels
- Properties defined at SP/IP partner profiles

32.4.5.1 OOTB Configurations for Outgoing SAML Messages

Following are the Out-of-the-box (OOTB) Boolean settings that indicate when OIF need to sign outgoing SAML messages (if set to true, OIF signs the outgoing message).

Global Level

- `saml20sendsignedauthnrequest`: SAML 2.0 AuthnRequest (true)

SAML 1.1 IdP Partner Profile

- `sendsignedrequestsoap`: SAML 1.1 Request via the Artifact/SOAP binding (true)

SAML 1.1 SP Partner Profile

- `sendsignedassertion`: SAML 1.1 Assertion (true)

- `sendsignedresponseassertionpost`: SAML 1.1 Response containing an Assertion over the HTTP-POST binding (false)
- `sendsignedresponseassertionsoap`: SAML 1.1 Response containing an Assertion over the Artifact/SOAP binding (false)
- `sendsignedresponsesoap`: SAML 1.1 Response not containing an Assertion over the Artifact/SOAP binding (true)

SAML 2.0 IdP Partner Profile

- `sendsignedrequestpost`: SAML 2.0 Request over the HTTP-POST binding (true)
- `sendsignedrequestquery`: SAML 2.0 Request over the HTTP-Redirect binding (true)
- `sendsignedrequestsoap`: SAML 2.0 Request over the Artifact/SOAP binding (true)
- `sendsignedresponsepost`: SAML 2.0 Response not containing an Assertion over the HTTP-POST binding (true)
- `sendsignedresponsequery`: SAML 2.0 Response not containing an Assertion over the HTTP-Redirect binding (true)
- `sendsignedresponsesoap`: SAML 2.0 Response not containing an Assertion over the Artifact/SOAP binding (true)

SAML 2.0 SP Partner Profile

- `sendsignedassertion`: SAML 2.0 Assertion (true)
- `sendsignedrequestpost`: SAML 2.0 Request over the HTTP-POST binding (true)
- `sendsignedrequestquery`: SAML 2.0 Request over the HTTP-Redirect binding (true)
- `sendsignedrequestsoap`: SAML 2.0 Request over the Artifact/SOAP binding (true)
- `sendsignedresponseassertionpost`: SAML 2.0 Response containing an Assertion over the HTTP-POST binding (false)
- `sendsignedresponseassertionsoap`: SAML 2.0 Response containing an Assertion over the Artifact/SOAP binding (false)
- `sendsignedresponsepost`: SAML 2.0 Response not containing an Assertion over the HTTP-POST binding (true)
- `sendsignedresponsequery`: SAML 2.0 Response not containing an Assertion over the HTTP-Redirect binding (true)
- `sendsignedresponsesoap`: SAML 2.0 Response not containing an Assertion over the Artifact/SOAP binding (true)

32.4.5.2 Configuring SAML 2.0 AuthnRequest

You can configure OIF to sign an outgoing SAML 2.0 AuthnRequest at the following levels:

- Global level
- IdP Partner Profile level
- IdP Partner level
- Partner Profile level

- Partner level

Perform the following steps to access Domain Runtime branch:

1. Enter the WLST environment:

```
$IAM_ORACLE_HOME/common/bin/wlst.sh
```

2. Connect to the WLS Admin server:

```
connect ()
```

3. Navigate to the Domain Runtime branch:

```
domainRuntime ()
```

Upon accessing the Domain Runtime branch, run any of the following commands to configure OIF to sign an outgoing SAML 2.0 AuthnRequest at the appropriate level.

- **To configure at a global level:**

```
putBooleanProperty("/spglobal/saml20sendsignedauthnrequest", "true/false")
```

Set the value to true to have OIF sign the outgoing AuthnRequest.

An example would be:

```
putBooleanProperty("/spglobal/saml20sendsignedauthnrequest", "true")
```

- **To configure SAML 2.0 IdP at a Partner Profile level:**

```
putBooleanProperty("/fedpartnerprofiles/PARTNER_PROFILE/  
sendsignedauthnrequest", "true/false")
```

Replace `PARTNER_PROFILE` by a SAML 2.0 IdP Partner Profile name.

Set the value to true to have OIF sign the outgoing AuthnRequest.

An example would be:

```
putBooleanProperty("/fedpartnerprofiles/saml20-idp-partner-profile/  
sendsignedauthnrequest", "true")
```

- **To configure SAML 2.0 at a IdP Partner level:**

```
updatePartnerProperty("PARTNER", "idp", "sendsignedauthnrequest", "true/  
false", "boolean")
```

Replace `PARTNER` by a SAML 2.0 IdP Partner name.

Set the value to true to have OIF sign the outgoing AuthnRequest.

An example would be:

```
updatePartnerProperty("AcmeIdP", "idp", "sendsignedauthnrequest",  
"false", "boolean")
```

- **To configure at a Partner Profile level:**

```
putBooleanProperty("/fedpartnerprofiles/PARTNER_PROFILE/  
PROPERTY_NAME", "true/false")
```

Replace `PARTNER_PROFILE` by a Partner Profile name.

Replace `PROPERTY_NAME` by the name of the property to set the value to true or false.

An example would be:

```
putBooleanProperty("/fedpartnerprofiles/saml20-idp-partner-profile/  
sendsignedrequestquery", "true")
```

- **To configure at a Partner level:**

```
updatePartnerProperty("PARTNER", "PARTNER_TYPE",  
"PROPERTY_NAME", "true/false", "boolean")
```

Replace `PARTNER` by a Partner name.

Replace `PARTNER_TYPE` by the type of the specified Partner (IdP or SP).

Replace `PROPERTY_NAME` by the name of the property to set the value to true or false.

An example would be:

```
updatePartnerProperty("AcmeSP", "sp", "sendsignedrequestquery",  
"true", "boolean")
```

32.4.5.3 Changing SAML 2.0 Metadata

Changing the `saml20sendsignedauthnrequest` property at a global level changes the following attribute in the SAML 2.0 Metadata generated by OIF.

- The `AuthnRequestsSigned` attribute in the `SPSSODescriptor` element is set based on `saml20sendsignedauthnrequest` property.

A sample SAML 2.0 Metadata shows these two attributes:

```
<md:EntityDescriptor ...>  
<dsig:Signature>  
...  
</dsig:Signature>  
<md:IDPSSODescriptor WantAuthnRequestsSigned="false" ...>  
...  
</md:IDPSSODescriptor>  
<md:SPSSODescriptor AuthnRequestsSigned="true"
```



```
WantAssertionsSigned="true"  
...>  
</md:SPSSODescriptor>  
</md:EntityDescriptor>
```

32.4.6 Signing Incoming Messages

This section provides information on how to configure the following settings.

- OOTB Boolean settings for the incoming SAML messages
- SAML 2.0 AuthnRequest at different levels
- SAML 1.1 Assertion at different levels
- SAML 2.0 Assertion at different levels
- Properties defined at SP/IP partner profiles

32.4.6.1 OOTB Boolean Settings for Incoming SAML Messages

Following are the Out-of-the-box (OOTB) Boolean settings that indicate when OIF need to require incoming SAML messages (if set to true, OIF requires the incoming message).

Global Level

- `saml20requiresignedauthnrequest`: SAML 2.0 AuthnRequest (false)
- `saml11requiresignedassertion`: SAML 1.1 Assertion contained in a Response message (true)
- `saml20requiresignedassertion`: SAML 2.0 Assertion contained in a Response message (true)

SAML 1.1 IdP Partner Profile

- `requiresignedresponseassertionpost`: SAML 1.1 Response via the HTTP-POST binding (false)
- `requiresignedresponseassertionsoap`: SAML 1.1 Response via the Artifact/SOAP binding (false)

SAML 1.1 SP Partner Profile

- `requiresignedrequestsoap`: SAML 1.1 Request via the Artifact/SOAP binding (false)

SAML 2.0 IdP Partner Profile

- `requiresignedrequestpost`: SAML 2.0 Request over the HTTP-POST binding (false)
- `requiresignedrequestquery`: SAML 2.0 Request over the HTTP-Redirect binding (false)
- `requiresignedrequestsoap`: SAML 2.0 Request over the Artifact/SOAP binding (false)
- `requiresignedresponseassertionpost`: SAML 2.0 Response containing an Assertion over the HTTP-POST binding (false)
- `requiresignedresponseassertionsoap`: SAML 2.0 Response containing an Assertion over the Artifact/SOAP binding (false)
- `requiresignedresponsepost`: SAML 2.0 Response not containing an Assertion over the HTTP-POST binding (false)

- `requiresignedresponsequery`: SAML 2.0 Response not containing an Assertion over the HTTP-Redirect binding (false)
- `requiresignedresponsesoap`: SAML 2.0 Response not containing an Assertion over the Artifact/SOAP binding (false)

SAML 2.0 SP Partner Profile

- `requiresignedrequestpost`: SAML 2.0 Request over the HTTP-POST binding (false)
- `requiresignedrequestquery`: SAML 2.0 Request over the HTTP-Redirect binding (false)
- `requiresignedrequestsoap`: SAML 2.0 Request over the Artifact/SOAP binding (false)
- `requiresignedresponsepost`: SAML 2.0 Response not containing an Assertion over the HTTP-POST binding (false)
- `requiresignedresponsequery`: SAML 2.0 Response not containing an Assertion over the HTTP-Redirect binding (false)
- `requiresignedresponsesoap`: SAML 2.0 Response not containing an Assertion over the Artifact/SOAP binding (false)



Note:

If an incoming message is signed, even though OIF does not require this type of message to be signed, OIF verifies the message and returns an error if signature validation fails.

32.4.6.2 Configuring SAML 2.0 AuthnRequest

You can configure OIF to sign an outgoing SAML 2.0 AuthnRequest at the following levels:

- Global level
- IdP Partner Profile level
- IdP Partner level
- Partner Profile level
- Partner level

Perform the following steps to access Domain Runtime branch:

1. Enter the WLST environment:

```
$IAM_ORACLE_HOME/common/bin/wlst.sh
```

2. Connect to the WLS Admin server:

```
connect ()
```

3. Navigate to the Domain Runtime branch:

```
domainRuntime()
```

Upon accessing the Domain Runtime branch, run any of the following commands to configure OIF to sign an outgoing SAML 2.0 AuthnRequest at the appropriate level.

- **To configure at a global level:**

```
putBooleanProperty("/spglobal/saml20sendsignedauthnrequest", "true/false")
```

Set the value to true to have OIF sign the outgoing AuthnRequest.

An example would be:

```
putBooleanProperty("/spglobal/saml20sendsignedauthnrequest", "true")
```

- **To configure SAML 2.0 IdP at a Partner Profile level:**

```
putBooleanProperty("/fedpartnerprofiles/PARTNER_PROFILE/  
sendsignedauthnrequest", "true/false")
```

Replace `PARTNER_PROFILE` by a SAML 2.0 IdP Partner Profile name.

Set the value to true to have OIF sign the outgoing AuthnRequest.

An example would be:

```
putBooleanProperty("/fedpartnerprofiles/saml20-idp-partner-profile/  
sendsignedauthnrequest", "true")
```

- **To configure SAML 2.0 at a IdP Partner level:**

```
updatePartnerProperty("PARTNER", "idp", "sendsignedauthnrequest", "true/  
false", "boolean")
```

Replace `PARTNER` by a SAML 2.0 IdP Partner name.

Set the value to true to have OIF sign the outgoing AuthnRequest.

An example would be:

```
updatePartnerProperty("AcmeIdP", "idp", "sendsignedauthnrequest",  
"false", "boolean")
```

- **To configure at a Partner Profile level:**

```
putBooleanProperty("/fedpartnerprofiles/PARTNER_PROFILE/PROPERTY_NAME",  
"true/false")
```

Replace `PARTNER_PROFILE` by a Partner Profile name.

Replace `PROPERTY_NAME` by the name of the property to set the value to true or false.

An example would be:

```
putBooleanProperty("/fedpartnerprofiles/saml20-idp-partner-profile/  
sendsignedrequestquery", "true")
```

- **To configure at a Partner level:**

```
updatePartnerProperty("PARTNER", "PARTNER_TYPE",  
"PROPERTY_NAME","true/false", "boolean")
```

Replace `PARTNER` by a Partner name.

Replace `PARTNER_TYPE` by the type of the specified Partner (IdP or SP).

Replace `PROPERTY_NAME` by the name of the property to set the value to true or false.

An example would be:

```
updatePartnerProperty("AcmeSP", "sp", "sendsignedrequestquery",  
"true","boolean")
```

32.4.6.3 Configuring SAML 1.1 Assertion for Incoming Messages

Upon accessing the Domain Runtime branch, run any of the following commands to configure OIF to sign or not sign an incoming SAML 1.1 Assertions at the appropriate level.

- **To configure at a global level:**

```
putBooleanProperty("/spglobal/saml11requiresignedassertion", "true/  
false")
```

Set the value to true to have OIF require incoming SAML 1.1 Assertions to be signed.

An example would be:

```
putBooleanProperty("/spglobal/saml11requiresignedassertion", "true")
```

- **To configure at a SAML 1.1 IdP Partner Profile level:**

```
putBooleanProperty("/fedpartnerprofiles/PARTNER_PROFILE/  
requiresignedassertion", "true/false")
```

Replace `PARTNER_PROFILE` by a SAML 1.1 IdP Partner Profile name.

Set the value to true to have OIF require incoming SAML 1.1 Assertions to be signed.

An example would be:

```
putBooleanProperty("/fedpartnerprofiles/saml11-idp-partner-profile/  
requiresignedassertion", "true")
```

- **To configure at a SAML 1.1 IdP Partner level:**

```
updatePartnerProperty("PARTNER", "idp", "requiresignedassertion", "true/  
false", "boolean")
```

Replace `PARTNER` by a SAML 1.1 IdP Partner name.

Set the value to true to have OIF require incoming SAML 1.1 Assertions to be signed.

An example would be:

```
updatePartnerProperty("AcmeSP", "sp", "requiresignedassertion", "false",  
"boolean")
```

32.4.6.4 Configuring SAML 2.0 Assertion for Incoming Messages

Upon accessing the Domain Runtime branch, run any of the following commands to configure OIF to sign or not sign an incoming SAML 2.0 Assertions at the appropriate level.

- **To configure at a global level:**

```
putBooleanProperty("/spglobal/saml20requiresignedassertion", "true/false")
```

Set the value to true to have OIF require incoming SAML 2.0 Assertions to be signed.

An example would be:

```
putBooleanProperty("/spglobal/saml20requiresignedassertion", "true")
```

- **To configure at a SAML 2.0 IdP Partner Profile level:**

```
putBooleanProperty("/fedpartnerprofiles/PARTNER_PROFILE/  
requiresignedassertion", "true/false")
```

Replace `PARTNER_PROFILE` by a SAML 2.0 IdP Partner Profile name.

Set the value to true to have OIF require incoming SAML 2.0 Assertions to be signed.

An example would be:

```
putBooleanProperty("/fedpartnerprofiles/saml20-idp-partner-profile/  
requiresignedassertion", "true")
```

- **To configure at a SAML 2.0 IdP Partner level:**

```
updatePartnerProperty("PARTNER", "idp", "requiresignedassertion", "true/  
false", "boolean")
```

Replace `PARTNER` by a SAML 2.0 IdP Partner name.

Set the value to true to have OIF require incoming SAML 2.0 Assertions to be signed.

An example would be:

```
updatePartnerProperty("AcmeSP", "sp", "requiresignedassertion",
"false", "boolean")
```

32.4.6.5 Changing SAML 2.0 Metadata of Incoming Messages

Changing the `saml20requiresignedauthnrequest` or `saml20requiresignedassertion` properties at a global level changes the following attributes in the SAML 2.0 Metadata generated by OIF.

- The `WantAuthnRequestsSigned` attribute in the `IDPSSODescriptor` element is set based on `saml20requiresignedauthnrequest` property.
- The `WantAssertionsSigned` attribute in the `SPSSODescriptor` element is set based on `saml20requiresignedassertion` property.

A sample SAML 2.0 Metadata shows these two attributes:

```
<md:EntityDescriptor ...>
<dsig:Signature>
...
</dsig:Signature>
<md:IDPSSODescriptor WantAuthnRequestsSigned="false" ...>
...
</md:IDPSSODescriptor>
<md:SPSSODescriptor AuthnRequestsSigned="true"
WantAssertionsSigned="true"
...>
</md:SPSSODescriptor>
</md:EntityDescriptor>
```

32.4.7 Configuring X.509 Certificate in Outgoing Message

The OIF can be configured to send the X.509 signing certificate in an outgoing XML SAML message sent via the HTTP-POST or SOAP binding.

The `includecertinsignature` Boolean property indicates whether or not the certificate will be added to the message.

For OIF to send the X.509 signing certificate in an outgoing message, run one of the following commands to set the `includecertinsignature` Boolean property.

- **To configure at a Partner Profile level:**

```
putBooleanProperty("/fedpartnerprofiles/PARTNER_PROFILE/
includecertinsignature", "true/false")
```

Replace `PARTNER_PROFILE` by a Partner Profile name.

Set the value to true or false.

An example would be:

```
putBooleanProperty("/fedpartnerprofiles/saml20-sp-partner-profile/  
includecertinsignature", "true")
```

- **To configure at a Partner level:**

```
updatePartnerProperty("PARTNER", "PARTNER_TYPE",  
"includecertinsignature", "true/false", "boolean")
```

Replace `PARTNER` by a Partner name.

Replace `PARTNER_TYPE` by the type of the specified Partner (idp or sp).

Set the value to true or false.

An example would be:

```
updatePartnerProperty("AcmeSP", "sp", "includecertinsignature", "true",  
"boolean")
```

32.4.8 Managing SAML 2.0 Encryption

With SAML 2.0, you can encrypt the following data in messages:

- Assertions
- NameIDs
- Attributes

OIF allows an administrator to specify which types of data should be encrypted.

32.4.8.1 OOTB Configuration to Encrypt Outgoing SAML Messages

The following OOTB Boolean values indicate when OIF should encrypt outgoing SAML messages (if set to true, OIF will encrypt the data):

- **SAML 2.0 IdP Partner Profile**
- `sendencryptednameid`: Indicates if NameID contained in LogoutRequest messages should be encrypted (false)
- **SAML 2.0 SP Partner Profile**
- `sendencryptedattribute`: Indicates if each attribute contained in a SAML Assertion should be encrypted (false)
- `sendencryptednameid`: Indicates if NameID contained in LogoutRequest, Assertion messages should be encrypted (false)

When creating a new SP Partner, the configuration for that Partner specifies that the OIF/IdP should not encrypt the Assertion:

- `sendencryptedassertion` on the partner entry: Indicates if the Assertion should be encrypted (false)

32.4.8.2 Encrypting Outgoing Assertion

Perform the following steps to configure OIF/IdP to encrypt the outgoing Assertion for an SP Partner via the OAM Administration Console:

1. Login to the OAM Administration Console: `https://oam-admin-host:oam-adminport/oamconsole`.
2. Navigate to **Identity Federation, Identity Provider Administration**.
3. Open **SP Partner**.
4. In the Advanced section, select the `Encrypt Assertion` checkbox.
5. Click **Save**.

To configure the SP Partner via WLST, perform the following steps:

1. Enter the WLST environment:

```
$IAM_ORACLE_HOME/common/bin/wlst.sh
```

2. Connect to the WLS Admin server:

```
connect()
```

3. Navigate to the Domain Runtime branch:

```
domainRuntime()
```

4. Execute the `updatePartnerProperty()` command:

```
updatePartnerProperty("PARTNER", "sp", "sendencryptedassertion",  
"true/false", "boolean")
```

Replace `PARTNER` by a Partner name.

Set the value to true or false.

An example would be:

```
updatePartnerProperty("AcmeSP", "sp", "sendencryptedassertion",  
"true", "boolean")
```

5. Exit the WLST environment:

```
exit()
```

32.4.8.3 Configuring NameID and Attributes Properties

To configure the properties, perform the following steps:

1. Enter the WLST environment:

```
$IAM_ORACLE_HOME/common/bin/wlst.sh
```


2. Connect to the WLS Admin server:

```
connect ()
```

3. Navigate to the Domain Runtime branch:

```
domainRuntime ()
```

4. To configure at a Partner Profile level:

```
putBooleanProperty("/fedpartnerprofiles/PARTNER_PROFILE/PROPERTY_NAME",  
"true/false")
```

Replace `PARTNER_PROFILE` by a Partner Profile name.

Replace `PROPERTY_NAME` by the name of the property to set.

Set the value to true or false.

An example would be:

```
putBooleanProperty("/fedpartnerprofiles/saml20-sp-partner-profile/  
sendencryptedaStribute", "true")
```

5. To configure at a Partner level:

```
updatePartnerProperty("PARTNER", "PARTNER_TYPE", "PROPERTY_NAME", "true/  
false", "boolean")
```

Replace `PARTNER` by a Partner name.

Replace `PARTNER_TYPE` by the type of the specified Partner (idp or sp).

Replace `PROPERTY_NAME` by the name of the property to set.

Set the value to true or false.

An example would be:

```
updatePartnerProperty("AcmeSP", "sp", "sendencryptedaStribute", "true",  
"boolean")
```

6. Exit the WLST environment:

```
exit ()
```

32.4.9 Encryption Algorithm

At the Partner or Partner Profile level, the encryption algorithm can be defined by setting the `defaultencryptionmethod` string property to one of the following values:

- <http://www.w3.org/2001/04/xmlenc#aes128-cbc> for AES-128 CBC
- <http://www.w3.org/2001/04/xmlenc#aes192-cbc> for AES-192 CBC
- <http://www.w3.org/2001/04/xmlenc#aes256-cbc> for AES-256 CBC

- <http://www.w3.org/2001/04/xmlenc#tripledes-cbc> for 3DES CBC

By default, that property is set to <http://www.w3.org/2001/04/xmlenc#aes128-cbc> (AES-128 CBC).

To configure the `defaultencryptionmethod` property, perform the following steps:

1. Enter the WLST environment by executing:

```
$IAM_ORACLE_HOME/common/bin/wlst.sh
```

2. Connect to the WLS Admin server:

```
connect()
```

3. Navigate to the Domain Runtime branch:

```
domainRuntime()
```

4. To configure at a Partner Profile level:

```
putStringProperty("/fedpartnerprofiles/PARTNER_PROFILE/  
defaultencryptionmethod", "ALGORITHM")
```

Replace `PARTNER_PROFILE` by a Partner Profile name.

Replace `ALGORITHM` by one of the above algorithm values.

An example would be:

```
putStringProperty("/fedpartnerprofiles/saml20-sp-partner-profile/  
defaultencryptionmethod", "http://www.w3.org/2001/04/  
xmlenc#tripledes-cbc")
```

5. To configure at a Partner level:

```
updatePartnerProperty("PARTNER", "PARTNER_TYPE",  
"defaultencryptionmethod", "ALGORITHM", "string")
```

Replace `PARTNER` by a Partner name.

Replace `PARTNER_TYPE` by the type of the specified Partner (idp or sp).

Replace `ALGORITHM` by one of the above algorithm values.

An example would be:

```
updatePartnerProperty("AcmeSP", "sp", "defaultencryptionmethod",  
"http://www.w3.org/2001/04/xmlenc#tripledes-cbc", "string")
```

6. Exit the WLST environment:

```
exit()
```

Part VIII

Managing the Adaptive Authentication Service and Oracle Mobile Authenticator

The Adaptive Authentication Service is a One Time Password Authenticator that provides *multifactor* authentication in addition to the standard user name and password type authentication.

This section contains the following chapters:

- [Introducing the Adaptive Authentication Service](#)
- [Configuring the Oracle Mobile Authenticator](#)
- [Configuring TOTP-based Multi Factor Authentication in OAM](#)

Introducing the Adaptive Authentication Service

The Adaptive Authentication Service offers stronger *multifactor* (also referred to as second factor) authentication for sensitive applications that require additional security in addition to the standard user name and password type authentication.

Multifactor authentication involves more than one stage when verifying the identity of an entity attempting to access services from a server or on a network. For example, when multifactor authentication is enabled and configured, the traditional user name and password is the first factor. Additional security is enforced by adding a One Time Pin (OTP) step, or an Access Request (Push) Notification step as a second factor in the authentication process. The following topics describe how the Adaptive Authentication Service and Access Manager Second Factor Authentication:

- [About Adaptive Authentication Service](#)
- [Working with the Adaptive Authentication Service](#)
- [Understanding Adaptive Authentication Service and OMA Configurations](#)
- [Configuring an Adaptive Authentication Service](#)

33.1 About Adaptive Authentication Service

The Adaptive Authentication Service offers the ability to add multiple steps to the authentication process.

Additional security may be enforced by adding a OTP step, or an Access Request (Push) Notification step after initial user authentication. This may or may not involve the use of the Oracle Mobile Authenticator, a mobile device app that uses Time-based One Time Password and push notifications to authenticate users within the second factor authentication scheme.



Note:

Installing Oracle Adaptive Access Manager is not required since the Adaptive Authentication Service uses a set of libraries that makes a OTP step feasible using the Oracle Mobile Authenticator.

The Adaptive Authentication Service has to be licensed and explicitly enabled in order to use it. Once the proper product license is procured you can enable the Adaptive Authentication Service using the Oracle Access Management Console. From the Oracle Access Management Console, the Adaptive Authentication Service can be enabled or disabled from the Available Services link on the Configuration Launch Pad.

 **See Also:**

- [Available Services of the Common Configuration Section](#)
- [Understanding the Oracle Access Management Console](#)
- [Working with the Adaptive Authentication Service](#)

33.2 Working with the Adaptive Authentication Service

The Adaptive Authentication Service offers second factor authentication. The second factor can be a One Time Pin (OTP) or an Access Request (or push) Notification. After an initial successful user/password authentication, a Second Factor Authentication page is displayed from which the user selects the preferred method of second factor authentication.

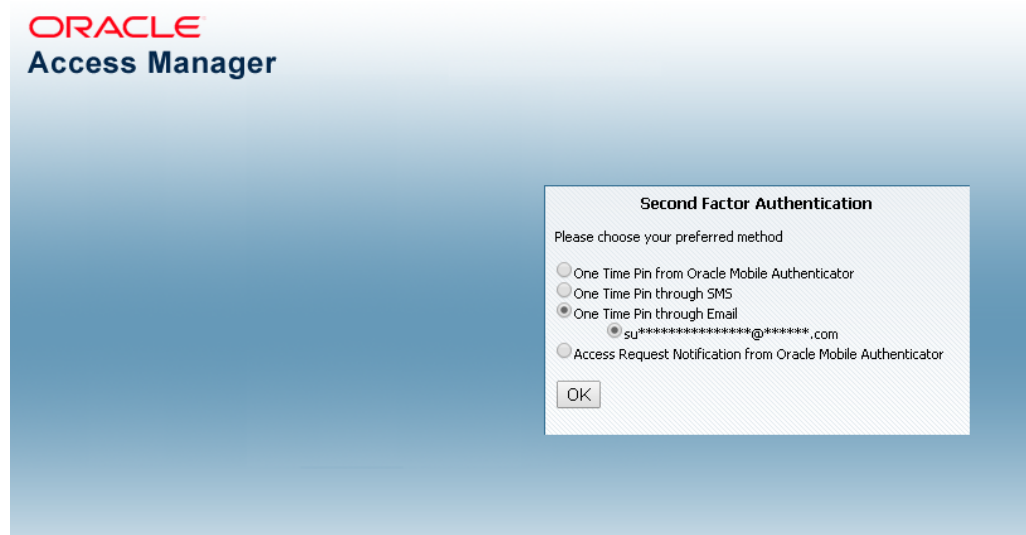
The following options are available:

- OTP from Oracle Mobile Authenticator
- OTP through SMS
- OTP through Email
- Access Request Notification from Oracle Mobile Authenticator

[Figure 33-1](#) shows the Second Factor Authentication page in which the user has selected the OTP Through Email option.

In this case, the user receives the OTP via a configured Email address.

Figure 33-1 Second Factor Authentication Preferred Method Page



If the selected option is either OTP From Oracle Mobile Authenticator or Access Request Notification from Oracle Mobile Authenticator, the Adaptive Authentication Service works in tandem with the Oracle Mobile Authenticator (OMA), a mobile device

app that uses Time-based One Time Password and push notifications to authenticate users within the second factor authentication scheme.

In advance of using the OTP from OMA or Access Request Notification from OMA options, a user must download a supported authenticator app to a mobile device (for example, Oracle Mobile Authenticator to an Apple iPhone) and configure it by clicking a link provided by the Access Manager administrator. (The OMA app is not needed if using the OTP through Email or OTP through SMS options.)



Note:

You must configure the Oracle Mobile Authenticator mobile device app to retrieve a secret key required to generate a OTP.

See [Generating a Secret Key for the Oracle Mobile Authenticator](#) for information about the secret key.

See [Understanding Oracle Mobile Authenticator Configuration](#) for information on how to configure the OMA.

The following topics describe each option and how the Oracle Mobile Authenticator works:

- [Understanding the One Time Password Option](#)
- [Understanding the Access Request \(Push\) Notification Option](#)
- [Using the Oracle Mobile Authenticator with OTP And Access Request](#)

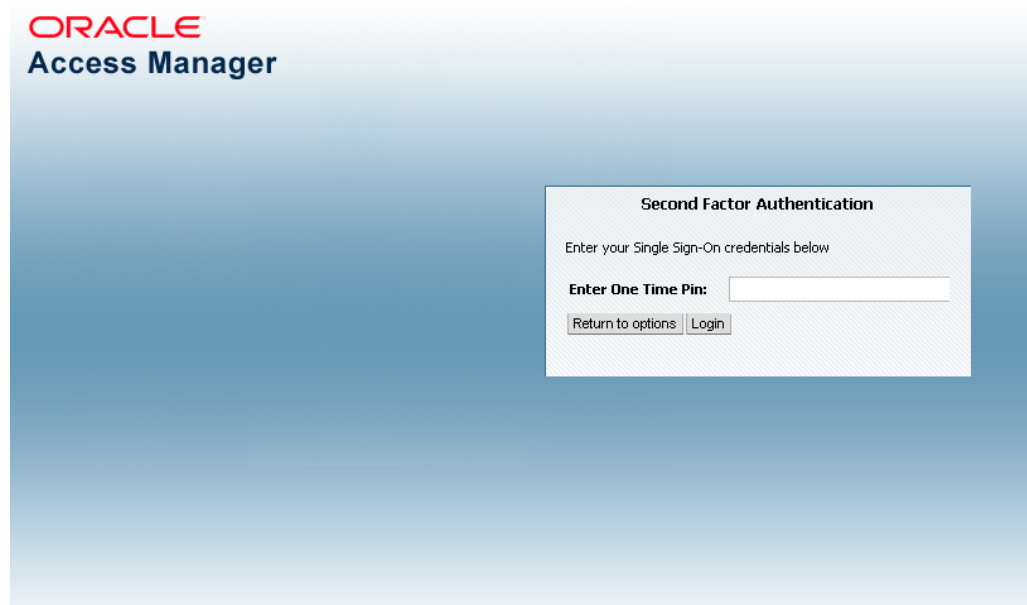
33.2.1 Understanding the One Time Password Option

After the successful authentication of initial credentials, the user needs to choose one of the OTP options as a second-factor authentication. Access to the protected resource is provided when the OTP received by the user is entered in the OTP login page.

Let's assume the Adaptive Authentication Service is enabled and configured for second factor authentication. When the user accesses a resource protected by Access Manager, a page is displayed that requests a user name and password. If these initial credentials are authenticated successfully, a Second Factor Authentication Preferred Method Page page is displayed and the user selects from one of the options. In this use case, the user selects one of the OTP options and receives a OTP through SMS/Email or generated and displayed by the OMA app. The user enters the OTP in the OTP login page.

[Figure 33-2](#) shows the OTP login page.

Figure 33-2 One Time Password Login Page



Once the OTP is successfully validated by Access Manager, the user is directed to the protected resource. On failure of any of the OTP options, an error message will be displayed, and the user will be returned to the same OTP page.

 **Note:**

Access Manager validates the OTP using the Time-based One Time Password (TOTP) algorithm. TOTP is a two-factor authentication scheme specified by the Internet Engineering Task Force (IETF) under RFC 6238 and used by the Adaptive Authentication Service. TOTP is an extension of the HMAC-based One Time Password algorithm and supports a time-based *moving factor* (a value that must be changed each time a new password is generated).

The following topics describe how the user may receive the OTP:

- [About using OTP through Email or SMS](#)
- [About using OTP from Oracle Mobile Authenticator](#)

33.2.1.1 About using OTP through Email or SMS

The user receives the OTP through an email or SMS and enters it in the OTP login page.

In cases where OTP through email or SMS is chosen, Access Manager will send a OTP to the configured email address or phone number respectively. The user then enters the received OTP and Access Manager will validate it. On a successful validation, the user will be directed to the protected resource.

The Adaptive Authentication Service expects that the required email address or phone number is configured in the appropriate field.

See [Configuring the Adaptive Authentication Plug-in in the Oracle Access Management Console](#).

When you use the OTP with Email or SMS option, the OTP is accessible from any device on which the email address can be accessed or from the SMS app that is associated with the specified phone number, respectively.



Note:

The OMA mobile app is not used for the OTP through Email or OTP through SMS options.

33.2.1.2 About using OTP from Oracle Mobile Authenticator

In the use case where a OTP will be generated and displayed by the OMA app on a mobile device, the app must be configured with the Access Manager server details.

Following this configuration, the user authenticates with Access Manager using the proper credentials and Access Manager will return a secret key. This secret key is unique to each user and known only to Access Manager and the OMA. The secret key is used to generate the OTP.

See [Generating a Secret Key for the Oracle Mobile Authenticator](#) on how to populate the secret key with the required data.

After Access Manager generates a OTP for the user using the secret key, the OTP is pushed to the OMA. The user then enters the OTP in the One Time Pin Login Page. If the OTP generated by Access Manager matches the OTP entered by the user, access to the protected resource is allowed. If the OTP entries do not match, access is not allowed.

See [Using the Oracle Mobile Authenticator with OTP And Access Request](#).



Note:

The OMA refreshes the OTP every 30 seconds so the OTP entered by a user is valid only for that period of time.

33.2.2 Understanding the Access Request (Push) Notification Option

The Access Manager sends an Access Request Notification to the notification server which is then pushed to the user's configured device.

Let's assume the Adaptive Authentication Service is enabled and configured for second factor authentication. When the user accesses a resource protected by Access Manager, a page is displayed that requests a user name and password. If these initial credentials are authenticated successfully, a Second Factor Authentication Preferred Method page is displayed and the user selects from one of the options. In this use case, the user selects Access Request Notification from Oracle Mobile Authenticator.

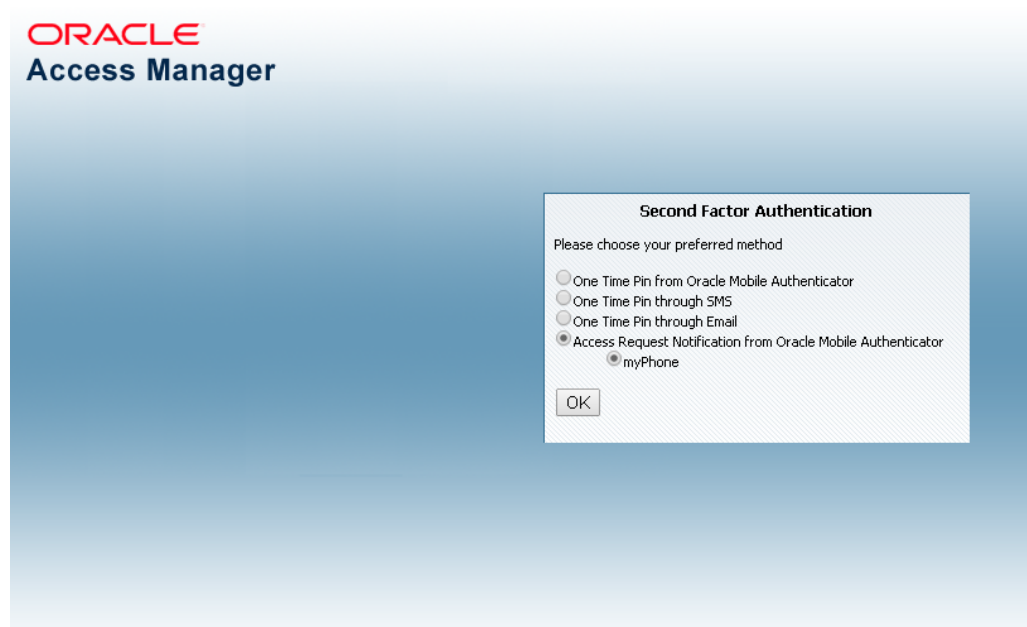


Note:

This is a push notification option which works in tandem with the OMA.
See [Using the Oracle Mobile Authenticator with OTP And Access Request](#).

Figure 33-3 shows the Second Factor Authentication Preferred Method Page with Access Request Notification that has been selected.

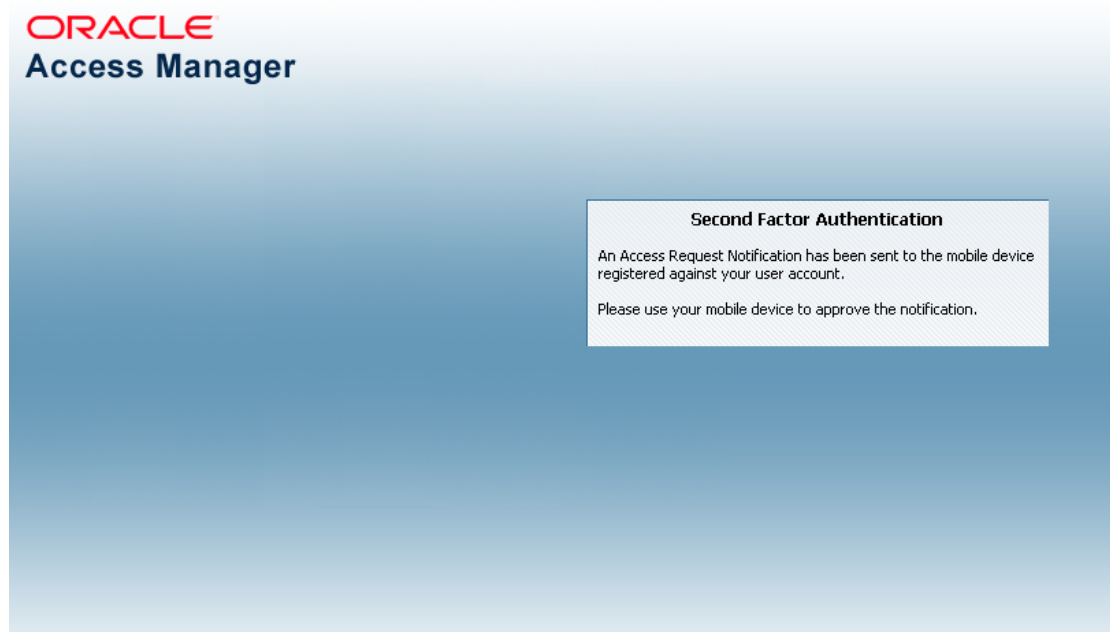
Figure 33-3 Access Request Notification Preferred Method Page



When the user selects Access Request Notification from the Second Factor Authentication Preferred Method Page, Access Manager sends an Access Request Notification to either the Apple Push Notification Server or the Google Notification Server depending upon the user's configured device. The notification server then pushes a notification to the mobile device and the user will approve or deny it. Based on a successful response, the user will be directed to the protected resource. On failure, an error message will be displayed and the user will be returned to the same OTP page.

Figure 33-4 shows the Access Request Notification message that is displayed during this process.

Figure 33-4 Access Request Notification Wait Screen



33.2.3 Using the Oracle Mobile Authenticator with OTP And Access Request

The user downloads the OMA app to the mobile device and configures it to receive the access request notification.

Depending on the selected option, the Adaptive Authentication Service will need to work in tandem with the Oracle Mobile Authenticator (OMA), a mobile device app that uses Time-based One Time Password and push notifications to authenticate users with the second factor authentication scheme. To receive the OTP or Access Request Notification using the OMA, a user downloads it to an Apple or Android mobile device and configures it by clicking a link provided by the Access Manager administrator. Access Manager and OMA must share a secret key.

See [Generating a Secret Key for the Oracle Mobile Authenticator](#) about the secret key.

See [Understanding Oracle Mobile Authenticator Configuration](#) on how to configure OMA.



Note:

The OMA app is not needed if using the OTP through Email or OTP through SMS options.

See [About using OTP through Email or SMS](#).

33.3 Understanding Adaptive Authentication Service and OMA Configurations

You need to configure the Adaptive Authentication Service and, depending on the option, the OMA.

To configure the Adaptive Authentication Service, perform the following procedures:

- See [Configuring an Adaptive Authentication Service](#).
- See [Understanding Oracle Mobile Authenticator Configuration](#).

33.4 Configuring an Adaptive Authentication Service

You can configure an Adaptive Authentication Service if you have already installed Access Manager, a WebGate, and Oracle HTTP Server (OHS).

Some of these configurations are specific to one or the other Adaptive Authentication Service options.

This section describes the following topics:

- [Generating a Secret Key for the Oracle Mobile Authenticator](#)
- [Configuring Oauth Services to enable the Secret Key API](#)
- [Configuring the Adaptive Authentication Plug-in in the Oracle Access Management Console](#)
- [Setting Credentials for UMS, iOS, and Android](#)
- [Creating a Java KeyStore for iOS Access Request \(Push\) Notifications](#)
- [Configuring Host Name Verifier for Android Access Request \(Push\) Notifications](#)
- [Configuring Access Manager for VPN in a Use Case](#)
- [Administering a Secret Key](#)

33.4.1 Generating a Secret Key for the Oracle Mobile Authenticator

A secret key needs to be shared between Access Manager and the OMA app. Businesses can generate secret keys in different ways so the means in which the secret key is generated is not important.

The following RESTful endpoint is used to generate the secret key for a user in the Oracle Access Management identity store.

```
http://<HOST>:<PORT>/oauth2/rest/resources/secretkey HTTP/1.1
```

Where, <HOST> and <PORT> are those of the OAM server.

In the case of OMA online configuration (which is Oracle's recommended method of configuration), OMA uses the RESTful endpoint to store the key for a user in the identity store. In the cases of OMA manual configuration or Google Authenticator, the administrator sets up a web application which allows the user to generate a secret key also using above mentioned RESTful endpoint. The secret key is stored as a String in an LDAP attribute in the identity store and the name of this attribute must be passed to

the business in the RESTful endpoint configuration before they generate the secret key.

See [Understanding Oracle Mobile Authenticator Configuration](#).

33.4.2 Configuring Oauth Services to enable the Secret Key API

There are three parts to enabling the Secret Key API.

The first part is to enable the secret key endpoint. The second part deals with enabling OAM configuration to enable the use of Time-based One-Time Password (TOTP). The third part deals with setting up OAM to produce a TOTP for a particular user Account.

To enable the Secret Key API:

1. Create DefaultDomain with customAttrs as described:
 - a. Ensure the following point to the same LDAP.
 - Authentication module. For example, LDAPScheme in the application domain
 - AdaptiveAuthenticationModule. For details, see [Creating an Authentication Policy](#)
 - AdaptiveAuthenticationPlugin
 - Default store. For example, **User Identity Store** under **configuration** in the OAM console.
 - b. Create DefaultDomain with customAttrs as shown. The following example shows OID as the identity store.

```
curl -u username:password -X POST
http://<Host>:<Port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/
oauthidentitydomain -H 'content-type: application/json' -d
'{"name":"DefaultDomain","identityProvider":"OID","enableMultipleResou
rceServe
r":false,"description":" Test
Domain","tokenSettings":
[{"refreshTokenEnabled":true,"refreshTokenLifeCycleEna
bled":true,"refreshTokenExpiry":5400,"lifeCycleEnabled":true,"tokenTyp
e":"ACCE
SS_TOKEN","tokenExpiry":1800},
{"refreshTokenEnabled":true,"refreshTokenLifeCyc
leEnabled":true,"refreshTokenExpiry":10800,"lifeCycleEnabled":true,"to
kenType"
:"AUTHZ_CODE","tokenExpiry":240}],
"customAttrs":{"\keyAttributeName\":"description\"}}'
```

Successfully created entity - OAuthIdentityDomain, detail - OAuth Identity Domain :: Name - DefaultDomain, Id - 5c75aeece3154fdc9af691e21b70b1b9, Description - Test Domain, TrustStore Identifiers - [DefaultDomain], Identity Provider - OID, TokenSettings - [{"tokenType":"SSO_LINK_TOKEN","tokenExpiry":3600,"lifeCycleEnabled":false,"refreshTokenEnabled":false,"refreshTokenExpiry":86400,"refreshTokenLifeCycleEnabled":false}, {"tokenType":"ACCESS_TOKEN","tokenExpiry":1800,"lifeCycleEnabled":true,"refreshTokenEnabled":true,"refreshTokenExpiry":5400,"refreshTokenLifeCycleEn

```
abled":true},
  { "tokenType": "AUTHZ_CODE", "tokenExpiry": 240, "lifeCycleEnabled": true, "refreshTokenEnabled": true, "refreshTokenExpiry": 10800, "refreshTokenLifeCycleEnabled": true }
], ConsentPageURL - /oam/pages/consent.jsp, ErrorPageURL - /oam/pages/servererror.jsp, CustomAttrs - { "keyAttributeName": "description" }
```

 **Note:**

Ensure that the attribute Name used is not used for other purposes

To delete a domain, use the following command:

```
curl -u username:password -X DELETE
http://<Host>:<Port>/oam/services/rest/ssa/api/v1/
oauthpolicyadmin/oauthidentitydomain?name=DefaultDomain
```

For information on parameters used in the command to create an identity domain, See [Creating an Identity Domain](#)

2. Test with Sample User that you are able to generate Secret Key. For example:
 - a. Create the following HTML file:

```
<html>
  <head>
    <title>Oracle Mobile Authenticator</title>
  </head>
  <body>
    <a href="oraclemobileauthenticator://settings?
LoginURL:=http://<Host>:<ManagedServerPort>/oauth2/rest/
resources/secretkey">
      Click Here
    </a>
  </body>
</html>
```

- b. Open this HTML page on your mobile device, on which you have the OMA app installed.
- c. Click on **Click Here**
- d. Enter the username and password of the user to access the protected resource
- e. The user is automatically added to the OMA app and the secret key is created. You will now start seeing OTP messages on OMA app on the mobile device.

Method 2

- a. Irrespective of whether the user is in OID or Useridentitystore1, create a secret key and then manually enter it in the OMA app on the mobile device.

```
curl -u username:password -X POST http://<Host>:<Port>/oauth2/rest/
resources/secretkey
-H 'cache-control: no-cache' -H 'content-type:application/x-www-form-
urlencoded'
-H 'x-oauth-identity-domain-name:DefaultDomain'
{"secret_key": "PTBHAI TFH25W3BOD"}
```

- b. Open the OMA app on your mobile device
- c. Click the + sign to add an account
- d. Click **Enter key manually**
- e. Type the key manually.

33.4.3 Configuring the Adaptive Authentication Plug-in in the Oracle Access Management Console

Access Manager provides the Adaptive Authentication Plug-in that you can use for two-factor authentication.

To configure the Adaptive Authentication plugin in the Oracle Access Management Console:

1. Log into the Oracle Access Management Console as System Administrator.
2. From the Application Security Launch Pad, click Authentication Plug-ins in the Plug-ins panel.
3. From the Authentication Plug-in tab, type *Adaptive* in the quick search box above the Plug-in Name column and hit Enter.

The AdaptiveAuthenticationPlugin is displayed.

4. Change the properties displayed under Plug-in Details: AdaptiveAuthenticationPlugin as applicable to your environment.

[Table 33-1](#) describes the Adaptive Authentication Plugin properties.

Table 33-1 Adaptive Authentication Plugin Properties

Property	Description	Default Value	Required for Challenge Method
IdentityStoreRef	Identity store name	Default_Store	All
TotpSecretKeyAttribute	Name of the user attribute in which the secret key is stored.	Attribute description	OTP using OMA, Time based OTP

Table 33-1 (Cont.) Adaptive Authentication Plugin Properties

Property	Description	Default Value	Required for Challenge Method
TotpTimeWindow	The number of OTP codes generated by the mobile device that Access Manager will accept for validation. Since the mobile device generates a new OTP every 30 seconds, if the value is 3, Access Manager will accept the current and last three OTPs generated by the mobile device.	3	OTP using OMA, Time based OTP
PushAPNsProdServer	If set to true, the APNS production server will be used to send notifications.	false	Access Request Notifications (iOS)
PushProxyHost	Name of the proxy host if notifications are to sent to the server using a proxy.		Access Request Notifications
PushProxyPort	Proxy port if notifications are to sent to the server using a proxy.	80	Access Request Notifications
PushProxyProtocol	Proxy protocol	https://	Access Request Notifications
UmsAvailable	When Adaptive Authentication Service requires UMS to send Email and SMS, set to true.	false	SMS, Email
UmsClientUrl	URL of UMS web service		SMS, Email
PhoneField	Attribute in the identity store where the user phone number is stored	mobile	SMS
EmailField	Attribute in the identity store where the user email address is stored	mail	Email
Totp_Enabled Email_Enabled Sms_Enabled Push_Enabled	Controls the options displayed in the UI. If enabled and user is not registered for Push, not setup for TOTP, or doesn't have email/phone populated in id store, those options won't be displayed. For example if user has not registered for TOTP and Push but has email populated then Email will be the only option shown.	true	NOTE: Properties should be set to false only when the Administrator wants to disable a particular feature for all users.

OTP REST Configuration Options

Table 33-1 (Cont.) Adaptive Authentication Plugin Properties

Property	Description	Default Value	Required for Challenge Method
ValidateAnyPin	<p>If <code>true</code>, user can submit any pin that has been generated for that user, if it is still valid. Pin is valid if it has not been successfully used and has not expired.</p> <p>If <code>false</code>, user can submit only the pin that matches the <code>correlationId</code> being submitted in order to validate the pin.</p>	false	
MaxAttempts	<p>Maximum attempts for a user to validate the pin.</p> <p>If <code>MaxAttempts</code> is exceeded, user must be reset.</p> <p>Value of 0, means no limit.</p>	0	
MaxPins	<p>Maximum number of pins to store for a single user.</p> <p>If more pins are requested after the maximum is reached, the oldest pin is replaced.</p>	5	
VerboseOutput	<p>If <code>true</code>, the REST output includes detailed information about errors.</p> <p>If <code>false</code>, the REST output includes minimal information about errors.</p>	true	
pinExpiry	<p>Availability of One Time Pin (OTP). The unit of measurement is a millisecond.</p> <p>The <code>pinExpiry</code> field in the <code>AdaptiveAuthenticationPlugin</code> governs the expiry of the Email/SMS OTP codes.</p>	300000	SMS, Email

 **Note:**

In addition to the properties (related to OTP generation in Adaptive Authentication plugin) specified in [Table 33-1](#), you can override settings in `oam-config.xml` by adding them to the `ConfigParams` section of the `OAMMFAOTP` definition.

This also allows for app-specific configuration by adding `app` as a prefix to the property name. For example, "app1.ValidateAnyPin" sets and validates any pin setting specifically for app1 without affecting the configurations for other applications.

5. Click Save.
6. Update the same properties as applicable in the AdaptiveAuthenticationModule by clicking Authentication Modules under Plug-ins in the Access Manager Launch Pad.

From the Authentication Modules tab, search for AdaptiveAuthenticationModule.

[Table 33-1](#) does not list all available Adaptive Authentication Service properties.

33.4.4 Enabling User Lockout During the Multi Factor Authentication Flow

You can lock the user after a fixed number of invalid attempts to login using incorrect PIN, during Second Factor Authentication.

The number of invalid attempts, is based on the value specified in `MaxAttempts` configured in Adaptive Authentication plugin on the OAM Console.

When user provides incorrect PIN for more than configured `MaxAttempts`, user account is locked using OAM password schema attributes.

To enable user lock out:

1. Add the `lockoutEnabled` property in the `oam-config.xml` file under the section for `AdaptiveAuthenticationPlugin`.

For example:

```
<Setting Name="28" Type="htf:map">
<Setting Name="globalUIOverride" Type="xsd:boolean">false</Setting>
  <Setting Name="instanceOverride" Type="xsd:boolean">false</Setting>
<Setting Name="value" Type="xsd:string">true</Setting>
<Setting Name="name" Type="xsd:string">LockoutEnabled</Setting>
  <Setting Name="length" Type="xsd:integer">100</Setting>
  <Setting Name="mandatory" Type="xsd:boolean">false</Setting>
  <Setting Name="type" Type="xsd:string">string</Setting>
</Setting>
```

For more information about editing the `oam-config.xml` file, see [Updating OAM Configuration](#)

2. Configure OAM password policy:
 - a. In the user identity store configuration in the console, enable Password Management
 - b. Configure the other attributes as required for OAM password policy. For details, see [Accessing Password Policy Configuration Page](#)
 - c. Ensure to extend the schema, as required, if it has already not been imported in the environment: `$OAM_HOME/idm/oam/server/pswdservice/ldif/OID_PWDPersonSchema.ldif`
 - d. Ensure that the Authentication module includes `PasswordManagementPlugin` to evaluate the OAM schema attributes used for locking the user after authentication.

33.4.5 Setting Credentials for UMS, iOS, and Android

Use the WLST command line script to set the credentials for the Oracle User Messaging Service (UMS), the iOS certificate or the Android API key.

These credentials are used by the OAM Server in the process of sending SMS/Email and push notifications. [Table 33-2](#) lists information that you need to complete the procedure.

Table 33-2 Server Side Configuration for Adaptive Authentication Service

Configuration	Information	Challenge Method
iOS Certificate/Password	https://developer.apple.com/library/mac/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html	Access Request (Push) notification using iOS
API Key	https://developers.google.com/web/updates/2015/03/push-notifications-on-the-open-web?hl=en	Access Request (Push) notification using Android
UMS Credential	UMS credentials that OAM will use to establish the connection to UMS Web service.	Email/SMS

To set credentials for UMS, iOS, and Android:

1. `cd <MW_HOME>/oracle_common/common/bin`
2. `./wlst.sh`
3. `connect()`
4. Enter the WebLogic user name and password when prompted.
5. Press Enter to accept the default URL or modify the host and port as necessary and press Enter.
6. Run one or more of the following commands to set credentials for the UMS server, iOS or Android depending on your deployment.

 **Note:**

Replace <UMS SERVER USER NAME>, <UMS SERVER PASSWORD>, <CERTIFICATE STORE PASSWORD> and <API KEY VALUE> with values specific to your environment. Do not change the values for any parameters in these commands but those listed and marked as variables.

- For OTP for email/SMS only:

```
createCred(map="OAM_CONFIG", key="umsKey", user="<UMS SERVER USER NAME>",
password="<UMS SERVER PASSWORD>")
```

For example:

```
createCred(map="OAM_CONFIG", key="umsKey", user="weblogic",
password="password")
```

- For Access Request (Push) Notifications on iOS only:

```
createCred(map="OAM_CONFIG", key="pushApnsCertKey", user="apnskey",
password="<CERTIFICATE STORE PASSWORD>")
```

For example:

```
createCred(map="OAM_CONFIG", key="pushApnsCertKey", user="apnskey",
password="password")
```

See [Creating a Java KeyStore for iOS Access Request \(Push\) Notifications when using iOS](#).

- For Access Request (Push) Notifications on Android only:

```
createCred(map="OAM_CONFIG", key="omaApiKey", user="omaApiKey",
password="<API KEY VALUE>")
```

For example:

```
createCred(map="OAM_CONFIG", key="omaApiKey", user="omaApiKey",
password="ADDGFDGDFGRTERSDFSDFSDFTYERTERTASDASD")
```

7. Verify the keys by logging into Fusion Middleware Control, navigating to Domain > Security > Credentials, and checking the OAM_CONFIG map for the keys input using the commands.

Note:

For information on how to update, delete or otherwise manage credentials using Fusion Middleware Control, see *Securing Applications with Oracle Platform Security Services*.

33.4.6 Creating a Java KeyStore for iOS Access Request (Push) Notifications

When using Access Request Notifications on iOS, create a Java KeyStore (JKS) by using the cert file and key file.

Once the JKS is created, rename it as `APNsCertificate.jks` and put it in the `<domain>/config/fmwconfig` directory of the Oracle Access Management installation. The JKS should contain the user's locally generated private key and the Apple Push Notification service (APNs) certificate downloaded from the Apple Developer Center.

The following sample commands generate and import the certificate:

```
openssl x509 -in aps_production.cer -inform DER -out aps_production.pem
-outform PEM
```

```
openssl pkcs12 -nocerts -in OMAKey.p12 -out OMAKey.pem
```

```
openssl pkcs12 -export -inkey OMAKey.pem -in aps_production.pem
```

```
-out iOS_prod.p12

keytool -import -keystore APNsCertificate.jks -file aps_production.cer
  -alias PushCert

keytool -importkeystore -destkeystore APNsCertificate.jks
  -deststoretype JKS -srcstoretype PKCS12 -srckeystore iOS_prod.p12
```

These commands assume:

- `aps_production.cer` to be the name of the APNs certificate downloaded from the Apple Developer Center.
- `OMASKey.p12` is the user's locally generated private key.

Also see [Setting Credentials for UMS, iOS, and Android](#).

 **Note:**

The section *Maintain Your Certificates, Identifiers, and Profiles* at the following Apple URL provides relevant information about app distribution certificates and APNs. <https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html>

33.4.7 Configuring Push Notifications on Mobile Device

This section provides steps to configure push notifications on a mobile device

- [Configuring Host Name Verifier for Android Access Request \(Push\) Notifications](#)
- [Creating OTPRestUserGroup and Modifying OAMMFAOTP](#)
- [Verifying Push Notification Settings](#)
- [Creating an Authentication Policy](#)
- [Connecting with Messaging Server](#)
- [Setting the GCM API key within the OAM Credential Store](#)
- [Installing the Google CA Files into the OAM Keystore](#)
- [Creating a Webpage to Deliver the OMA Application Profile to the Mobile Device](#)
- [Testing SFA through Push Notification](#)
- [Troubleshooting Push Notifications](#)

33.4.7.1 Configuring Host Name Verifier for Android Access Request (Push) Notifications

If you are setting up Android for Access Request notification, use the WebLogic console to update the WebLogic Managed Server for host name verification.

This step is required for Access Request notification configuration on Android only. It allows the verification of host names represented using wildcards; for example, `*.googleapis.com`.

To configure host name verifier for Android access requests (push) notifications:

1. Navigate to `base_domain -> Summary of Environment -> Summary of Servers -> oam_server1`.
2. Click the SSL tab.
3. Expand Advanced and select the Hostname verification entry to configure the Hostname Verifier.
4. Enter `weblogic.security.utils.SSLWLSWildcardHostnameVerifier` as the Custom Hostname Verifier.
5. Click Save.
6. Restart the `oam_server1`.

33.4.7.2 Creating OTPRestUserGroup and Modifying OAMMFAOTP

Create a group with the name `OTPRestUserGroup` and add the user member of this group to the LDAP used.

1. Export the `oam-config.xml` file from the database. See [Updating OAM Configuration](#) for details.
2. Change the LDAP store to the current LDAP store. That is, change the value of `UserIdentityStore1` from the following setting to the same LDAP name as provided in `User Identity Store` section in the `oamconsole`.

```
<Setting Name="OAMMFAOTP" Type="htf:map">
...
...
  <Setting Name="UserStore" Type="xsd:string">UserIdentityStore1</
Setting>
```

3. Import the `oam-config.xml`. See [Updating OAM Configuration](#) for details.
4. Restart admin and managed servers.

33.4.7.3 Verifying Push Notification Settings

Verify the push notification settings in the `AdaptiveAuthenticationModule` and `AdaptiveAuthenticationPlugin`

1. Login to the OAM console and navigate to the authentication modules section. Search for the `AdaptiveAuthenticationModule` and edit the following parameters in the Steps tab:
 - Ensure that the `SFATypes` includes Push.
 - Ensure that `Push_Enabled` is set to `true`.
 - Set the `IdentityStoreRef` to the name of the IDS profile that was created.

Note:

This needs to include `IDSPROFILE-` as a prefix. For example, if the IDS profile created is called `OUD` then enter the value as `IDSPROFILE-OUD`.

- Ensure that the `PushProxyHost`, `PushProxyPort`, and `PushProxyProtocol` are set correctly, if the OAM managed server needs to make a proxy server connection to reach the Google servers.
2. In the OAM console, navigate to the authentication plugins section. Search for the `AdaptiveAuthenticationPlugin` and make sure the same parameters and values from the previous step are reflected in the plugin too.

 **Note:**

In some cases OAM uses the authentication module and in some cases uses the plugin, so setting the same values in both ensures that the correct settings are used.

33.4.7.4 Creating an Authentication Policy

Create an authentication policy to protect the resource that contains the post-authentication rule to switch to the `AdaptiveAuthentication` scheme.

The adaptive authentication scheme needs to piggy-back on top of the existing `LDAPScheme`. The end-user will authenticate with a username/password and then a post-authentication rule will engage to "switch" to the `AdaptiveAuthenticationScheme`. Create the protected resource to use the authentication scheme `LDAPScheme` and then define a post-authentication rule to "switch" to the `AdaptiveAuthenticationScheme`. For example the below condition "`3>2`" is always true so all resources protected by this authentication policy will display the above SFA page.

33.4.7.5 Connecting with Messaging Server

Create a Google firebase project enabled for Google Cloud Messaging (GCM).

To send a push notification to the mobile device, OAM connects to the Google cloud messaging servers and delivers the notification to Google, which then delivers the notification to the mobile device. Google cloud messaging requires that a Google firebase project be created. The following information details how to create a Google project.

 **Note:**

The steps to create a firebase project are specific to Google and may change over time.

1. Login to the Google firebase console at <https://console.firebase.google.com/u/0/>.
2. Click the **Add Project** button and specify a name for the project. The name can be any text as it is not used by OAM. For example, `OMA-OAM`.
3. Navaigate to the project settings by clicking the **gears** icon next to the project name in the upper left-hand part of the browser.
4. Click on the **Cloud Messaging** option in the blue bar and make note of the `Sender ID` and `Server Key`, which is required in the later steps.

 **Note:**

Note that Google indicates that the legacy server key will work for some period of time. However, it is recommended that you use the longer server key.

33.4.7.6 Setting the GCM API key within the OAM Credential Store

The server key from the Google project needs to be saved within the OAM credential store so that OAM can use it along with the sender ID to make a connection to the GCM servers.

Create a new key using WLST as described:

1. Navigate to the `$MW_HOME/oracle_common/common/bin` directory and run `wlst.sh` command to connect to the AdminServer.
2. Run the following command to create a new key called `omaApiKey`, where the password value is the server key from the Google project created.

```
createCred(map="OAM_CONFIG", key="omaApiKey", user="omaApiKey",  
password("<API KEY VALUE>"))
```

 **Note:**

If the `omaApiKey` credential already exists then you can edit the key from within the EM console.

- a. Navigate to `Farm_base_domain/WebLogic Domain/<domain name>`
- b. Right-click and select **Security/Credentials**.
- c. Expand the `OAM_CONFIG` key and click on the **omaApiKey** to edit it with a new value

33.4.7.7 Installing the Google CA Files into the OAM Keystore

Google uses the GlobalSign certificate authority, therefore, the CA root certificates must be loaded into the trust keystore used by the OAM managed server.

The OAM server makes an SSL connection to the GCM server to send push notifications.

By default, OAM uses the `MW_HOME/wlserver_10.3/server/lib/DemoTrust.jks` keystore.

1. Collect the root certificates needed by running the following command. This returns the certificate chain that can be used to build files to load into the OAM keystore.

```
openssl s_client -connect android.googleapis.com:443 -showcerts
```

2. Once the certificate files are available load them into the `MW_HOME/wlserver_10.3/server/lib/DemoTrust.jks` keystore using the following command:

```
cd MW_HOME/wlserver_10.3/server/lib
keytool -importkeystore -srckeystore trust.jks -destkeystore DemoTrust.jks
-srcstorepass <password> -deststorepassDemoTrustKeyStorePassPhrase
```

3. Verify the presence of all of the certificates by running the following command

```
-keystore DemoTrust.jks -storepass DemoTrustKeyStorePassPhrase -storetype
jks'
```

4. Restart the OAM server to re-read the keystore files.

**Note:**

Google maintains a webpage related to their CA certificates. See <https://pki.google.com> for details.

33.4.7.8 Creating a Webpage to Deliver the OMA Application Profile to the Mobile Device

By default, the OMA application has no connections to an OAM server. A user-profile must be generated to connect the OMA application to an OAM server.

To connect the OMA application to an OAM server, load a configuration settings file into OMA.

1. Create an HTML file that contains a link to the configuration settings. This way the mobile device can access the web page in its browser and this will launch the OMA application automatically to configure the profile. For push notifications, the configuration settings need to include the following parameters:
 - `ServiceName` - Defines the name of the profile as it will be shown in the OMA application on the mobile device.
 - `ServiceType` - Defined whether to enable OTP or push notification or both. For push notifications, the type needs to be `Notification`
 - `PushPreferencesEndpoint` - Where push notification preferences must be sent, for example, to save the mobile device identification information.
 - `ChallengeAnswerEndpoint` - Where push notification responses must be sent, for example, to relay the allow/deny action by the user.
 - `SenderID` - The sender ID retrieved from the Google firebase project that was created. Note that the sender ID is different than the server key. The sender ID is used here in the configuration file while the server key is set as the `omaApiKey` in the credential store.
 - `NotificationAuthServerType` - The only supported value is `HTTPBasicAuthentication`

```
<html>
  <head>
```



```

        <title>OMA Configuration</title>
    </head>
    <body>
        <a href="oraclemobileauthenticator://settings?ServiceName
::=Google-OMA-
Push&ServiceType::=Notification&PushPreferencesEndpoint
::=http://hostname:14100/oam/services/rest/11.1.2.0.0/oma/
Preferences&ChallengeAnswerEndpoint
::=http://hostname:14100/oam/services/rest/11.1.2.0.0/oma/
ChallengeResponse&NotificationAuthServerType
::=HTTPBasicAuthentication&SenderID::=xxxx">Google OMA Push</a>
    </body>
</html>

```

2. Place this HTML file on a webserver that is accessible to the mobile device

 **Note:**

This URL is typically NOT protected by OAM. Since this is a registration link that users will need to register their device for use with second factor authentication, they need to be able to access this device before SFA is setup. The resource can be protected by a webgate, in that case, select an authentication scheme that does not engage the AdaptiveAuthenticationScheme, since the user will not have SFA yet configured for their account.

3. Install the OMA application onto the mobile device
The Oracle Mobile Authenticator application must to be installed on the device where the push notification are shown. The OMA application is available in both the Google Play Store and the Apple App Store as a free application.
4. Register the user account within the OMA application
The user must open a browser on the mobile device and access the specified OMA configuration settings web page. Since this is a configuration settings for OMA (as identified by oraclemobileauthenticator://settings) the OMA application must open up and ask the end-use to supply the LDAP credentials for OAM login. Once the credentials are validated by OAM the managed server OMA creates an OMA application profile.

 **Note:**

On the mobile device, navigate to **Setting > sound & notification > App notifications > Select OMA**. Turn on the **treat as priority** option.

33.4.7.9 Testing SFA through Push Notification

Test the push notification flow by accessing the protected resource and use the step-up authentication scheme to select the type of SFA required.

The user must see the name of the mobile device they are using.

Selecting the **Access Request Notification** option causes the OAM server to generate a push notification and deliver it to the GCM servers. That must show up on the users mobile device, where the user is asked to allow or deny the access.

By default, the OAM server checks for a user response every eight seconds. The browser briefly flashes as it waits until either the timeout period has passed and the browser displays an error message, or a user response is detected and the browser is redirected appropriately.

33.4.7.10 Troubleshooting Push Notifications

This section provides various errors, fixes, workarounds for troubleshooting the push notifications

Enable Logging

There are several things that can go wrong with this configuration as it relies on communication between the OAM server, Google servers, and a mobile device.

To debug issues the following logging modules must be enabled on the OAM managed server to get detailed output.

```
oracle.oam.plugin - TRACE:32
oracle.oam.admin.service.config - TRACE:32
```



Note:

The messages between OAM and GCM are logged only at TRACE:32.

401 Not Authorized Error

If you get a 401 not authorized error, while running the following command:

```
curl -u username:password -X POST
http://<Host>:<Port>/oam/services/rest/auth/api/v1/mfa/createOTP -H 'content-
type: application/json' -d
'{"userId":"user1","appName":"asasas","deliveryChannel":"none"}'
{"correlationId":"1ad7b4bd-4ac9-4960-9da8-a28cc2c8f856","resultCode":"0",
"validTime":"1544092972705","minorCode":"6571511656"}
```

Check if the directory server is used as the system store (by default, the WebLogic Server embedded directory used by `UserIdentityStore1`), the group `OTPRestUserGroup` is created and the `user1` is added in this group.

Push Notification is Never Received on the Mobile Device

Check the following:

- Ensure the mobile device is not connected to the VPN software.
- If the OMA application is not able to connect to the OMA server for an extended period of time.
- If the LDAP user account is used for both one-time passwords and push notifications.

Push Notification has Worked Before, but is Not Working Now

Delete the OMA application profile and re-register the user from the OMA application again.

To delete the existing profile long-click the profile name and then once it is highlighted, click the trash can option.

 **Note:**

This only works if push notifications have worked in the past. If the device has never successfully received a push notification then re-registration does not fix the problem and the OAM diagnostic logs must be reviewed.

Note that

User not found in the LDAP directory

During registration of the mobile device, the user is prompted to enter credentials to login. If the user is not found, the following error is seen in the OAM diagnostic log:

```
<Error> <oracle.oam.user.identity.provider>  
<OAMSSA-20142> <Authentication Failure for user user1, user not found  
in  
idstore UserIdentityStore1 with exception  
oracle.igf.ids.EntityNotFoundException: Entity not found for the search  
filter (&(objectclass=person)(uid=user1)).>
```

Note that the idstore listed is UserIdentityStore1. By default, OAM looks for users in the embedded WebLogic LDAP. If the error shows UserIdentityStore1, review the steps in [Creating OTPRestUserGroup and Modifying OAMMFAOTP](#). If the idstore listed is a custom store then check the ldapsearch that is generated to see why the user is not found. Once the push notification is received on the device the end-user needs to choose to allow or deny the login request.

If the user selects the allow option then the OMA application shows the confirmation screen and at the same time communicates back to the OAM managed server to indicate that the user has been permitted access

Once the OAM server receives the approval for login it then redirects to the requested URL.

33.4.8 Configuring Access Manager for VPN in a Use Case

You can configure Access Manager when a user needs to have access to a protected resource with VPN software.

To configure Access Manager for VPN in a use case:

1. Log into the Oracle Access Management Console as System Administrator.
2. From the Application Security Launch Pad, click Application Domains in the Access Manager panel.

The Application Domain tab is displayed.

3. Click Search to display all available Application Domains.
4. Click the Application Domain name that contains the resource being protected.

The Application Domain opens in a new tab.

5. Click Authentication Policies in the Application Domain tab.
6. Click the name of the Authentication Policy that is being used to protect the particular resource for which two factor authentication is being configured.

The appropriate Authentication Policy opens in a new tab.

7. Click Advanced Rules in the Authentication Policy tab.
8. Add a new rule by clicking the plus sign (+) under Post Authentication.

The Add Rule dialog is displayed.

9. Enter a Rule Name and the following jython script.

```
location.clientIP.startswith('10.')
```

See [Context Data for Advanced Rules](#).

10. Select the AdaptiveAuthenticationScheme Authentication Scheme from the If Condition is True drop-down list.

This Authentication Scheme will be used when the defined condition is true.

11. Click Add and then Apply to complete the procedure.

33.4.9 Administering a Secret Key

Enabling Secret Key Lifecycle

By default OAM only allows to create an OAuth secret key using the user name and password with the BASIC authorization header. It is possible to administer a secret key using an access token with the BEARER authorization header by enabling the Secret Key Lifecycle feature. This is done by using the `updateConfigProperty WLST` command in ONLINE mode.

The syntax to enable this feature is:

```
updateConfigProperty(propertyIdentifier="EnableSecretKeyLifecycle",propertyValue="true"
)
```

The syntax to disable this feature is:

```
updateConfigProperty(propertyIdentifier="EnableSecretKeyLifecycle",propertyValue="false"
)
```

Creating a Secret Key

To create a secret key use the following REST API:

```
curl -X POST http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/resources/secretkey -H 'Content-Type: application/x-www-form-urlencoded' -H 'x-oauth-identity-domain-name: <OAuthIdentityDomain>' -H 'Authorization:Bearer <AccessToken>'
```

Getting a Secret Key

To retrieve a secret key use the following REST API:

```
curl -X GET http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/resources/secretkey -H 'Content-Type: application/x-www-form-urlencoded' -H 'x-oauth-identity-domain-name: <OAuthIdentityDomain>' -H 'Authorization:Bearer <AccessToken>'
```

Updating a Secret Key

To update a secret key use the following REST API:

```
curl -X PUT http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/resources/secretkey -H 'Content-Type: application/x-www-form-urlencoded' -H 'x-oauth-identity-domain-name: <OAuthIdentityDomain>' -H 'Authorization:Bearer <AccessToken>'
```

Deleting a Secret Key

To delete a secret key use the following REST API:

```
curl -X DELETE http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/resources/secretkey -H 'Content-Type: application/x-www-form-urlencoded' -H 'x-oauth-identity-domain-name: <OAuthIdentityDomain>' -H 'Authorization:Bearer <AccessToken>'
```

Creating a Secret Key Using Basic Authentication

To create a secret key using Basic Authentication, use the following REST API:

```
curl -X POST http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/resources/secretkey -H 'Content-Type: application/x-www-form-urlencoded' -H 'authorization:Basic <Base64EncodedUsernamePassword>' -H 'x-oauth-identity-domain-name:<OAuthIdentityDomain>'
```

Configuring the Oracle Mobile Authenticator

The Oracle Mobile Authenticator is a mobile device app that uses Time-based One Time Password (TOTP) and push notifications to authenticate users with a two-factor authentication scheme. The Oracle Mobile Authenticator mobile device app must be configured to retrieve the secret key required to generate a One Time Password (OTP). The following sections contain configuration details when using the Oracle Mobile Authenticator app on an iOS, Android, or Windows mobile device.

- [Understanding Oracle Mobile Authenticator Configuration](#)
- [Using the Oracle Mobile Authenticator App](#)
- [Managing the Oracle Mobile Authenticator App](#)
- [Configuring the Google Authenticator App](#)

34.1 Understanding Oracle Mobile Authenticator Configuration

The Oracle Mobile Authenticator (OMA) app can retrieve a secret key required to generate a OTP or register with Access Manager to receive push notifications.

Provisioning the secret key can be done online or offline however registering for push notifications can only be done while online.



Note:

For details on the secret key, see [Generating a Secret Key for the Oracle Mobile Authenticator](#).

- Online Configuration uses the REST web services and the Mobile OAuth Services described in [Generating a Secret Key for the Oracle Mobile Authenticator](#) and [Configuring OAuth Services to enable the Secret Key API](#). Once enabled, the OMA app can invoke this service to get a secret key or register for push notifications. To invoke the REST web services, OMA needs to know its location URL. In this case, the Oracle Access Management administrator creates a web page to configure the OMA. When the user taps on the web page's link (provided via e-mail), it launches the OMA, passes the location URL to the app and the REST web services location is configured. The format of the location URL is as follows.

```
oraclemobileauthenticator://settings?ServiceName::=<name_of_service>
&ServiceType::=SharedSecret/Notification/Both&
SharedSecretAuthServerType::=HTTPBasicAuthentication/OAuthAuthentication
&LoginURL::=http://<host>:<port>/secretKeyURL
&NotificationAuthServerType::= HTTPBasicAuthentication
&PushPreferencesEndpoint::=http://<host>:<port>/preferencesURL
&ChallengeAnswerEndpoint::=http://<host>:<port>/challengeAnswerURL
&SenderID::=<senderID>
&OAuthClientID::=<clientID>
```

```
&OAMOAuthServiceEndpoint:=http://<host>:<port>/oauthserviceURL
&OAuthScope: =<OAuthScope>
```

Table 34-1 documents definitions for the location URL parameters.

Table 34-1 Location URL Parameter Definitions

Parameter	Definition
ServiceName	Name of the service. This name should be unique in OMA. If another configuration with same name is sent then it will prompt the user to overwrite the previous one
ServiceType	The type of service provided by this configuration i.e. one-time password, notification or a hybrid service which combines both one-time password and notification. Value can be SharedSecret, Notification or Both.
SharedSecretAuthServerType	The type of authentication by which shared secret provisioning REST endpoint is protected. Value can be HTTPBasicAuthentication or OAuthAuthentication.
LoginURL	The REST endpoint that provisions the shared secret for generating one-time passwords. The value specified for the LoginURL query parameter is based on the OAuth settings for Oracle Mobile Authenticator.
NotificationAuthServerType	The type of authentication by which notification registration endpoint is protected. Currently only HTTP basic authentication is supported thus the value is HTTPBasicAuthentication.
PushPreferencesEndpoint	The REST endpoint where push notification preferences should be sent.
ChallengeAnswerEndpoint	The REST endpoint where push notification responses should be sent.
SenderId	The Android sender ID for sending push notifications. The SenderID is only required on Android; it is not required when using iOS.
OAuthClientID	OAuth client ID if SharedSecretAuthServerType is set for OAuth
OAMOAuthServiceEndpoint	OAM OAuth service endpoint to get OAuth profiles available on the server.
OAuthScope	The OAuth scope required to access the shared secret.

 **Note:**

Oracle recommends using online configuration.

- Offline Configuration supports use cases in which the mobile device can not connect to the REST end point or the parameters needed to generate the OTP are different than the defaults. The Access Manager administrator sets up a web application which allows the user to generate or recreate a secret key. The user logs into this web application and, after authentication, the user is allowed to view the secret key and enter it in the OMA app manually. The secret key can also be delivered via an offline configuration URL so the administrator has the option of

changing the OTP generation parameters (time step, hashing algorithm and the like). The format of the offline configuration URL is:

```
oraclemobileauthenticator://settings?SharedSecretValue::=<secret_key>
&AccountName::=<username>&SharedSecretEncoding::=Base32/Base64String
&OTPAlgorithm::=TOTP
&HashingAlgorithm::=MD5/SHA-1/SHA-224/SHA-256/SHA-384/SHA-512
&OTPLength::=<lenght_of_OTP>&TimeStep::=<time_in_seconds>
```

Table 34-2 contains details regarding the parameters.

Table 34-2 Offline Configuration URL Parameters

Parameter	Description
SharedSecretValue	Mandatory value is the secret key
AccountName	Prompts the user for input if omitted
SharedSecretEncoding	Default is Base32
OTPAlgorithm	Default is TOTP
Hashing Algorithm	Default is SHA-1
OTPLength	Default is 6
TimeStep	Default is 30 sec

34.2 Using the Oracle Mobile Authenticator App

The Oracle Mobile Authenticator (OMA) app is a mobile device app that you can use as a second verification method by tapping **Allow** on the login request notification sent to your phone or by using the one-time passcode (OTP) that the app generates.

A mobile app uses either OTP or push notifications to prove that the user has possession of the mobile device. Only the mobile app that is in possession of the user's secret key can generate a valid OTP. You can download the Oracle Mobile Authenticator app from the app store.

OMA App Version	Mobile Platform Version
Version 4.0	iOS 7.1+
Version 8.0	Android 4.1+
Version 1.0	Windows 8.1+

34.2.1 Adding an Account to the OMA App by Scanning the QR Code

After you install the Oracle Mobile Authenticator (OMA) app, you can link the App to an account by scanning the Quick Response (QR) code.

In the case of offline configuration, it is assumed that the customer develops a web application and a user is authenticated by said application. The OMA scans the QR code which must have the shared secret, shared secret encoding information and optionally the OTP validity duration, the hashing algorithm to be used for TOTP or the length of the OTP (5 digits/6 digits).

The QR code needs to be created from any of the following configuration URLs:

- `oraclemobileauthenticator://settings?LoginURL::=http://OAMhost:port/oauth2/rest/resources/secretkey`
- `oraclemobileauthenticator://settings?AuthServerType::=HTTPBasicAuthentication&&LoginURL::=http://OAMhost:port/oauth2/rest/resources/secretkey&&ServiceName::=MyBank`

See [Understanding Oracle Mobile Authenticator Configuration](#)

Create the QR code manually using the configuration URLs you have received from your Administrator to proceed with account creation process. Also, you can receive the QR code directly from your Administrator and add an account just by scanning that QR code from the **Add Account** page.

To add an account to the OMA app:

1. Open the OMA app on your phone, and then tap **Add Account**.

 **Note:**

The OMA app may prompt you to enter the user name and password.

2. Click **Scan QR code to add account**.
3. Scan the QR code.

The account is added to the OMA App.

34.2.2 Adding an Account to the OMA App Using the Configuration URL

After you install the Oracle Mobile Authenticator (OMA) app, you can link the App to an account by tapping the configuration URL.

 **Note:**

You must perform these steps from your mobile device using a supported mobile browser: iOS – Safari, Android and Windows – Any mobile browser.

1. In your browser, open the configuration URL you have received from your Administrator.
2. If prompted, click **Open** to view the page in Authenticator.

The account is added to the OMA App.

34.2.3 Adding an Account to the OMA App by Entering the Key Manually

After you install the Oracle Mobile Authenticator (OMA) app on your device, you can link the App to an account by entering the key manually.

1. Open the OMA app on your phone, and then tap **Add Account**.

2. Tap **Enter Key Manually**.
3. On the **Add Account** page, fill in appropriate information and enter the key.
4. Tap **Save**.

The account is added to the OMA App.

34.2.4 Using the Oracle Mobile Authenticator App as an Authentication Method

After you enroll the Oracle Mobile Authenticator (OMA) app as a 2–Step Verification method, use it to provide a second method of verification to securely log in to applications.

1. Enter your user name and password in an Adaptive Authentication Service-protected environment.
2. Which authentication method that appears depends on the method that your Administrator has enabled:
 - a. Mobile App OTP

 **Note:**

Ensure that your device clock is synchronized.

- You are prompted to enter the OTP that is generated by the OMA app on your mobile device.
 - Tap the OMA app on your device to launch it.
 - Tap the account for which you want to generate a new OTP. An OTP for the account appears, and the countdown begins until a new OTP is automatically generated.
 - Enter or paste that OTP into the **OTP** box, and then click **Verify**.
- b. Mobile App Notification
 - You are prompted to open and respond to the notification that was sent to the OMA app on your mobile device.
 - Open the notification in the OMA app, and then tap **Allow**.

 **Note:**

Windows does not support notifications. You cannot enable or disable notifications if you are using a Windows phone.

34.3 Managing the Oracle Mobile Authenticator App

The Oracle Mobile Authenticator (OMA) app makes it easy for you to customize how you view your accounts, manage your PIN, and manage notifications.

Topics

- [Switching Between Grid View and List View in the OMA App](#)
- [Editing Accounts in the OMA App](#)
- [Reordering Accounts in the OMA App](#)
- [Deleting an Account in the OMA App](#)
- [Enabling OMA App Protection](#)
- [Changing Your OMA App PIN](#)
- [Disabling OMA App PIN Protection](#)
- [Managing Notification History in the OMA App](#)

34.3.1 Switching Between Grid View and List View

You can change how you view your list of accounts in the Oracle Mobile Authenticator (OMA) app.

1. Launch the OMA app, and then tap the menu icon in the upper-left corner.
2. Tap **Grid View** or **List View** to toggle between the two views.

Note:

For Windows phones, in the lower-right corner, tap the grid or list icon to toggle between the two views.

34.3.2 Editing Accounts in the OMA App

You can edit your accounts in the Oracle Mobile Authenticator (OMA) app.

iOS

1. While in List View, swipe left on the account tile that you want to edit. While in Grid View, swipe down.
2. Tap **Edit**. The Edit Account screen appears.

Note:

To edit an account when using VoiceOver mode, you must be in Grid View. The Edit option is not available in List View when using VoiceOver mode.

3. Make your changes, and then tap **SAVE**.

Android

1. While in List View, long tap the account that you want to edit. While in Grid View, tap the account, and then long tap it when it appears in detail view.
2. Tap the pencil icon that appears in the upper-right corner. The Edit Account screen appears.
3. Make your changes, and then tap **SAVE**.

Windows

1. Tap and hold the account tile that you want to edit. A menu appears.
2. Tap **Edit**. The Edit Account screen appears.
3. Make your changes, and then tap **Save**.

34.3.3 Reordering Accounts in the OMA App

You can change the order in which you view accounts in the Oracle Mobile Authenticator (OMA) app.

iOS

- While in List View, long tap the account to enter editing mode, and then hold the reorder icon on the right to drag. Tap **Done** when you finish.
- While in Grid View, long tap the account tile, and then drag (supported in iOS9 and above).

Android

- Tap and hold the account tile, and then drag it.

Windows

- While in List View, long tap the account tile. From the menu that appears, tap **Reorder**, and then drag.
- While in Grid View, long tap the account tile, and then drag.

34.3.4 Deleting an Account in the OMA App

You can delete accounts in the Oracle Mobile Authenticator (OMA) app.

iOS

1. While in List View, swipe left on the account tile that you want to delete. While in Grid View, swipe down.
2. Tap **Delete**. A Delete Account confirmation window appears.

 **Note:**

To delete an account when using VoiceOver mode, you must be in Grid View. The Delete option is not available in List View when using VoiceOver mode.

3. Tap **Delete Account**.

Android

1. Tap and hold the account tile that you want to delete.
2. Tap the trash can icon that appears in the upper-right corner.
3. In the Delete Account window, tap **Delete Account**.

Windows

1. Tap and hold the account tile that you want to delete. A menu appears.
2. Tap **Delete**. A Delete Account confirmation window appears.
3. Tap **Delete Account**.

34.3.5 Enabling App Protection

Add an additional level of security to the Oracle Mobile Authenticator (OMA) app by using an app PIN or by using biometrics such as Touch ID or Fingerprint to protect the app.

App PIN protection requires a PIN to unlock the OMA app before you can generate a one-time passcode (OTP) or approve a notification. Biometric protection requires Touch ID or Fingerprint verification to unlock the App before you can generate an OTP or approve a notification.

Note:

The OMA app does not support biometrics using a Windows device. Touch ID with the OMA App is only supported with iOS version 8 and higher.

1. To enable an app PIN:

Note:

Your application may require you to set up a PIN when you enroll.

- a. Launch the OMA app, and then tap the menu icon in the upper-left corner.
 - b. Tap **App Protection**.
 - c. Slide to enable PIN or Touch ID protection for the OMA app.
 - d. Enter your PIN at the prompt, enter it again to verify, and then tap **Done**.
2. To enable Biometrics:

 **Note:**

When you initially enable Touch ID or Fingerprint, you are prompted to set your PIN if you have not. If you have set your PIN, you are prompted to enter your PIN first before enabling Touch ID or Fingerprint.

- a. Enter your PIN at the prompt.
- b. Enter your PIN again to verify, and then tap **OK**.

The next time that you open the App, you are prompted to use your fingerprint to gain access to the OMA app.

34.3.6 Changing Your OMA App PIN

Change your PIN in the Oracle Mobile Authenticator (OMA) app.

1. Launch the OMA app, and then tap the menu icon in the upper-left corner.
2. Tap **App Protection**, and then tap **Change PIN**.
3. Enter the current PIN, the new PIN, confirm the new PIN, and then tap **Done**.

34.3.7 Disabling OMA App PIN Protection

You can disable PIN protection for the Oracle Mobile Authenticator (OMA) app.

 **Note:**

Your application may not allow you to disable PIN protection.

1. Launch the OMA app, and then tap the menu icon in the upper-left corner.
2. Tap **App Protection**, and then slide to disable PIN protection for the OMA app.
3. Enter your PIN, and then tap **Done**.

34.3.8 Managing Notification History in the OMA App

You can access and view details about your notification history in the Oracle Mobile Authenticator (OMA) app.

1. Launch the OMA app, and then tap the menu icon in the upper-left corner.
2. Tap **Notifications**. The Notification History page displays all notifications for the account.

 **Note:**

For the iOS platform, pending notifications that are currently in the Notification center of your device do not appear in the OMA app when you manually launch the OMA app.

3. Tap a notification to view login request details.

34.4 Configuring the Google Authenticator App

The Google Authenticator app only supports manual configuration.

To initiate configuration in the Google Authenticator app, the user creates an account for two-factor authentication using the app. After account creation, the user manually enters the secret key received from the resource owner. (For details on the secret key, see [Generating a Secret Key for the Oracle Mobile Authenticator](#).) Additionally, ensure that TOTP is enabled at the bottom of the Google Authenticator screen. Google Authenticator generates the OTP code in an offline, disconnected mode; it does not interact with Access Manager.

Configuring TOTP-based Multi Factor Authentication in OAM

This section provides details for configuring TOTP-based Multi Factor Authentication (MFA) in OAM



This content applies only to OAM Bundle Patch 12.2.1.4.210920 and later releases.

Perform the following steps to configure TOTP-based MFA in OAM:

1. Configure MFA using the `configureMFA` command with `config-utility.jar`. For example:

```
$JAVA_HOME/bin/java -cp $ORACLE_HOME/oam/server/tools/config-utility/
config-utility.jar -Doracle.net.tns_admin=/u01/IDMTOP/config/domains/
IDMDomain/config/
jdbcoracle.security.am.migrate.main.ConfigCommand $DOMAIN_HOME
configureMFA $DOMAIN_HOME/propertyfile
```

The `propertyfile` must include the following properties:

```
oam.entityStore.schemaUser=<schemaUser>
oam.entityStore.ConnectionString=jdbc:oracle:thin:@//<connection string>
oam.entityStore.schemaPassword=<Password>
oam.user.store="<identityStoreName>"
oam.user.role="<RequiredRolename>"
```

Note:

- `oam.user.store` is optional. If this is not specified, the default Identity Store is used.
- `oam.user.role` must be specified with the correct role name of the Administrator.

2. Set the Post-Authentication Rule
 - a. Log in to the Oracle Access Management Console as Administrator.
 - b. In the Oracle Access Management Console, click **Application Security** at the top of the window.
 - c. In the Application Security console, click **Application Domains**.

- d. Search and select the required Application Domain
 - e. In the Application Domain window, click **Authentication Policies** tab, search and select the required Authentication Policy
 - f. In the Authentication Policy window, click **Advanced Rules** tab.
 - g. Under **Post Authentication**, click the plus sign (+) to add a new rule.
 - h. Specify the following details in the **Add Rule** window:
 - i. **Rule Name:** Specify a name for the rule
 - ii. **Condition:** Add `config.configMap['MFAEnabled'] == 'true'` in this field
 - iii. **If condition is true *Switch Authentication Scheme to:** Select **AdaptiveAuthenticationScheme** from the dropdown.
 - iv. Click **Add** to apply this rule.
 - i. Click **Apply** to save this updated authentication policy
3. Enable MFA using the following REST API:

```
https://<ManagedServerHost>:<ManagedServerPort>/oam/services/rest/  
access/api/v1/control/feature?state=enable&componentName=mfa
```

For more information, see [REST API to Enable MFA in Oracle Access Manager](#) .

Also see, [Configuring the Oracle Mobile Authenticator](#)

Part IX

Managing the Oracle Access Management OAuth Service and OpenIDConnect

The Oracle Access Management OAuth Service allows organizations to implement the open standard OAuth 2.0 Web authorization protocol in an Access Manager environment. OAuth enables a client to access Oracle Access Manager (OAM) protected resources that belong to another user (that is, the resource owner). OpenIDConnect implements authentication as an extension to the OAuth 2.0 authorization process. It provides easily consumable ID Tokens that are obtained by Clients using OAuth 2.0 flows.

OpenIDConnect provides an identity layer on top of OAuth 2.0 protocol. It allows clients to:

- Verify the identity of the end-user based on the authentication performed by an Authorization Server.
- Obtain profile information in an interoperable REST-like manner.

Part IX contains the following chapters:

- [Understanding OAuth Services](#)
- [Configuring OAuth Services in 12c](#)
- [Understanding OpenIDConnect](#)
- [OIDC Client Integrations with Social Identity Providers](#)

See Also:

- <http://openid.net/connect/>
- http://openid.net/specs/openid-connect-core-1_0.html

36

Understanding OAuth Services

OAuth provides a method to exchange identity credentials for an access token. This token, in return, can be used for granting access of private resources in a user's account on one service provider site to a second, consumer site without having to divulge the identity credentials to the consumer site.

Oracle Access Management implements the OAuth Core 2.0 specifications to offer OAuth Services. This chapter describes the purpose and capabilities of the Oracle Access Management OAuth Services.

This section describes the following topics:

- [About Oracle Access Management OAuth Services](#)
- [Understanding OAuth Services Authorization for Web Clients](#)
- [Understanding the OAuth Services Components](#)
- [About OAuth Tokens](#)

36.1 About Oracle Access Management OAuth Services

OAuth is an open standard authorization protocol that provides authentication and access control between a Client (such as Web services) and a Resource Owner (or Service Provider) on the Web.

Oracle Access Management OAuth Services is based on this standard and designed:

- To address enterprise-level extranet use cases.
- To provide secure access to APIs.
- To leverage built-in Oracle Access Management features (including authentication schemes, strong authentication, fraud detection, session management and federated authentication).
- To secure confidential clients with a high level of security.

Oracle Access Management OAuth Services are available for Web clients. OAuth Services for Web clients implement the standard OAuth 2.0 use cases. In this case, the clients rely on a Client ID/Client Password (or secret) to secure itself. For an example, see <http://tools.ietf.org/html/rfc6749#page-4>.

See Also,

- [Understanding OAuth Services Authorization for Web Clients](#)

36.2 Understanding OAuth Services Authorization for Web Clients

In the most common OAuth scenario, the Client accessing the protected resource is issued a different set of credentials than those of the user. (In this case, the user does not disclose

their credentials to the client.) Oracle Access Management OAuth Services acts as the intermediary Authorization Server, interacting directly with the Client, the service hosting the user's protected resource (Resource Owner) and the server on which the resource is located (Resource Server).

It issues access tokens to a Client that has (already) successfully authenticated with the Resource Server - in effect, authorizing the client to access private resources or activities on the server. A single Authorization Server instance can issue access tokens accepted by multiple resource servers.



Note:

OAuth does not impose special requirements on the interaction between a Resource Server and an Authorization Server.

The following sections describe web-based scenarios in which OAuth Services works.

- [Understanding 3-Legged Authorization](#)
- [Understanding 2-Legged Authorization](#)

The scenarios introduce the concept of OAuth Services endpoints. The scenarios also use the following terms:

- The Resource Owner is an entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user.
- The Client is an application making protected resource requests on behalf of the resource owner and with its authorization.
- OAuth Services refers to the Authorization Server, Oracle Access Management.
- The Resource Server is the machine on which the protected resource is stored. It can be any website or Web service where restricted resources are located; for example, a photo sharing site, a blogging platform and an online bank service control access to private resources and activities. The Resource Server is deployed in a different location from Oracle Access Management and the Client. The Resource Server needs to be capable of accepting and responding to protected resource requests using access tokens.

36.2.1 Understanding 3-Legged Authorization

In 3-legged authorization, the Resource Owner grants access to an OAuth-enabled Client to request access to resources stored on an OAuth protected Resource Server.

Oracle Access Management OAuth Services validates the Resource Owner's identity and presents the owner with a consent form in a Web browser when approval is required.

The third leg in this authorization scheme is the step in which the user grants or denies the client access.

If Consent Management is enabled, subsequent OAuth 3-legged flows skip consent processing. For details, see [Enabling Consent Management](#).

The following text has more details and [OAuth 3-Legged Flow Diagram](#) illustrates the process.

 **Note:**

A WebGate proxy is required to use 3-legged authorization with an external LDAP directory server.

1. The Resource Owner (user) undertakes an action in the user-agent (a browser, for example) that requires the Client web service (or app) to access protected resources belonging to the user on a different site.
2. The Client initiates the OAuth flow by invoking the OAuth Services authorization endpoint to get a request token. The Client sends its identifier, the requested scope, and a redirection URI to which the Authorization Server will direct the user-agent once access is granted or denied.
3. OAuth Services redirects the user-agent to request the Resource Owner's password credentials.
4. Access Manager displays a login page requesting a user name and password from the Resource Owner. OAuth Services supports all authentication schemes provided by Access Manager.
5. The Resource Owner enters a user name and password.
6. Access Manager validates the credentials, returns a request token and redirects the user-agent to OAuth Services.
7. OAuth Services determines that the Resource Server requires the user's consent before the authorization code can be sent to the Client.
8. OAuth Services displays the user consent form.

 **Note:**

If Consent Management is enabled, the consent form is displayed only during the first 3-legged authorization flow. Once the user grants access, the consent gets persisted in the database and OAM skips the consent on subsequent 3-legged OAuth flows.

Web-based clients require the consent form to be protected by a WebGate.

9. The user approves the request.
10. OAuth Services returns an authorization code to the Client using the redirection URI.

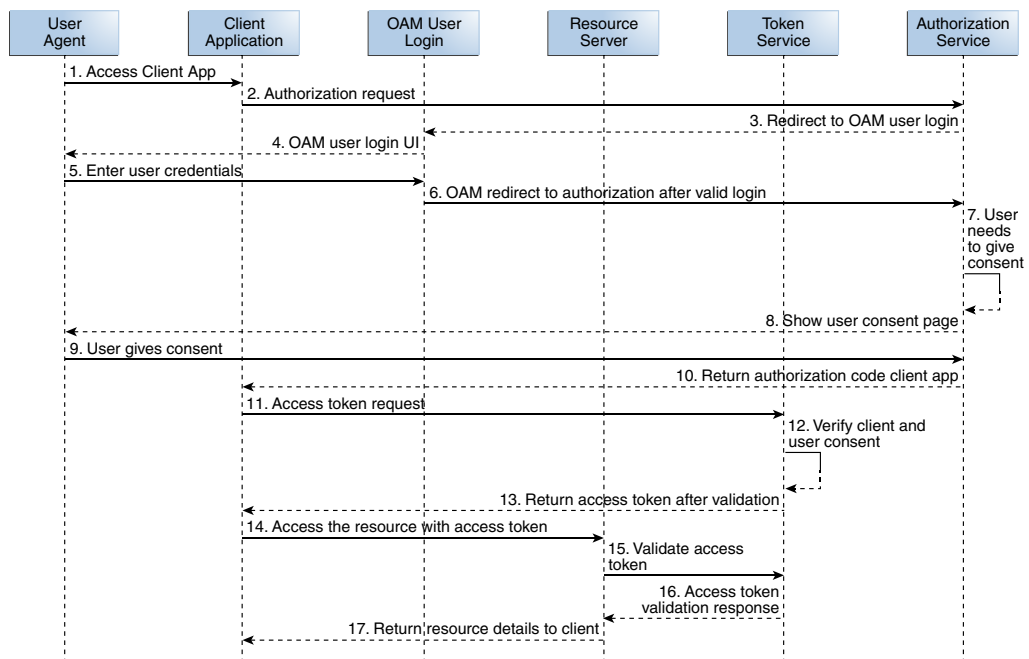
 **Note:**

The Authorization Code grant type is required for 3-legged authorization. See [Clients](#) for details.

11. The Client sends the authorization code in a POST request (including the redirection URI used to obtain the authorization code for verification) to the token endpoint and requests an OAuth access token. When making the request, the Client authenticates with OAuth Services.

12. If the client type requires client credentials, the OAuth Services authenticates the client credentials, validates the authorization code, and ensures that the redirection URI received matches the URI previously used to return the authorization code. OAuth Services also validates the requested scope based on the Resource Server's configuration and the user's consent details.
13. OAuth Services returns an access token to the Client.
A refresh token may also be returned with the access token if the client sends a refresh token request. For more information, see [About OAuth Tokens](#).
14. The Client presents the access token to the Resource Server.
15. The Resource Server validates the access token by sending a request to the OAuth Services token endpoint and waits for a success or failure response.
16. OAuth Services validates and sends the token success or failure response back to the Resource Server.
17. If the token is deemed valid, the Resource Server returns the requested resource to the Client.

Figure 36-1 OAuth 3-Legged Flow Diagram



36.2.2 Understanding 2-Legged Authorization

In 2-legged authorization, the OAuth Client is pre-approved to access resources; thus, the user consent form step (described in [Understanding 3-Legged Authorization](#)) is not required. In this scenario, Access Manager returns an access token to the Client based on the requested grant type and the client credentials. Since the client is already registered to request for specific scopes, OAuth Server returns an access token to the client without displaying the consent form. This arrangement fits a service-to-service model, especially when the requesting service (Client) and the Resource

Server are in a close partnership and Resource Owner approval is either assumed or not required.



Note:

The Client Credentials grant type or the Resource Owner Credentials grant type are required for 2-legged authorization. See [Creating a Client](#) for more information.

36.3 Understanding the OAuth Services Components

The following sections contain information about the Identity Domains configuration options. See [Configuring OAuth Services in 12c](#) for details on configuring these components.

- [Identity Domains](#)
- [Clients](#)

36.3.1 Identity Domains

Identity Domains are entities that contain all artifacts required to provide standard OAuth Services .

Each Identity Domain is an independent entity. One of the primary use cases of the Identity Domain is for multi tenants deployments. Each Identity Domain will correspond to a tenant. This can apply to different departments in an organization if there is a need for independence. This will also be useful for cloud deployments where each Identity Domain can correspond to a separate tenant or entity. The following artifacts are just some of the components configured within an OAuth Services Identity Domain.

- One or more Clients
- One or more Resource Servers

For information on configuring Identity Domains, See [Creating an Identity Domain](#).

36.3.2 Clients

The Client initiates the OAuth protocol by invoking the OAuth Services.

Client profiles must be created before the protocol can be initiated. At a minimum, client profiles include the application name, a client ID, and one or more URIs to which OAuth Services will redirect the user-agent once access is granted or denied. Clients may be public or confidential.

- Web clients are a type of confidential client, assigned with a client ID and secret. These clients can interact with the OAuth Services server by sending the client ID and secret as part of an authorization header. It is up to each individual client to determine how the secret issued to them is securely stored.
- Public clients are assigned with a client ID but no secret. Typically these profiles pertain to browser based applications like Javascript or can be mobile based apps.

The client ID and secret are explained in the following bullet points.

- The Client ID is a unique string that represents the registration information and is required for each client. You can create a unique client ID or have OAuth Services generate one. OAuth Services compares the defined Client ID with the value the client sends over HTTPS or HTTP as part of an authorization request. If the values do not match, the request is rejected. Client IDs are Base64 encoded when they are sent as authorization header.
- The Client secret is the client password. You can create a unique client secret or have OAuth Services generate one. Web clients are required to have a Client ID and a Client secret. Public clients, on the other hand, do not have a client secret and are given only a Client ID.

To request an access token, the client obtains authorization from the resource owner. The authorization is expressed in the form of an authorization grant, which the client uses to request the access token. The OAuth 2.0 specification provides authorization grant types for different security use cases. OAuth Services has implemented some of these grant types. Web and Public Clients can access the various OAuth Services grant types that are appropriate to them. The following grant types are supported by OAuth Services.



Note:

(For general information about the OAuth specification grant types, see <http://tools.ietf.org/html/rfc6749#section-1.3>.)

- Authorization Code - The Resource Owner logs in using Oracle Access Management. The token endpoint exchanges the authorization code along with client credentials for an access token. The Authorization Code grant type is required for 3-legged flows.
- Resource Owner Credentials - The Resource Owner provides the client with a user name and password. This is only suitable for highly trusted client applications because the client could abuse the password, or the password could unintentionally be disclosed to an attacker. Per the OAuth 2.0 specification, the authorization server and client should minimize use of this grant type and utilize other grant types whenever possible. The Resource Owner Credentials grant type is required for 2-legged authorization scenarios.
- Client Credentials – The client requests an access token using only its client credentials (or another supported means of authentication). This is suitable if the client is requesting access to protected resources under its control, or those of another resource owner when previously arranged with the authorization server. The Client Credentials grant type is required for 2-legged authorization scenarios.
- Refresh Token - Select this option to return a refresh token together with an access token in the token response. See [About OAuth Tokens](#) for more information.
- JWT Bearer - Allows a JWT assertion to be used to request an OAuth Services access token.

Allowed Scopes can also be configured on a client by client basis. OAuth Services allows for the configuration of scopes to bypass the need for user consent. For information on configuring Clients, see [Creating a Client](#).

36.4 About OAuth Tokens

OAuth generates two types of tokens namely, Access Tokens and Refresh tokens.

Access tokens carry the necessary information to access a resource directly. In other words, when a client passes an access token to a server managing a resource, that server can use the information contained in the token to decide whether the client is authorized or not. Access tokens usually have an expiration date and are short-lived.

Refresh tokens carry the information necessary to get a new access token. In other words, whenever an access token is required to access a specific resource, a client may use a refresh token to get a new access token issued by the authentication server. Common use cases include getting new access tokens after old ones have expired, or getting access to a new resource for the first time. Refresh tokens can also expire but are rather long-lived.. The following sections contain additional details.

- [OAuth Access Tokens](#)
- [OAuth Refresh Tokens](#)
- [OAuth Token Revocation](#)

36.4.1 OAuth Access Tokens

Before the user accesses the protected resource, the client service provider authenticates with OAuth Services and requests the Token Endpoint for an OAuth Access Token.

If OAuth Services determines that a user must consent to the request for access to a protected resource, a consent form is displayed. After the user consents, OAuth Services returns an authorization code to the Client service provider. The Client then sends the authorization code to the Token Endpoint and requests an OAuth Services Access Token. (When making the request, the Client authenticates with OAuth Services.) If received, the Access Token allows access to the protected resources.

Oracle Access Management can embed custom attributes in Access Tokens. Custom attributes are configured as part of the Client Profile or the Custom Resource Server. They are defined as static or dynamic.

- **Static Attributes** - Attribute name and value pairs where the value is fixed at the time that you define the attribute. For example, `name1=value1`.
- **Dynamic Attributes** - User-profile specific attributes. For example:

```
$session.id,  
{ "attrName": "sessionId",  
  "attrValue": "$session.id",  
  "attrType": "DYNAMIC"  
}
```

[Creating a Resource](#) and [Creating a Client](#) contain more information. Keep the following guidelines in mind when configuring custom attributes:

- Do not use the same name for a static and dynamic attribute.
- Avoid using the same name when adding custom attributes to the service profile configuration and the scope configuration. If you define the same attribute name in both locations, the scope-based attribute value takes precedence.

Custom attributes appear as claims in access tokens. JWT-based access tokens contain standard JWT claims along with OAuth Services specific ones. For example:

- **Standard**

```
"exp":1357596398000,  
"iat":1357589198000,  
"aud":"oam_server1",  
"iss":"OAuthServiceProfile",  
"prn":null,  
"jti":"340c8324-e49f-43cb-ba95-837eb419e068",
```

- **OAuth Services Specific**

```
"oracle.oauth.user_origin_id":"john101",  
"oracle.oauth.user_origin_id_type":"LDAP_UID",  
"oracle:idm:claims:client:macaddress":"1C:AB:A7:A5:F0:DC",  
"oracle.oauth.scope":"brokerage",  
"oracle.oauth.client_origin_id":"oathsssoapplid",  
"oracle.oauth.grant_type":"oracle-idm:/oauth/grant-type/resource-access-  
token/jwt"
```

These claims are available as part of the access token generated by OAuth Services. Because the custom attributes appear as claims in a JWT-based access token, the following naming restrictions apply:

- Avoid JWT standard claim names.
- Avoid names with an "Oracle" prefix (as shown above)

36.4.2 OAuth Refresh Tokens

OAuth Services can be configured to allow the Client to use a refresh token to obtain additional access tokens with identical or narrower scope.

The refresh token is used when the access token is no longer valid. The purpose of a refresh token is to improve security. Access tokens are short-lived, so if stolen, they are only useful for a limited period. Refresh tokens are longer-lived, but are less frequently sent to the server, thus reducing the likelihood that they will be stolen.

Refresh tokens are generated only for Resource Owner Credentials and Authorization code flow. This is also controlled by the tokenSettings configuration in the IdentityDomain.

36.4.3 OAuth Token Revocation

OAM provides enhanced security to the OAuth flow by supporting revocation of access and refresh tokens.

See [Revoking OAuth Tokens](#) for details.

37

Configuring OAuth Services in 12c

Oracle Access Management (OAM) OAuth helps secure access to services. OAuth services are enabled as a part of the OAM 12c installation process. OAM provides an API based approach for configuring OAuth Services. During set up, you need to configure OAuth clients and resources in Oracle Access Manager.

This chapter contains the following sections:

- [Set-up OAuth Services](#)
- [Configuring OAuth Services Settings](#)
- [Enabling User Lock Validation](#)
- [Enabling Consent Management on MDC](#)
- [Configuring OAuth in Multi-Data Centers](#)
- [Optional Parameters for Consent Management in Multi-Data Centers](#)
- [Error Codes and Troubleshooting Steps for Consent Management on MDC](#)
- [Dynamic Client Registration](#)
- [SSO Session Linking for OAuth Tokens](#)
- [Runtime REST APIs for OAuth 12c](#)
- [Revoking OAuth Tokens](#)
- [Configuring Client Authentication](#)
- [Configuring mTLS Client Authentication](#)
- [Proof Key for Code Exchange \(PKCE\) Support in OAM](#)
- [Revoking OAuth Tokens](#)
- [Token Exchange Support in OAM](#)

37.1 Set-up OAuth Services

You are an administrator and are responsible for setting up OAuth. You want to configure OAuth to secure access to services. During set up, you need to configure OAuth clients and resources in Oracle Access Manager. This section describes how to enable and manage OAuth Services using APIs.

You have the following responsibilities as an administrator.

- Configure and manage OAuth Identity Domain
- Configure and manage OAuth Resources
- Configure and manage OAuth Clients
- Ensure that the communication between different services is secure
- Access protected services through REST API calls

Pre-requisite for OAuth configuration

- Ensure you have the required OAuth Administrator permissions
- Ensure the 12c environment is installed. See *Installing the Oracle Identity and Access Management Software in Fusion Middleware Installing and Configuring Oracle Identity and Access Management*
- Ensure OAuth and OpenIDConnect Service is enabled in Available Services of the Configuration Section. See [Available Services of the Common Configuration Section](#)
- Configure OAuth Resources and Clients
- Obtain a Client Access Token

Setting Up OAuth: Task Flow

This section describes the high-level tasks in setting up OAuth in OAM. You start setting up OAuth by creating an identity domain and registering a resource. An OAuth Resource needs to be registered before registering an OAuth Client, as the resource information, specifically the API details, are required while registering a client.

1. To create an identity domain using REST API calls, refer to [Creating an Identity Domain](#)
2. To register a new resource using REST API calls, refer to [Creating a Resource](#)
3. After a resource is registered, you can configure and register an OAuth Client. To register a trusted client using REST API calls, refer to [Creating a Client](#)

For more information on OAuth REST APIs, See REST API for OAuth in Oracle Access Manager

37.2 Configuring OAuth Services Settings

OAuth Services has many components that must be configured before the authorization protocol can be used.

Descriptions of the OAuth Services components and how they work together can be found in [Understanding the OAuth Services Components](#). This section includes information on configuring the OAuth Services components.

This section describes the following topics:

- [Creating an Identity Domain](#)
- [Creating a Resource](#)
- [Creating a Client](#)

37.2.1 Creating an Identity Domain

An Identity Domain corresponds to the notion of a tenant. All clients and resource servers are created under an Identity Domain.

The important parameters used in the `curl` command to create an identity domain are:

- **identityProvider:** UserIdentityStore to perform the authentication against (Password Grant Flows). If not specified this is defaulted to the DefaultIdentityStore - "UserIdentityStore1"
- **errorPageURL:** Custom error page to be used in the case of 3 legged flows. If not specified it is defaulted to OAM server's error page.
- **consentPageURL:** Customer consent page to be used in case of 3 legged flows. If not specified uses the custom consent page shipped with OAM.
- **tokenSettings:** Token defaults are maintained at the IdentityDomain level. If tokenSettings is not specified the default values for the ACCESS_TOKEN and others are used.

 **Note:**

If RefreshToken needs to be generated along with AccessToken, `refreshTokenEnabled=true` must be set, under ACCESS_TOKEN settings.

Endpoint for CRUD operations:

```
http:<AdminServerHost:Port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/
oauthidentitydomain
```

 **Note:**

Use `Content-Type:application/json` in the REST API HTTP request.

There are 2 ways to create the Identity Domain

- **Simple:** In this mode, just the name and description of the IdentityDomain to be created are used. The rest of the values are defaulted.
 - **Detailed:** In this mode, you can give specific values to the different parameters.
1. In Simple mode, a sample `curl` command to create a domain is shown.

```
curl -i -H 'Content-Type: application/x-www-form-urlencoded' -H
'Authorization:Basic d2VibG9naWM6V2VsY29tZTE='
--request POST http:<Servername>:<Port>/oam/services/rest/ssa/api/v1/
oauthpolicyadmin/oauthidentitydomain -d
'{"name":"TestDomain","description":"Test Domain"}'
```

2. In Detailed mode, a sample `curl` command to create a domain using scopes is shown below.

```
curl -i -H 'Content-Type: application/x-www-form-urlencoded' -H
'Authorization:Basic d2VibG9naWM6V2VsY29tZTE=' --request
POST http:<Servername>:<Port>/oam/services/rest/ssa/api/v1/
oauthpolicyadmin/oauthidentitydomain -d
'{"name":"TestDomain","identityProvider":"UserIdentityStore1","description
```

```

": "Test Domain"}'

HTTP/1.1 200 OK
Date: Fri, 28 Jul 2017 13:01:41 GMT
Content-Length: 860
Content-Type: text/plain
X-ORACLE-DMS-ECID: 78d30c19-07b6-4ac2-a39b-f1cbd8182ebb-000003fd
X-ORACLE-DMS-RID: 0
Set-Cookie:
JSESSIONID=_oGJSc7Vt2vIWLNQ_uwYCZz151JqOXewJRIkyvstnnio8WsNborT!-187
5566563; path=/; HttpOnly

Sucessfully created entity - OAuthIdentityDomain, detail - OAuth
Identity Domain :: Name - TestDomain,
Id - 1636d0492f36447087780abdfdc4c15f, Description - Test Domain,
TrustStore Identifiers - [TestDomain],
Identity Provider - UserIdentityStore1, TokenSettings -
[{"tokenType": "ACCESS_TOKEN", "tokenExpiry": 3600, "lifeCycleEnabled":
false, "refreshTokenEnabled": false,
"refreshTokenExpiry": 86400, "refreshTokenLifeCycleEnabled": false},
{"tokenType": "AUTHZ_CODE", "tokenExpiry": 3600, "lifeCycleEnabled": fals
e, "refreshTokenEnabled": false, "refreshTokenExpiry": 86400,
"refreshTokenLifeCycleEnabled": false}, {"tokenType": "SSO_LINK_TOKEN"
, "tokenExpiry": 3600, "lifeCycleEnabled": false,
"refreshTokenEnabled": false, "refreshTokenExpiry": 86400, "refreshToken
LifeCycleEnabled": false}],
ConsentPageURL - /oam/pages/consent.jsp, ErrorPageURL - /oam/pages/
error.jsp, CustomAttrs - null

```

37.2.1.1 Token Signing Using Third-Party Certificates

Access tokens can be signed using a self-signed key pair generated out-of-the-box. OAM extends this support to allow signing of access tokens using third-party key pairs. Administrators can manage the life-cycle of the key pair using REST APIs.



This content applies only to OAM Bundle Patch 12.2.1.4.210920 and later releases.

To allow signing of access tokens using third-party certificates, perform the following steps:

1. Upload the required key pair to the server and create an alias for that key pair using the `https://<admin-host>:<admin-port>/oam/services/rest/ssa/api/v1/keypairadmin/keypair` REST API. For example, you can upload the public and private key to the server and create an alias for that key pair called `KeyPair1` as shown:

```

curl --location --request POST 'https://<admin-host>:<admin-
port>/oam/services/rest/ssa/api/v1/keypairadmin/keypair' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--data-raw '[

```

```

{
  "aliasName": "KeyPair1",

  "publicKey": "MIERDCCAYygAwIBAgIJAJ2KzwSAbV8GMA0GCSqGSIb3DQEBBQUAMIGjMQswC
  QYDVQQGEwJERTEQMA4GA1UECBMHQmF2YXJpYTEPMA0GA1UEBxMGTXVuaWNOMRgwFgYDVQQKEw9
  NSVQteHBlcnRzIEdtYkxgxFjAUBGNVBAsTDUhCQ1RWLURFTU8tQ0ExGzAZBgNVBAMTEml0di5ta
  XQteHBlcnRzLmNvbTEiMCAGCSqGSIb3DQEJARYTAw5mb0BtaXQteHBlcnRzLmNvbTAEFw0xNzE
  wMjIxMTA4NDJaFw0yMjEwMjExMTA4NDJaMIGhMQswCQYDVQQGEwJERTEQMA4GA1UECAWHQmF2Y
  XJpYTEPMA0GA1UEBwwGTXVuaWNOMRgwFgYDVQQKDA9NSVQteHBlcnRzIEdtYkxgEDA0BgNVBAs
  MB1RFU1QgQ0ExHzAdBgNVBAMMFmRlc3Rib3gubW10LXhwZXJ0cy5jb20xIjAgBgkqhkiG9w0BC
  QEWE21uZm9AbW10LXhwZXJ0cy5jb20wggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC
  4GnIwD1Mxts1ZDct6JPTV0mvFn+ZrwqE/
  4WFNAaqtRaChaP21NQ1H55NFNYo1Dl2AhDDNK1Muk+rq6LZOWm8XiuMBA/
  fs3uBNEloa9WYoAEB3ozS14AG+dlyq41diN14F2ys1fl0s4gW/
  H27UH12G1Bgb9Zx1yFZHYItjGKpT15I8fo/
  MQtWFvqoK9rY0UxYHps6Tnfc7ArrQMNsoFu4015N8JuDDtizNxsq8sOK2MgQZNeuOg+ST+8jrJ
  R8CbXRujejfhZM2QMfBeAcJFyxQGBn4UZkys46Y51XJCx7n6Zixlp+y9qNrJjEdup9O9q9VIjS
  86K1wPz62JqaV16R7AgMBAAGjezB5MAkGALUdEwQCAAwLAYJYIZIAyB4QgENBB8WHU9wZW5TU
  0wgR2VuZXJhdGVkIENlcnRpZmljYXR1MB0GALUdDgQWBBRCN+ztRIOc3iF130yCGbkBAOK+oTA
  fBgNVHSMEGDAWgBTEbk/zORG4/M/
  5q7Z9Er8rKAgPcjANBgkqhkiG9w0BAQUFAAOCAQEAEUHPBfoLvB/
  krCTgBZhsxbLutQCB8hDG3rPspJsD3TUHhSU1WxHulxSBMfNKMv11KA1SX/
  4+2epkpz825e047u0SLsmS1XdzfsOt0GLqbB9IQOnTxu2/3z/
  gtNaHULDo2JQf24CfGAFQVv+2Fp8+3B5DvKvuHIEGR6A27Ua/
  wBR78Of1jdwvMBEgDeN5+R5r0HjCt/
  A5ODdG9j8p+wpYQaMWZ6A2iOMWEdgVQnQeW2B5gD9mOV1gCh4HCYMXf/
  apuWZEoafm6qd1Q+SeB4D/
  LbsnDQN1uzqrTj8Jg7C1h55KNMeYECYWBKiqeztxBdLGEjkqCZarnUNYefzXDPbpw==",

  "privateKey": "MIIFDjBABGkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQ0wDgQIjJCdZa17o0YC
  AggAMBQGCCqGSIb3DQMHBAhwwPasgNmZVASCBMjTi8XLv8f70s7vuJE8n4WmQCSp7Q6LwZ05cd
  vBtVQDbKzT1ZbRudIz54jDLQDkaXT8zPBIPdnF4frjKqIzrK1EY0+3oTILGeWabxvbsDKdbyE
  S98euhunNzcBnvm+JYmAm6IcMv11kPu57as8uUUDMoFpqsy7DZ7QhE9BxKdKSAkyPXkefuB5Up
  YnTjzZgES9BAYNedhjPKDXX2et1PwnFpbTndbT6Ur5SbnzMpZYPUg+G9+sedCJMpspd8yS178h
  f0UfftAhKLHpIjzQGYWh+c3BvMzd3f7xIK1/iTme5TJV3SOQNPEQ7E1tyVtwip/
  LvXEKEUvw0yOsFBqS+teHMT9wLnCn5oAXyufI7RcH0x3Y57AJu4xthoL71wSoiYB+WrzD5AmQm
  MfAli01AC4LA8TLGID0GvWm58AD8p+9QJwAbi4rag0wwuBDObt3FOqc0donyo4/
  NS2S8WZmr93A7WwB9KzD1SrYviPpIqPNSsExost7dgcW0LzmKcE1zqzkPb1C2nzQu+Y8vmlNyD
  UW3jP0Ao0fudJZJ1+BEiNc1wMnBQXNke42u7rDF8/
  Jo2r8tsOKa0ErHoRjNLpyzsfjLp+C6wwBcnCJRxsXwvf+CPeN09cMkr3Jt2qzwZJ22NvzeTgK+
  2vA0dCH4rFgylHKvag2FhGYw50/
  8JWMN5mKLUB03GmMz3PjRG7DT5acEn01YV3AAoeoSD84RNRS8oxH7A0hSwvMi8Xu/
  SqkuYfRy2hr0xEmE5M790aPiGkDUTY1A0pPyr02fvQFf5P3VQ1kDT3+W402dD17yVGmvgWSz+
  Abm8cGS1eyTq/O43XEPD55LH6o8gnFH3Ba6/
  sI6vRTDkn0mqVuxTJgpnPmpYsItcLN10660FgzdzvBdxREP5qjy+meqdx+j6rm42CTnDDuWsyY
  KvFzK89LD4o0QjHdt+t9sMhS5PYcj5QTAqKfOVW1AFru41th7nSnRUzonBKAD40qsgnUHOwDRS
  nMRqmoXbTMKt6mx6xgPPeK266GSqMcrcXcfdzRwU3Ad1Xw6fAcFXnqbi9CQuQ8ADSg61W3BZqZy
  M0I+jr9vib1tdTrTuMkDtDglgDcVzcgaRLJ7GJF2Az5mfyGs61uDBhycxRgAhA07ehu7cEU708
  y5UYjaTizWtGmpnAt+bS7KNopqEhzP6OpP8FMKfvqvgwMwpm+
  +AyEMbAswhjaz9LjIOHtTuQTK1BLBHzm1TVIOwn+5bd4W917uVfXCJdS80aEwkFzGBXhJvkwBG
  lMvYO+gr3Kbio202QfjfxN1gob8EFDMPH3N8gkpdSulLzYEzK+21Nz3h1bm07evz2w2xTy9PYP
  1UD1UCugFA18RI+4YdnaIBSyeReODRhaE4AF+T2oAu3rE9c9iCauC9QOSICSULyGbLny54R9tk
  1wdAJI4xASPHoizz8GqqF8s6Y2/
  F46LS+43042rkfQmvdR5yrjYsF7YphSQzp7MREq+QwgXjC8kc5LMn81o2Ii6DcsyEQVBimfu05
  5FFE1IHnUYKKBrxJU77+jpgb/

```

```
pXlqvsm7ucBVCpRD11EvRkSji8qPk60aFotPfNnxOvIWLOjHkaYzfASbtZ/  
G8H2nzMZWTdtxY+tgS8R6Bo+sJZH/NnE="
```

```
  },  
  {  
    "aliasName": "KeyPair2",  
    "publicKey": "MIIeQDCCApCgAwIBAgIUk5Ns4y2CzosB/  
ZoFlaxjZqoBTIIwDQYJKoZIhvcNAQELBQAwfjELMAkGA1UEBhMCVVMxZzY2ODxkZANBgNVBAoMBkZFNNTT  
kNhbG1mb3JuaWEwFjAUBGNVBAcMDVNBhbiBGcmFuY21zY28xDzANBgNVBAoMBkZFNNTT  
DEXMC8GA1UEAwwoQmFku1NMIENsaWVudCBSb290IENlcnRpZmljYXR1IEF1dGhvcml0e  
TAEFw0xOTEwMjcwMDE5NTdaFw0yMTEwMjcwMDE5NTdaMG8xCzAJBgNVBAYTA1VTRMWE  
QYDVQQIDApDYWxpZm9ybmlhMRwwFAYDVQQHDA1TYW4gRnJhbmlhbnQ2VydG1maWNhdGUwGgEiMA0GC  
AZCQYWRU0wxiIjAgBgNVBAMMGUJhZFNNTTCDDBG1lbnQgQ2VydG1maWNhdGUwGgEiMA0GC  
SgQSIB3DQEBAQUAA4IBDwAwggEKAoIBAQDHN18R6x5Oz+u6SOXLoxIscz5GHR6cDcCLg  
yPaX2XfXHdJs+h6fTy61WGM+aXehR2SIwbj5997s34m0MsbvkJrFmn0LHK1fuTLCihEE  
mxGdCGZA9xrxwFYAkEjP7D8v7cAWRMipYF/  
JP7VU7xNUo+QSkZ0sOi9k6bNkABKL3+yP6PqAzsBoKIN51N/  
YRLrppsDmk6nrRDo4R3CD+8JQ19quEoOmL22Pc/  
qpOjL1jgOIFSE5y3gwbzD1fCYoAL5V+by1vu0yJShTTK8oo5wvphcFfEHaQ9w5jFg2ht  
dq99UER3BKuNDuL+zejqGQZCWb0Xsk8S5WBuX813Brrg5giqNagMBAAGjLTArMAkGA1U  
dEwQCMAAAwEQYJYIZIAyb4QgEBBAQDAgeAMAsGA1UdDwQEAwIF4DANBgkqhkiG9w0BAQs  
FAAOCAGEAZBauLzFSOijkDadcippr9C61aHebb0oRS54xAV70E9k5Gxfr/  
E2EMuQ8X+miRUMXxKquffcdsSxzo2ac0flw94hDx3B6vJIYvsQx9Lzo95Im0DdTDkHFX  
hTlv2kjQwFVnEsWYwyGpHMTjanvNk07sBP9p1bN1qTE3QAeyMZNKwJk5xP1U298ERar6  
t13Z2C18m06yLhrq4ba6iPGw08SENxzuAJW+n8r0rq7EU+bMg5spgT1CxExzG8Bb0f98  
ZXMk1pYFogkcuH4OUOFyRodotrotm3iRbuvZNk0Zz7N5n1oLTP1bGPMwBcqaGXvK62N1  
aRkwjnbkPM4MYvREM0bbAgZD2GHyanBTso8bdWvhLvmoSjsFSqJUJp17AZ0x/  
ELWZd69v2zKW9UdPmw0evyVR19elh/  
7dmtF6wbewc4N4jxQnTqIItuhIWKWB9edgJz65uZ9ubQWjXoa+9CuWcV/  
1KxuCkBLhdZxiboLrKm4S1WmMYWd0sJm95H9mJzcLyhLF7ix2kK6K9ug1y02YCVXBC9W  
GZc2x6GMS71DkXSkJFy3EWhCmfXkmFGwOgWkt3Jd1pF9ftcSEMhu4WcMgxi9vZr90dkJ  
LxmK033sVKI/  
hnkPaHwg0Y2YBH5v0xmi8sYU7weOcwynkjZARpUltBUQ0pWCF5uJsEB8uE8PPDD3c4="
```

```
  ,  
  
  "privateKey": "MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIFGJVk+  
2yRcEAggAMBQGCCqGSIb3DQMHBAgI8pa9pbh96ASCBMgOltVUbtKUpcupvL2Eh2Cy5Y  
IO9cCQcgfA9xTBUnhCivgyPdaQJ6iM0ryIdVdvHLn0gySpqo3TSUZheygU7cBQvILdtt  
6Of02yf3Gp7Xs8CAs3qNODUTK51QGwDSJ4zyFquiEUNgSeG6UMji/  
9Y09791ONVYmvz7ZqPwYOK0HwSOF9ttXzAVA9GKMpnCy97G0ezkzPhInBtf/  
nMYuRbWwCddN/zt0IX7Yo6AnF91QzluGUXGnEnJufvKj7q3DSTVolWyQgHlCrq/  
0BMextYxe9GSjHF2UTCUDbmi6Vv1Z3ezkkKuQMPXCZ/  
qOgQFyx4PWmtXKbj3EPZYTI0dTebhCmXv614UIglFwyPQPLVJR0G1ZIRAw4pM01BBMS1  
wLrjMBcW7EUUJ7DyvBX9bJe7vZ0nMS1ksQyWU031RRbK1Yt16QtdV+sw9ZCCB8ZEQ+4y  
06dJvpapmGzH5QjSZzgh9Sp+wXoqiOvvjdKcxzPg3WjclSj/  
MtetD1BDDq9K1hEuc5eNL8Qr/yiX8C/  
9Uk8JbcUaJMeq4H2nZiOgY+DzpMuEEJJo81WztrrFu8KB2oeky5EZ3akB00Ky0bL8MSc  
LkVnbJFzv290P/Jj/  
y6JTo2VzoF+bu519mySjDMxci218+NC7oOwiIzNJPR3IWoLmoRYoXUAYl1tTaZWAmHai  
6iypXsesk1Ezvp4STMK7oqbkf2DXNY+F5wDP3HdD1Yb6FrziKbrQ67+qY+XSXIV/  
Abm2c5ELnHsIFkGV/xDdWkoh+C5Lg8Xssnqh0tcuCuYgqZKDHBVoa/  
xvpq5rCd2i2MwrImC0lvYoksDfsluzAXSXpmtHwiPir6iDb5srlmaJCEKvrRdNxnK28  
kFzCshNtEivEiu9DH+5cMdbAilBYEFbk0N/urNzG4doieCELYMD/  
yZE51oq7S7JVq61Ff706KT1SZQVusumCeyE28H1ocyWuNMmc9jJ1YJ68dBH/  
9YPeGneBu8qCgXdFgsBEdcAD2p5FNzMyZuWRwENne7Mnt8EgUBLpc+kPZ071RNYF8GFx  
3A+U2etNPjGA6a64jGhcM5KcZreJ8wffprFpAuY4rFPJkQ5xyax1RrrlhSorAj6wtiN
```



```
ZVWcuA20URdilhOpHzbq+J9zrtnNCLMkIVCB8YFtjt5puciltD3cD8eg5DPHqntLVdJiUfyZsA
pC8RWamwzfCsN9MRo70Qx14pC4qKYmBDYdyGeQpfi7C69+7153AKqr2ZjhCdaJ3/
g1EAh3PJOCKqetrPqkNfZnEMJz+9nhmgcIn1013t1DFbB1Bishop6GHx+d6PT7AZRei704JsAn
yCOD1xyWd8U1WlXgbr0zQo14c5KiByho7mZlajVWzqKapz40BnTFKmwW9KofSXWp0kKGcxAGPC
+FL8EGXErVSlmbVf8MKfs4eLnKidzL1MKW1pS4n9cwKHmZjPVv0YYUGeW8FAuH0DjQBvnLrCb
WLY3DXHR/GAf9hagSa0G0F07MvQcSWE5eqXwICJ9PMhxiEEEx9PF0QJuMxhs6bni/
thZqFOYDN0kbmSNntB0kG4EwWCxKoegEUMX7Ug5R0IGilLJ37Q9Nqvh21msH3mi8Kcfu01kees
dZJPbFFfkCVHCPMGx6QRY8="
    }
  ]'
```

You can also upload the key pair using the p12 file and create an alias `KeyPair1` for that key pair as shown in the following example:

 **Note:**

The key size must be at least 2048 bits and the p12 file must contain only one public-private key pair.

```
curl --location --request POST 'https://<admin-host>:<admin-port>/oam/
services/rest/ssa/api/v1/keypairadmin/keypair?
password=<your_password>&aliasName=KeyPair1' \
--header 'Content-Type: application/x-pkcs12' \
--data-binary '@/u01/keypairfiles/p12files/your_file-client.p12'
```

, where `password` is the password of the p12 file.

For more information, see [Add a New KeyPair](#) REST API documentation.

- Associate the key pair with the required OAuth Identity Domain using `defaultSigningKeyPair` and `keyPairRolloverDurationInHours`.

`defaultSigningKeyPair`

Specify the name of the keypair alias to be used for signing tokens from this domain.

 **Note:**

The key pair alias must be created in the system before using it here. See the previous step for details.

`keyPairRolloverDurationInHours`

Specify the number of hours for the previous key pair to remain active in the domain, after the new key pair has been associated with the domain. During this time, tokens signed with the previous key pair are validated (unless tokens have expired), and the public key of

the previous key pair will be part of the JSON web keyset of the domain.

Values supported: 0 to 99999 hours.

Default is 48 hours.

See REST API documentation: [Delete a KeyPair Based on the Alias Name](#)

For example,

```
curl --location -g --request PUT 'https://<admin-host>:<admin-  
port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/  
oauthidentitydomain?name=Domain1' \  
--header 'Authorization: Basic dGVzdDp0ZXN0=' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
    "defaultSigningKeyPair": "KeyPair1",  
    "keyPairRolloverDurationInHours": "24"  
}'
```

For more information, see [Identity Domain REST Endpoints](#) documentation.



Note:

To perform signature verification of the tokens, the trust certificate can be retrieved using the following `jwtks_uri`: `http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/security`. The `jwtks_uri` can be retrieved using OpenIDConnect Discovery Endpoint. For details, see [Configuring OpenIDConnect Discovery Endpoint](#)

37.2.2 Enabling Consent Management

You can enable Consent Management for each of the OAuth Identity Domains or all the OAuth Identity Domains in OAM.

During the 3-legged OAuth flow, OAM presents a consent page enabling you to grant access to the resource. If Consent Management is enabled, your consent is saved, and OAM skips the consent on subsequent 3-legged OAuth flows. For details about the 3-legged OAuth Flow, see [Understanding 3-Legged Authorization](#)

By default, Consent Management is disabled.

To enable Consent Management per OAuth Identity Domain, create or modify the OAuth Identity Domain by setting the custom attribute `consentExpiryTimeInMinutes` using the Admin Server OAuth API.

For example:

```
curl --header 'Authorization: Basic d2VibG9naWM6d2VsY29tZTE=' \  
--header 'Content-Type: application/json' \  
--request POST 'http:<Servername>:<Port>/oam/services/rest/ssa/api/v1/
```

```

oauthpolicyadmin/oauthidentitydomain' \
--data-raw
'{"name":"MyDomain","identityProvider":"UserIdentityStore1","description":"My
Domain",
"tokenSettings":
[{"tokenType":"ACCESS_TOKEN","tokenExpiry":3600,"lifeCycleEnabled":false,"ref
reshTokenEnabled":false,"refreshTokenExpiry":86400,"refreshTokenLifeCycleEnab
led":false}],
"errorPageURL":"/oam/pages/error.jsp","consentPageURL":"/oam/pages/
consent.jsp",
"customAttrs":{"\"domainCertValidityInDays\": \"30\",
\"consentExpiryTimeInMinutes\": \"10\"}}
'

```

where, `consentExpiryTimeInMinutes` in minutes is the duration, during which the OAuth consent stays valid.

Beyond this duration, OAM presents the consent page again during the 3-legged OAuth flow, and you must grant the consent.

OAM allows each of the OAuth Identity Domains to enable, disable, or change the consent validity period using `consentExpiryTimeInMinutes`.

To enable Consent Management for all the OAuth Identity Domains, add `consentExpiryTimeInMinutes` system property while starting OAM.

1. Stop all the Administration and Managed Servers.
2. Edit the `$OAM_DOMAIN_HOME/bin/setDomainEnv.sh`, and add the `consentExpiryTimeInMinutes` property under `EXTRA_JAVA_PROPERTIES` as shown

```
EXTRA_JAVA_PROPERTIES="-DconsentExpiryTimeInMinutes=10"
```

3. Start the Administration and Managed Servers.



Note:

System property overrides the individual Consent Management configuration on each OAuth Identity Domains.

See Also, REST API for OAuth in Oracle Access Manager.

37.2.3 Creating a Resource

A Resource Server hosts protected resources. The resource server is capable of accepting and responding to protected resource requests using access tokens.

The important parameters used in the `curl` command to create a resource are:

- **Name:** Name of the Resource Server
- **Scopes:** The following two parameters are used
 - `scopeName` - Name of the scope
 - `description` - Description of the scope

- **idDomain** - Name of the IdentityDomain under which this resource server is created
- **tokenAttributes** - List of custom attributes that are sent by the server, as part of the access token. The attributes can be "STATIC" in which case the value is substituted as is. If "DYNAMIC", the attributeValue is evaluated and populated in the final AccessToken.

 **Note:**

Scopes are referred to by prefixing the resource server name. This makes them unique across resource servers.

Endpoint for CRUD operations:

`http:<AdminServerHost:Port>/oam/services/rest/ssa/api/v1/
oauthpolicyadmin/application`

 **Note:**

Use `Content-Type:application/json` in the REST API HTTP request.

- A sample `curl` command to create a resource using scopes is shown below.

```
{ "name": "ResServer1", "description": "TestResourceServer", "scopes":
  [ { "scopeName": "scope1", "description": "ViewPage" },
    { "scopeName": "scope2", "description": "UpdatePage" },
    { "scopeName": "scope3", "description": "ModifyPage" } ], "tokenAttributes"
  :
  [ { "attrName": "sessionId", "attrValue": "$session.id", "attrType": "DYNAMIC" },
    { "attrName": "resSrvAttr", "attrValue": "RESOURCECONST", "attrType": "STATIC" } ], "idDomain": "TestDomain1", "audienceClaim": { "subjects":
    [ "ab0" ] } }
```

For more information on OAuth REST APIs, See REST API for OAuth in Oracle Access Manager.

37.2.4 Creating a Client

A Client is an application making protected resource requests on behalf of the resource owner and with the resource owner's authorization.

The important parameters used in the `curl` command to create a client are:

- **Name:** Name of the client
- **idDomain:** Name of the identityDomain under which the client is created
- **secret:** Client secret incase of a CONFIDENTIAL_CLIENT

- **clientType**: Type of client. Supported values - CONFIDENTIAL_CLIENT, PUBLIC_CLIENT, MOBILE_CLIENT
- **redirectURIs**: List of redirectURIs configured for the client
- **attributes**: List of custom attributes configured for the client
- **grantTypes**: List of allowed grant types. Allowed values - PASSWORD, CLIENT_CREDENTIALS, JWT_BEARER, REFRESH_TOKEN, AUTHORIZATION_CODE
- **Scopes**: List of scopes that the client can request access to.
 - scopeName - Name of the scope. This is referred to by the **<ResourceServerName>.<ScopeName>**
- **defaultScope** - This is the default scope that the access token is generated with, If no scope is specified during the Runtime Flows.

Endpoint for CRUD operations:

`http:<AdminServerHost:Port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/client`



Note:

Use `Content-Type:application/json` in the REST API HTTP request.

- A sample `curl` command to create a client using scopes is shown below.

```
{
  "attributes":
  [{"attrName":"customeAttr1","attrValue":"CustomValue","attrType":"static"}
  ],
  "secret":"welcome1",
  "id":"TestClient",
  "scopes":
  ["ResServer1.scope1"],
  "clientType":"CONFIDENTIAL_CLIENT",
  "idDomain":"TestDomain1",
  "description":"Client Description",
  "name":"TestClient",
  "grantTypes":
  ["PASSWORD","CLIENT_CREDENTIALS","JWT_BEARER","REFRESH_TOKEN","AUTHORIZATION_CODE"],
  "defaultScope":"ResServer1.scope1",
  "redirectURIs":
  [{"url":"http://localhost:8080/Sample.jsp","isHttps":true}]
}
```

For more information on OAuth REST APIs, See REST API for OAuth in Oracle Access Manager.

37.3 Enabling User Lock Validation

You must enable the user lock validation to invalidate the tokens when the user is locked or disabled.

If the user is locked or disabled, the OAuth user validation flow fails. OAuth user validation is performed during OAuth authorization grant, JWT bearer grant, refresh token grant, and access token validation flows.

For example, enabling the user lock validation ensures that the access tokens (using refresh tokens) are not issued for the locked or disabled user.

Prerequisite

Before you proceed, perform the following steps to verify if the `LDAPNoPasswordValidationSchemeOAuth` Authentication Scheme exists:

1. Log in to the Oracle Access Management Console.
2. In the Oracle Access Management Console, on the top-right, click **Application Security**
3. From the Application Security Launch Pad, under Access Manager, click **Authentication Schemes**
4. Specify the name `LDAPNoPasswordValidationSchemeOAuth` and click **Search**.
5. Based on whether the search returns the authentication scheme, following one of the following:
 - If the authentication scheme exists, follow the steps listed in [Enabling User Lock Validation if LDAPNoPasswordValidationSchemeOAuth Exists](#)
 - If the authentication scheme is not found, you must download the latest OAM patch and follow the steps listed in [Enabling User Lock Validation if LDAPNoPasswordValidationSchemeOAuth Does Not Exist](#)

37.3.1 Enabling User Lock Validation if `LDAPNoPasswordValidationSchemeOAuth` Exists

To enable user lock validation perform the following steps:

1. Log in to the Oracle Access Management Console.
2. In the Oracle Access Management Console, on the top-right, click **Configuration**.
3. Click **User Identity Stores**
4. Under **OAM ID Stores**, select your user identity store and click **Edit**.
5. Check the box beside **Use Native ID Store Settings**.
6. Under Password Management, check the box beside **Enable Password Management**.

For more information about these parameters, see [User Identity Store Settings](#)

37.3.2 Enabling User Lock Validation if `LDAPNoPasswordValidationSchemeOAuth` Does Not Exist

To enable user lock validation perform the following steps:

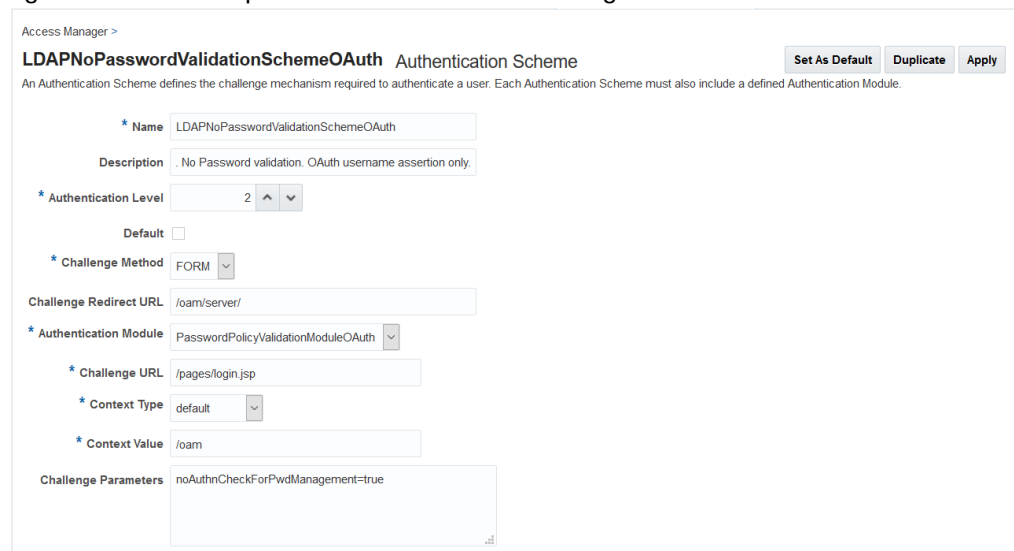
1. Follow the steps provided in [Enabling User Lock Validation if LDAPNoPasswordValidationSchemeOAuth Exists](#)
2. Create a new Authentication Scheme (for example, `LDAPNoPasswordValidationSchemeOAuth`. The name can be any string):
 - a. In the Oracle Access Management Console, on the top-right, click **Application Security**

- b. From the Application Security Launch Pad, under Access Manager, click **Authentication Schemes**
- c. From the drop-down for Authentication Module, select PasswordPolicyValidationModuleOAuth

 **Note:**

PasswordPolicyValidationModuleOAuth is available out-of-the-box with OAM installation.

- d. Set the challenge parameter `noAuthnCheckForPwdManagement=true` in the authentication scheme.
- e. Set the other parameters in the authentication scheme. For example, the following figure shows a Sample Authentication Scheme Page:



Access Manager >

LDAPNoPasswordValidationSchemeOAuth Authentication Scheme Set As Default Duplicate Apply

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme must also include a defined Authentication Module.

* Name: LDAPNoPasswordValidationSchemeOAuth

Description: No Password validation. OAuth username assertion only.

* Authentication Level: 2

Default:

* Challenge Method: FORM

Challenge Redirect URL: /oam/server/

* Authentication Module: PasswordPolicyValidationModuleOAuth

* Challenge URL: /pages/login.jsp

* Context Type: default

* Context Value: /oam

Challenge Parameters: noAuthnCheckForPwdManagement=true

3. Update the Authentication Policy with the new scheme created:
 - a. From the Application Security Launch Pad, under Access Manager, click **Application Domains**
 - b. Click **Search**, and under the Search Results, click **IAM Suite**
 - c. In IAM Suite Application Domain, click the **Authentication Policies** tab and click **OAuth Assertion Policy**
 - d. From the drop-down for Authentication Scheme, select the new scheme that you have created. For example, `LDAPNoPasswordValidationSchemeOAuth`
 - e. Click **Apply**.

37.4 Enabling Consent Management on MDC

Follow the steps in this section to enable Consent Management on MDC.



Note:

To enable consent management for a pre-existing MDC configuration, you must re-configure MDC.

1. Setup two OAM environments DC1 (Master) and DC2 (Clone).
2. Set the `consentExpiryTimeInMinutes` parameter in the OAuth Identity Domain on the Master. For details, see [Enabling Consent Management](#).
3. Configure OAuth on the Multi-Data Centers. For details, see [Configuring OAuth in Multi-Data Centers](#)
4. Enable Automated Policy Synchronization. For details, see [Enabling Automated Policy Synchronization](#)

37.5 Configuring OAuth in Multi-Data Centers

You can configure OAuth support in Multi-Data Centers(MDC) using REST APIs.

The following scenario illustrates the flow to configure OAuth in an MDC environment using REST APIs. Perform the following procedures in the sequence given here:.

1. Create the OAuth Artifacts - Identity Domain, Resource Server, Client and associated trust artifacts on the MasterDC.
2. Follow the steps given in [Configuring Multi-Data Centers](#) to setup MDC between two data centers.



Note:

As part of Step 2, the requests **exportAccessStore** on Master and **importAccessStore** on Clone DC are performed. This ensures that artifacts created on MasterDC are visible on CloneDC. Step 2, also ensures that the OAuth Artifacts get copied over to the Clone DC.

3. Perform `GET` commands of these artifacts on the Clone DC to confirm that OAuth has been successfully setup in MDC mode.
4. Enable Automated Policy Synchronization
5. Now execute the 2 legged flows to verify MDC flows.
 - a. Create an Access Token as part of Password Grant Flow on DC1.
 - b. Send the same token to the Clone DC end point for validation.
 - c. The token should be valid on DC2.

37.6 Optional Parameters for Consent Management in Multi-Data Centers

You can set the following optional parameters in the system property of the Runtime Server on the Clone Data Center.

1. Stop all the Administration and Managed Servers.
2. Edit the `$OAM_DOMAIN_HOME/bin/setDomainEnv.sh` and add the parameters under `EXTRA_JAVA_PROPERTIES`.
For example,

```
EXTRA_JAVA_PROPERTIES="-DconsentExpiryTimeInMinutes=10"
```

3. Start the Administration and Managed Servers.

Table 37-1 Optional Parameters for Consent Management on MDC

Parameter	Default Value	Description
<code>failOnConsentStoreError</code>	<code>true</code>	By default, if master DC is not available, the user consent request is not processed and an error is displayed to the user. To turn off the error, set the value of the parameter to <code>false</code> . However, the error will exist in the logs.
<code>printEntitiesInRequest</code>	<code>false</code>	Prints the body of HTTP request and response to the log.
<code>runtimeResourceConnTimeout</code>	60000 (in milliseconds; 60 seconds)	Specifies the connection timeout for the HTTP Request to store/fetch/delete entities to the Master DC.
<code>runtimeResourceReadTimeout</code>	60000 (in milliseconds; 60 seconds)	Specifies the HTTP response read timeout from the Master DC.
<code>sourceDcJournalThreshold</code>	100	Defines the limit for on-demand replication from the runtime server. After this limit, the on-demand replication sets in and all the user requests also try to fetch the results from Master.
<code>sourceDcJournalThresholdUpperLimit</code>	250	Defines the upper limit for on-demand replication. If the Journal Sequence difference between the Clone DC and Master DC grows beyond the defined limit, the OAMRE-07023 error is displayed to the administrators for all user requests.

Table 37-1 (Cont.) Optional Parameters for Consent Management on MDC

Parameter	Default Value	Description
sourceDcLastPingThreshold InSec	600 (in seconds; 10 minutes)	<p>Defines the limit for availability of the Master.</p> <p>If the last ping to the Master is successful within this time limit, the master is considered available.</p> <p>You must always set the value of this parameter greater than the value of <code>sourceHealthCheckInterval</code> InSec</p>
sourceHealthCheckInterval InSec	120 (in seconds; two minutes)	<p>Specifies the polling frequency for checking the connection and the latest journal sequence in Master DC.</p>
replication.poller.thread .interval	120 (in seconds; two minutes)	<p>Running Replication on Policy Manager can result into multiple instances due to Policy Manager cluster. Replication Instance Manager Thread ensures that a single instance of Replication is running in a clustered environment.</p> <p>This parameter defines the interval, in which Replication Instance Manager Thread must run.</p>
replication.database.thre ad.interval	120 (in seconds; two minutes)	<p>Single replication instance on Policy Manager Cluster is maintained by having an entry in database and continuously updating it. Every cluster keeps performing the insert or update operation.</p> <p>This parameter determines a valid database entry duration, that is, if an instance has updated the database entry within the duration, for example, 2 min, then the other instances cannot override it.</p> <p>But if the instance fails to update the entry, other instances can update the database entry post the duration.</p>

37.7 Error Codes and Troubleshooting Steps for Consent Management on MDC

The section lists the error codes and the troubleshooting steps for consent management on MDC.

Error Code	Error Description	Resolution
OAMRE-07001	Error is logged in the Master DC if the user data has been modified but the deleted data is still present in the main store.	Manually delete the runtime data identified by the unique id present in the log.
OAMRE-07002 , OAMRE-07018, OAMRE-07019	Error is logged in the Clone DC when there are multiple replication agreements pointing to the same Master DC.	There must be only one replication agreement from Master DC (Source DC) and Clone DC(target DC). Remove the multiple replication agreements using the replication agreement Rest APIs.
OAMRE-07004	Error occurs if the replication agreement has been modified and deleted, and the older replication agreement is no longer valid.	Restart the managed servers in Clone DC.
OAMRE-07006	Error occurs if the Master DC replication API does not work or the current replication agreement is not valid.	Restart the managed servers in Clone DC.
OAMRE-07008	Error is logged if MDC has been setup without the Policy Manager REST endpoint.	Add the Policy Manager REST endpoint, in the MDC configuration, with the name <code>PolicyManagerRESTEndpoint</code>
OAMRE-07009	Error is logged if the Replication Agreement is not present and/or the replication agreement does not have the authorization code using which the master can be connected.	Ensure the replication agreement exists with necessary authorization code.
OAMRE-07010 , OAMRE-07011, OAMRE-07012, OAMRE-07013 , OAMRE-07016, OAMRE-07020, OAMRE-07022	Error is logged in the Clone DC when the Master DC resources are not present and/or Master DC is not available.	Check if the Master DC resource is available and if the Master DC has been patched correctly.
OAMRE-07014, OAMRE-07015, OAMRE-07016	Error occurs if there is missing entity in the Master DC.	Delete the entity from the clone DC.
OAMRE-07017	Error occurs if the source server (Master DC) setting does not exist in the configuration.	Check the Entity replication or Import and Export the access store again and redo the Entity and Runtime Entity Replication.

Error Code	Error Description	Resolution
OAMRE-07023	Error occurs if the runtime entity replication is out-of-sync and the journal backlog for replication has grown beyond the value of <code>sourceDcJournalThresholdUpperLimit</code> parameter.	This gets auto corrected once the runtime entity replication gets in sync. Manually Import access store and Export access store to synchronize the data again.
Error in the clone processing server response.	Enable the HTTP request and response log by setting the system property <code>printEntitiesInRequest=true</code>	Request and response Logging is enabled by setting this property.

37.8 Dynamic Client Registration

Dynamic client Registration (DCR) provides a way for the native mobile apps (Android) to dynamically register as clients with the OAuth Server (OAM).

This section provides the detailed process for dynamically registering clients with the OAuth Server (OAM).

1. [Enabling Dynamic Client Registration](#)
2. [Creating OAuth Client Template](#)
3. [Getting Registration Tokens](#)
4. [Registering the Client using the Registration Token](#)
5. [Reading Client Details](#)

37.8.1 Enabling Dynamic Client Registration

You must enable Dynamic Client Registration (DCR) at the identity domain level.

Create an Identity domain with the custom attribute `isDcrRegEnabled` set to `true` using the Admin Server OAuth API, as shown.



Note:

If the `isDcrRegEnabled` flag is not specified, or set to `false` then DCR is disabled.

```
curl -X POST \
  http(s)://<Admin-Server-URL>:<Admin_Server_Port>/oam/services/
  rest/ssa/api/v1/oauthpolicyadmin/oauthidentitydomain \
  -H 'authorization: Basic d2VibG9naWM6d2VsY29tZTE=' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d
  '{"name":"dcr_domain","enableMultipleResourceServer":false,"description
  ":"DCR Domain",
  "tokenSettings":
```

```
[{"refreshTokenEnabled":true,"refreshTokenLifeCycleEnabled":true,"refreshTokenExpiry":5400,
"lifeCycleEnabled":true,"tokenType":"ACCESS_TOKEN","tokenExpiry":1800},
{"refreshTokenEnabled":true,"refreshTokenLifeCycleEnabled":true,"refreshTokenExpiry":10800,
"lifeCycleEnabled":true,"tokenType":"AUTHZ_CODE","tokenExpiry":240}],
"customAttrs":{"isDcrRegEnabled":"true"}"
}]'
```

37.8.2 Creating OAuth Client Template

The OAuth client template serves as a blueprint for creating the actual clients.

Create the OAuth client template using the Admin Server OAuth API, as shown.

```
curl -X POST \
http(s)://<AdminServerHost:Port>/oam/services/rest/ssa/api/v1/
oauthpolicyadmin/client \
  -H 'authorization: Basic d2VibG9naWM6d2VsY29tZTE=' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{"id":"DCR_REG_STUB_oma", "secret":"welcome1",
"redirectURIs": [{"url":"http://www.dcr.com/access","isHttps":"false"}],
"scopes":["dcrreg"],"grantTypes":
["IMPLICIT"],"clientType":"PUBLIC_CLIENT","idDomain":"dcr_domain",
"description":"dcr client for acme app
registration","name":"DCR_REG_STUB_acme","defaultScope":"dcrreg"}'
```

Table 37-2 Mandatory Property and Values for Creating the OAuth Client Template

Property	Values
grantTypes	IMPLICIT.
clientType	PUBLIC_CLIENT
idDomain	Must be the same as the domain under which the actual clients need to be created.
name	Must be prefixed with DCR_REG_STUB_.




Note:


The prefix must not be used in regular client names.

Table 37-2 (Cont.) Mandatory Property and Values for Creating the OAuth Client Template

Property	Values
defaultScope	dcrreg

 **Note:**
The scope field must contain only one scope and its value must be `dcrreg`.

redirectURIs	Specify the URIs as required by the actual client.
--------------	--

 **Note:**
The redirectURI values are automatically assigned to the actual client.

37.8.3 Getting Registration Tokens

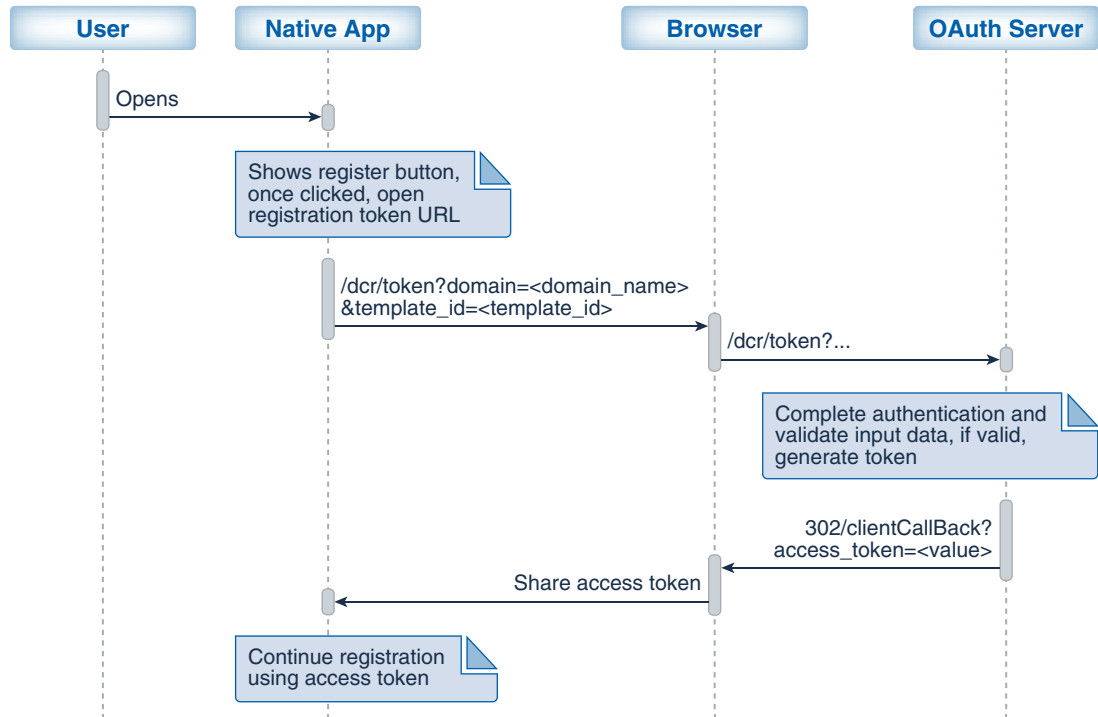
The mobile device apps that need to register dynamically with the OAuth Server (OAM) must first acquire the registration token.

Dynamic Client Registration (DCR) must be enabled and the OAuth Client Template must be created.

Process Flow for Getting Registration Token

1. User opens native app on the mobile device
2. The native app displays Register button
3. When the user clicks on the Register button, the app launches the browser with registration token URI. The request to the OAuth Server includes `domain_name` and `template_id`.
4. Based on the authentication policy configured, the user is redirected to the login form.
5. After successful authentication, OAuth Server generates the registration token and returns it as a token or qrcode, based on the request.
6. Using the registration token, the native app can continue to register the client.

Figure 37-1 Diagram Showing the Flow for Getting Registration Token



Registration Token Sample Request

On the mobile device, when the user (or resource-owner) opens the app, the app must send a GET request with `/dcr/token` endpoint to the Oauth Server (OAM). The `/dcr/token` endpoint must be protected.

The following is a request sample

```
GET http(s)://<server-host>:<server-port>/oauth2/rest/dcr/token?
domain=dcr_domain&template_id=DCR_REG_STUB_acme&response_type=token
```

domain	Name of the domain under which the client has to be registered.
template_id	The Oauth Client template ID
response_type	Response type can be either of the following: <ul style="list-style-type: none"> token qrcode

Table 37-3 Registration Token Sample Response

Configuration	Sample Response
Redirect URI is defined in the Client Template. In this case <code>response_type</code> parameter if passed is ignored.	<code><redirect_uri from template>? access_token=<registration access token value></code>
Redirect URI is not present and <code>response_type</code> is not passed, or passed as token.	HTTP/1.1 200 OK Content-Type: application/json Cache-Control: no-cache Pragma: no-cache { "access_token": "token value", "token_type": "Bearer", "expires_in": 1800 }
Redirect URI is not present and <code>response_type</code> is passed as <code>qrcode</code>	HTTP/1.1 200 OK Content-Type: image/png Cache-Control: no-cache Pragma: no-cache <QR Code Image in png format>

The following table shows the responses in case of errors:

Table 37-4 Registration Token Error Responses

Scenario	HTTP Status Code	Error Message	Secondary Message
OAuth Service is not enabled	403	Unauthorized	Unauthorized
Dynamic Client Registration is not enabled	403	Unauthorized	Unauthorized
Required fields are not provided	400	Invalid Request	Required fields are missing
Invalid Domain	400	Invalid Request	Invalid Domain
Invalid Client: <ul style="list-style-type: none"> Name does not have the prefix <code>DCR_REG_STUB_</code> Client passed doesn't exist 	400	Invalid Request	Invalid Client
Invalid Response Type	400	Invalid Request	Unsupported Response Type

Table 37-4 (Cont.) Registration Token Error Responses

Scenario	HTTP Status Code	Error Message	Secondary Message
Client passed is not authorized to execute DCR flow	403	UnAuthaorized Client	Unauthorized Client

37.8.4 Registering the Client using the Registration Token

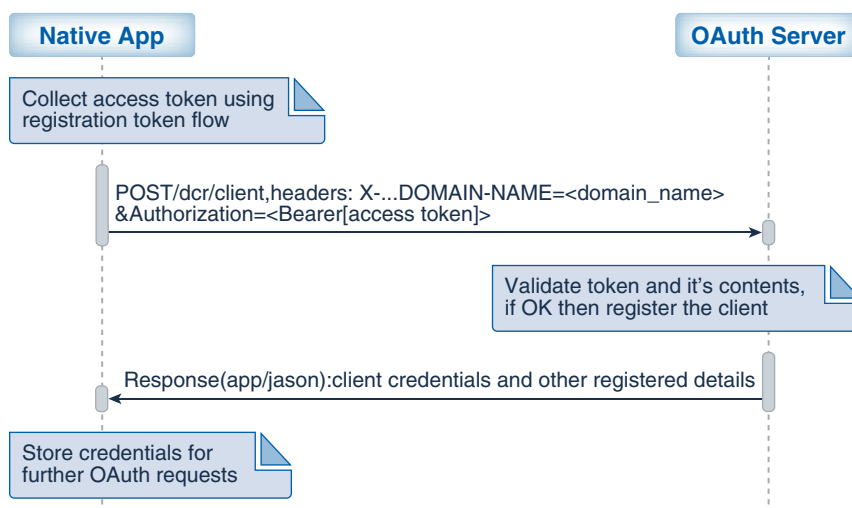
Use the registration token to register the clients with the OAuth Server (OAM).

Process Flow for Client Registration

1. The native app sends a request containing the registration token to OAuth Server for registering the client.
2. OAuth Server validates the token and registers the client. The client profile includes the following fields:

client_id	auto-generated
client_secret	auto-generated
client_name	auto-generated
grant type	Client is registered with grant-type as Authorization_Code and Refresh_token only.
scopes/default scope	dcrread is the only scope added to client profile. This is also the defaultScope.
redirect_uris	Derived from client template.
client type	By default assigned as Mobile/Native Client
token attributes	Derived from client template.

Figure 37-2 Diagram Showing the Flow for Client Registration



Client Registration Sample Request

The application must use the `POST` method with `/dcr/client` to register as client.

```

POST /oauth2/rest/dcr/client HTTP/1.1
Host: <OAuth Server host-name>:<OAuth server port name>
Content-Type: <can be any valid value as there is no input>
X-OAUTH-IDENTITY-DOMAIN-NAME: dcr_domain
Authorization: Bearer eyJraWQiOi.abcdsfdf.ascsfsdff
    
```

<code>X-OAUTH-IDENTITY-DOMAIN-NAME</code>	Name of domain under which the client must be created. This value must be the same as the domain name specified during token acquisition, otherwise the request will fail.
<code>Authorization</code>	Registration token is provided as Bearer token.

Client Registration Sample Response

```

HTTP/1.1 201 Created
Content-Type: application/json
Cache-Control: no-cache
Pragma: no-cache
{
  "client_id": "cd885f3da58f498e830c4f636636dd23",
  "client_secret": "gjDmqUVWlk",
  "client_name": "oma498e830c4f636636dd23",
  "redirect_uris": [
    "app://callBack"
  ],
  "client_secret_expires_at": 0,
  "client_get_uri": "http(s)://<host>:<port>/oauth2/dcr/client?
    
```

```
client_id=<id of created client>"
}
```

The following table shows the responses in case of errors:

Table 37-5 Client Registration Error Responses

Scenario	HTTP Status Code	Error Message	Secondary Message
OAuth Service is not enabled	403	Unauthorized	Unauthorized
Dynamic Client Registration is not enabled	403	Unauthorized	Unauthorized
Required fields are not provided	400	Invalid Request	Required fields are missing
Invalid Domain	400	Invalid Request	Invalid Domain
Invalid Token	401	Unauthorized	Access token is {0} , where {0} could be: <ul style="list-style-type: none"> • expired • malformed • invalid • revoked
Fails on server	500	Internal Server Error	Detailed error message on what went wrong
Client already exists	409	Client already exists	Client already exists

37.8.5 Reading Client Details

Read APIs are protected using OAuth. This section provides details on how to read client information.

Generate the authorization code, for the client that needs to be read, using `scope=dcrread` as shown in the following sample request.

```
http(s)://<server-host>:<server-port>/oauth2/rest/authz?
response_type=code&client_id=cd885f3da58f498e830c4f636636dd23&domain=dcr_domain
&scope=dcrread&state=xyz&redirect_uri=http://www.dcr.com/access
```

Pass this authorization code to the `POST` method with the `/oauth2/rest/token` endpoint to generate the access token with `scope=dcrread`.

Pass this access token as bearer access token using the `GET` method with the `/dcr/client` endpoint to read the client.

Sample Request for Retrieving Client Details

```
GET /oauth2/rest/dcr/client/cd885f3da58f498e830c4f636636dd23 HTTP/1.1
Host: <OAuth Server host-name>:<OAuth server port name>
Content-Type: <can be any valid value as there is no input>
X-OAUTH-IDENTITY-DOMAIN-NAME: dcr_domain
Authorization: Bearer <Access token>
```

Sample Response

```
{
  "client_id": "cd885f3da58f498e830c4f636636dd23",
  "domain": "domainName",
  "client_name": "oma498e830c4f636636dd23",
  "redirect_uris": [
    "http://www.dcr.com/access", "other redirect URI"
  ]
}
```

37.8.6 Deleting Dynamically Registered Client

This section provides details on how to delete the dynamically registered client.

Generate the authorization code, for the client that needs to be deleted, using `scope=dcrdel` as shown in the following sample request.

```
http(s)://<server-host>:<server-port>/oauth2/rest/authz?
response_type=code&client_id=cd885f3da58f498e830c4f636636dd23&domain=dcr_domain
&scope=dcrdel&state=xyz&redirect_uri=http://www.dcr.com/access
```

Pass this authorization code to the `POST` method with the `/oauth2/rest/token` endpoint to generate the access token with `scope=dcrdel`.

Pass this access token as bearer access token using the `DELETE` method with the `/dcr/client` endpoint to delete the client.

```
GET /oauth2/rest/dcr/client/cd885f3da58f498e830c4f636636dd23 HTTP/1.1
Host: <OAuth Server host-name>:<OAuth server port name>
Content-Type: <can be any valid value as there is no input>
X-OAUTH-IDENTITY-DOMAIN-NAME: dcr_domain
Authorization: Bearer <Access token>
```

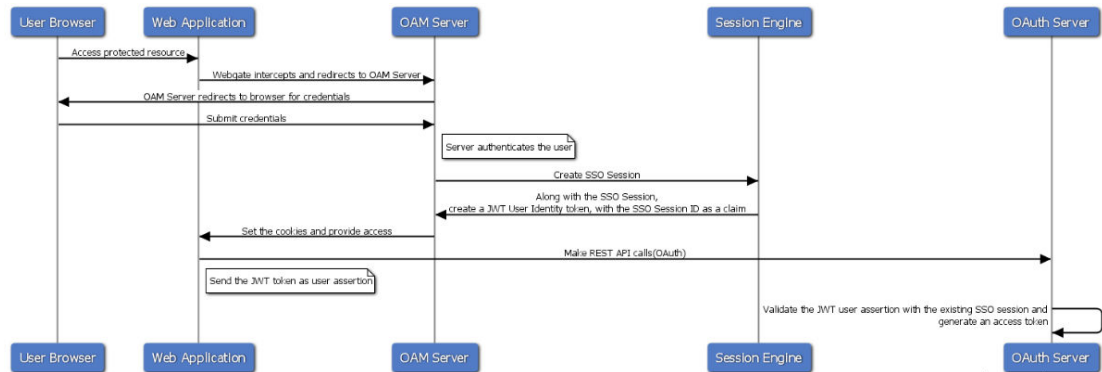
37.9 SSO Session Linking for OAuth Tokens

In deployment scenarios where a few resources are protected by OAM while some might be accessed with OAuth, to achieve seamless SSO between the different mixes of applications, it is necessary to link the SSO session with the Access Token. SSO Session Linking for OAuth Tokens supports key OAuth deployments requiring 2 legged flows involving native mobile apps and Synchronization of OAuth Tokens with SSO tokens.

Use Case Flow

[Figure 37-3](#) illustrates the use case flow of the SSO Ssession linking.

Figure 37-3 Use case flow for SSO Session Linking for OAuth Tokens



Server Changes to Link SSO Session

SSO Linked JWT Token Creation

1. When the SSO Session is created, a JWT User Token is also created. The JWT User Token has the **SSO "session_id"** as part of its claims.
2. Creating this JWT Token is based on configurations. When created, this token can be sent either as a cookie or a header to downstream applications. Currently the configurations are set as challenge parameters at the scheme level.

By default, the SSO link JWT token is set in the cookie.

Note:

If the `OAUTH_TOKEN_RESPONSE_TYPE` is header, the JWT token is set with the cookie name `JWTAssertion`.

If the `OAUTH_TOKEN_RESPONSE_TYPE` is cookie, the JWT token is set with the cookie name `OAUTH_TOKEN`.

Challenge Parameters

```
IS_OAUTH_OAM_SSO_LINK_ENABLED=true
IS_OAUTH_USER_ASSERTION_ENABLED=true
OAUTH_TOKEN_RESPONSE_TYPE=header
```

3. **Token Signing:** On bootstrap a default OAuth key-certificate is boot-strapped into the server. The JWT token will be signed by the Identity Domain private key. When the JWT token is received as an assertion back, the X5T value is retrieved from the header and the associated public key is fetched, which can be used to verify the token.

SSO Linked JWT Token Validation

1. When the token is sent back as part of the JWT Bearer flow in the OAuth Token request, the OAM server retrieves the **SSO "session_id"** from the token.

2. **Check Valid Session:** If the JWT Token has a session ID, the server knows this is a SSO Linked JWT Token. It retrieves the "sessionId" claim from the token and checks if the server session with the given ID is still valid.
3. If the session is valid, the subject from the SSO Session is compared with the "sub" field in the JWT Token. If this matches, the access token for this user is generated and returned to the client.

SSO Linked JWT Token Validation in MDC Flows

1. In case of an MDC Enabled environment, as part of the JWT Token creation, another claim "**mdc_sso_link**" is also added to the token. This claim contains the **clusterId** of the machine on which the session was anchored and the **UserIdentityStore** reference
2. When the token is sent back as part of the JWT Bearer flow in the OAuth Token request, the OAM server will retrieve the SSO Session ID from the token.
3. **Check Valid Session:** If the JWT Token has a session ID, the server knows this is a SSO Linked JWT Token. It retrieves the "sessionId" claim from the token and the clusterid from the mdc_sso_link claim and retrieves the session. The normal MDC flows for checking validity of the session are maintained here.
4. If session is valid, the subject from the SSO Session will be compared with the "sub" field in the JWT Token. If this matches, the access token for this user is generated and the returned to the client.

Session IdleTimeout and the SSO Linked Token

If the session has been idle for more than 15 mins(configured value), when this JWT token is checked for validity it will fail. This ensures that the rules of the session are also applied to the OAuth Access Tokens.

37.10 Runtime REST APIs for OAuth 12c

Runtime REST APIs for OAuth provides REST calls for 2-legged and 3-legged OAuth Services flows in the new 12c OAuth Server. The sections provide sample REST requests that show how to get a resource access token.

The new end point is:

```
http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token
```

Creating Access Tokens

2 - Legged Flows

1. Using Resource Owner Credentials

Following is a sample request against the server:

```
curl -i -H 'Authorization: Basic U1NPTGlua0NsaWVudDp3ZWxjb211MQ==' -H
"Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -H "X-OAUTH-
IDENTITY-DOMAIN-NAME: SSOLink" --request POST http://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token -
d 'grant_type=PASSWORD&username=weblogic&password=welcome1&scope=SSOLink.link
1'
```

 **Note:**

Headers of interest for all requests

- Authorization : Base64 Url encoded ClientID:secret combination.
- X-OAUTH-IDENTITY-DOMAIN-NAME: Identity Domain that the client belongs to.

2. Using Client Credentials

Following is a sample request against the server:

```
curl -i -H 'Authorization: Basic U1NPTGlua0NsaWVudDp3ZWxjb211MQ==' -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -H "X-OAUTH-IDENTITY-DOMAIN-NAME: SSOLink" --request POST http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token -d 'grant_type=CLIENT_CREDENTIALS&scope=SSOLink.link1'
```

3. Using JWT Bearer Token

Following is a sample request against the server:

```
curl -i -H 'Authorization: Basic U1NPTGlua0NsaWVudDp3ZWxjb211MQ==' -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -H "X-OAUTH-IDENTITY-DOMAIN-NAME: SSOLink" -- request POST http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token -d 'grant_type=JWT_BEARER&scope=SSOLink.link1&assertion=<assertion token value>'
```

- **Using JWT Bearer Flow to get User Data through UserInfo Endpoint**

If an access token got from the JWT bearer flow is required to fetch user data through /UserInfo endpoint, the following scopes have to be assigned to the client and requested during runtime.

Scope	Corresponding OpenID Scope
UserInfo.email	openid email
UserInfo.me (default)	openid (returns the username)
UserInfo.address	openid address
UserInfo.phone	openid phone
UserInfo.profile	openid profile

Client Profile

Following is a sample client profile, which is registered with the UserInfo related scopes.

```
curl -X POST \
  http://<AdminServerHost:Port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/client \
  -H 'Authorization: Basic d2VibG9naWM6d2VsY29tZTE=' \
  -H 'Content-Type: application/json' \
  -d '{"attributes":
[{"attrName":"UserAttr","attrValue":"CustomStaticValue","attrType":"STATIC"},
  {"attrName":"ResServerConstAttr","attrValue":"Overriding client
- static attribute","attrType":"STATIC"}],
"secret":"welcome1","id":"DemoClientID","scopes":["UserInfo.address",
"UserInfo.email"],"clientType":"CONFIDENTIAL_CLIENT","idDomain":"DemoDomain","description":"Client Description","name":"DemoClient","grantTypes":
```

```
["JWT_BEARER"], "defaultScope": "UserInfo.email", "redirectURIs":
[{"url": "http://localhost:8080/Sample.jsp", "isHttps": true}]}'
```

Example

Following is an example of the token request.

```
curl -i -H 'Authorization: Basic U1NPTGlua0NsaWVudDp3ZWxjb211MQ==' -H
"Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -H "X-
OAUTH-IDENTITY-DOMAIN-NAME: SSOLink" -- request POST http://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token - d
'grant_type=JWT_BEARER&scope=UserInfo.email
UserInfo.address&assertion=assertion token value'
```

4. Using Refresh Token

Following is a sample request against the server:

```
curl -i -H 'Authorization: Basic U1NPTGlua0NsaWVudDp3ZWxjb211MQ==' -H
"Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -H "X-OAUTH-
IDENTITY-DOMAIN-NAME: SSOLink" --request POST http://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token -
d 'grant_type=REFRESH_TOKEN&scope=SSOLink.link1&refresh_token=<RefreshTokenVa
lue>'
```

5. Using Resource Owner Credentials with JWT - Client Assertion Token

Following is a sample request against the server:

```
curl -i -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -
H "X-OAUTH-IDENTITY-DOMAIN-NAME: SSOLink" --request POST http://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token -d
'grant_type=PASSWORD&scope=SSOLink.link1&client_assertion=<ClientAssertionTok
enValue>&client_assertion_type=JWT_BEARER&username=weblogic&password=welcomel
'
```

Note:

- Allowed values for `client_assertion_type` are **JWT_BEARER** and **urn:ietf:params:oauth:client-assertion-type:jwt-bearer**
- `redirect_uri` is not expected for 2 legged flows.
- **scope** is optional in 2 legged flows. If not provided, the access token will be generated with the **defaultScope** associated with the client (provided during client registration).

3 - Legged Flows — process

In order to achieve 3-legged flow, you need to perform few manual steps both on the OAM server and on the Webgate before proceeding. The consent page and approver page need to be protected through OAM. If the consent page is customized, this needs to be protected by a Webgate.

- OAM server - On the created Application Domain, you need to add couple of 3 legged resources as described in the given steps.
- Webgate - Modify the `mod_wl_ohs.conf` as mentioned in this section.

OAM Server side steps to be performed

- List of all the resources to be added as part of 3 legged setup. Details for each and every resource is mentioned as in step2.

Row	Resource Type	Host Identifier	Resource URL	Query String	Authentication Policy	Authorization Policy
1	HTTP	webgate11g	/oam**			
2	HTTP	webgate11g	/oauth2/rest**			
3	HTTP	webgate11g	/oam/pages/consent.jsp		Protected Resource Policy	Protected Resource Policy
4	HTTP	webgate11g	/oauth2/rest/approval		Protected Resource Policy	Protected Resource Policy
5	HTTP	webgate11g	/*		Protected Resource Policy	Protected Resource Policy

Columns Hidden 2

- Create a resource `/oauth2/rest/approval`. This has to be protected by Webgate.

Uri

* Resource URL

Query Name Value list String

Name	Value
No Data to Display	

Operations

* Operations Available

- All
- POST
- PUT
- HEAD
- TRACE

Protection

* Protection Level

Authentication Policy

Authorization Policy

- Create a resource `/oauth2/rest/approval/skip`. This has to be protected by Webgate.

Uri

* Resource URL

Query Name Value list String

Name	Value
No Data to Display	

Operations

* Operations Available

- All
- GET
- POST
- HEAD
- TRACE

Protection

* Protection Level

Authentication Policy

Authorization Policy

4. Create a resource **"/oam/pages/consent.jsp"** which is the out of the box consent page. If you are using a custom consent page, it needs to be protected by Webgate and the appropriate resource has to be added here.

Uri

* Resource URL

Query Name Value list String

Query		+ Add	X Delete
Name	Value		
No Data to Display			

Operations

* Operations Available

- All
- GET
- POST
- HEAD
- TRACE

Protection

* Protection Level

Authentication Policy

Authorization Policy

5. Create a resource **"/oauth2/rest/**"** and mark the Protection level as Excluded.

* Resource URL

Query Name Value list String

Query		+ Add	X Delete
Name	Value		
No Data to Display			

Operations

Protection

* Protection Level

6. Create a resource **"/oam/**"** and mark the Protection level as Excluded.

Uri

* Resource URL

Query Name Value list String

Query		+ Add	X Delete
Name	Value		
No Data to Display			

Operations

* Operations Available

- All
- GET
- POST
- HEAD
- TRACE

Protection

* Protection Level

Authentication Policy

Authorization Policy

Webgate Side steps to be performed

Open and update the **mod_wl_ohs.conf** file which is located at `<OHS_HOME>/user_projects/domains/base_domain/config/fmwconfig/components/OHS/<ohs instance name>` location and add the below entry.

```
<Location /oauth2>
SetHandler weblogic-handler
WebLogicHost <Managed Server Host Name>
WebLogicPort <Managed Server Port>
ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
</Location>
```

```
<Location /oam>
SetHandler weblogic-handler
WebLogicHost <Managed Server Host Name>
WebLogicPort <Managed Server Port>
ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
</Location>
```

3 Legged Flows

1. Through the Browser request the Authorization Code

Following is a sample request against the server

```
http://<OHS Hostname>:<OHS Port>/oauth2/rest/authz?
response_type=code&client_id=TestClient2&domain=TestDomain1&scope=TestRS.scope1+Tes
tRS.scope2+TestRS.scope3&state=xyz&redirect_uri=http://localhost:8080/SampleTest/
index.jsp
```

2. Generate the Access Token using the Authorization Code

Following is a sample request against the server

```
curl --request POST \ --url http://<ManagedServerHost>:<ManagedServerPort>/oauth2/
rest/token --header 'authorization: Basic U1NPTGlua0NsaWVudDp3ZWxjb21lMQ==' --
header 'cache-control: no-cache' --header 'content-type: application/x-www-form-
urlencoded' --header 'x-oauth-identity-domain-name: SSOLink' --data
grant_type=AUTHORIZATION_CODE&code=bnAreDZVMUxEemZtZmJPUEE2U1N2QT09fmVBUVJZYnFtYmZF
SU1EaUFpSktvQjVwQ0ZGQm4xV1R4dmJrekp0MTdDZXdPYjJFNjEwVkdhZlN3VWJjTWcvRUpwL3RqWERUZw1
iZWdUSzZPQkxQNktwQk03c0ZKMEV1NmN3SmxwbG15b1U4MnZ6S1pXRFB6ekdiU1k3V3FEZ31LSjgxM0NwUG
NwUjk1eXI5enRKb0ZLb1VWZ0hqNm53TkVFTepKMmtKNmY3b1ZHWDFtcFkvL1haMUs4N0xiRGlmbkFwTWpHd
1J5QjVuZkdXTzh4U01hamdWZnNmT3doSlo1SS9KY3NtOGNaQkxJxMDd3SzgrWXBicVYxYlgxYzFLSWhubW5M
WndZQTg5ZnV0aU1Kam54bytZaGZhbW5IK2xrNjFByVhXOHb5SEdENG5SRzJ2aytDcjRHRlg2OWZFbTdT&re
direct_uri=http%3A%2F%2Fredirect_uri'
```

Validating Access Tokens

```
curl -i -H "X-OAUTH-IDENTITY-DOMAIN-NAME: SSOLink" --request GET "http://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/info?
access_token=<AccessToken>"
```

37.11 Revoking OAuth Tokens

OAM provides Runtime and Admin REST APIs for revoking OAuth Access and Refresh Tokens.

Topics

- [Revoking OAuth Tokens by OAuth Clients](#)
- [Revoking OAuth Tokens for a User, Client and Resource Server](#)

37.11.1 Revoking OAuth Tokens by OAuth Clients

OAM provides support for the OAuth clients to revoke access and refresh tokens using the `http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/revoke` Runtime REST API.

- OAM supports token revocation of only those access and refresh tokens that are generated through 3-legged OAuth flow (Authorization code flow). For tokens generated through 2-legged OAuth flow, there is no user consent and the client has all the information required to regenerate the tokens. If the client is compromised, it is recommended to delete the client.

OAM returns the following error if you provide tokens generated by 2-legged OAuth flow in the request: **RESPONSE CODE: 415**

```
{"error": "unsupported_token_type", "error_description":  
  "Revocation of the presented token type not supported."}
```

- Consent Management must be enabled for the OAuth tokens to be revoked. If it is not enabled, OAM returns the following error: **RESPONSE CODE: 500**

```
{  
  "error": "token_revocation_without_cmlc_not_supported",  
  "error_description": "Consent management must be enabled for  
  TOKEN REVOCATION support"  
}
```

For details about how to enable consent management, see [Enabling Consent Management](#)

For details about how to enable consent management in an MDC setup, see [Enabling Consent Management on MDC](#)

- In an MDC setup, the time taken for the OAuth token revocation to propagate to all the data centers is dependent on the value set in `pollInterval`. Recommended value is 5 seconds. That is, if the client revokes the OAuth tokens on one data center, it takes at least five seconds for the changes to get propagated to the other data center in the MDC topology. For more details about changing `pollInterval`, see [Modifying the Polling Interval in Clone Data Centers](#). Also, it is recommended to set the `BatchSize` to 300. For details, see [Table 19-2](#)

Ensure the following headers are added in the request:

- `Authorization`: Base64 URL encoded `<ClientID>:<Secret>` combination.
- `X-OAUTH-IDENTITY-DOMAIN-NAME`: Identity Domain that the client belongs to.
- `Content-Type`: `application/x-www-form-urlencoded`: The token is provided as a key-value pair in the request body.

For more information about the parameters, response codes, error codes, and so on, see [Revoke Token REST Endpoints](#) REST API documentation.

Revoking a Specific Access or Refresh Token

OAuth clients can use the `http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/revoke` API to revoke a specific access or refresh token.

OAM implements token revocation as defined in **RFC 7009**. Refer to <https://tools.ietf.org/html/rfc7009> for information.

Sample Request to Revoke a Token

```
curl --location --request POST '<ManagedServerHost>:<ManagedServerPort>/
oauth2/rest/token/revoke' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Authorization: Basic RGVtb0NsaWVudElkOndlbGNvbWUx' \
--header 'Cookie: JSESSIONID=VKgaCGYiIiQ_3-
gTdIOymIqW4uMXGt2OPNg1GjTvJUaVyp4gkNY3!-472705583' \
--data-urlencode
'token=eyJraWQiOiJEZW1vRG9tYWluIiwieDV0IjoizG1HQi1zR1BlcHEzb1E2ZWdyRnNkM3c4ZU
9
JiIwiYWxnIjoUlMyNTYifQ.eyJpc3MiOiJodHRwOi8vc2xjMDdoYWYudXMub3JhY2xlLmNvbToyM
jIyL29hdXR0MiIsImF1
ZCI6WyJEZW1vUmVzU2VydmVyIiwiaWF0Ij0sImV4cCI6MTYxNTM1MDY3NywiYW90IjoiaXN0b21BdHRyaXNlc3Npb25JZCI6I
kNkc0F3Q2JvcHY2OHoz
USIsImhhdCI6MTYxNTM0NzA3Nywic3ViIjoiaXNlc3Npb25JZCI6I
jM4M2E0YjA3LTg4NWUt
NGIwYy05MzZhLWRjYTVlYzlmOGMyZnwwSDZ2OHZJc0NzaUR1T3l1MFN0RHVnUE1rdXQ1VUxVSU0tQZ
GZpait0ejhnPSIsImk
Y19zc29fbGluayI6IjY2YmJjLXNsYzA3aGFmLnV-
flVzZXJZJGVudGl0eVN0b3JlMSIsImNlc3RvbUF0dHJpMiI6Ii1JFU09V
UkNFQ090U1QiLCJjbG11bnQiOiJEZW1vQ2xpZW50SWQ1LCJzY29wZSI6WyJEZW1vUmVzU2VydmVyL
kRlZmF1bHRTY29wZSJD
LCJkb21haW4iOiJEZW1vRG9tYWluIiwiaXNlc3RvbUF0dHJpMiI6Ii1JFU09V
UfH2Pi9ipTKIaAAB5CP
B5lwLd8ga_39ruDQEfckqYcgHAToTqFfn6uibbEn0EluxJqE_rnT6ABLWQ0VABruMRRiK2f7qGS
SWXdakjjWmYG4pVJTgN
64OrEroUPM-65ZvqIDmMiYG-80dVJpQq3QxG9_7yJhG0g8Rf1cxGITJb2_RLn19Ke1Wmex5LcMKRG
GhiezA98o_jluknmzdUi
Pw1C2NGxXESvTgMnM7Q49exnG1py-
aZ7KaohF8misS2_Hbgx4f-2ipG8CQtefiwCEDvPpk30QgKuU4GGmV9yU1V3qd-yqJTUA
4EbvffsNeadSjhs0v3Sjw8ZNq1g'
```

Sample Response

```
{
  "status": "success"
}
```

Revoking Related Consents, Access, or Refresh Tokens

If a refresh token is created along with the access token during the 3-legged OAuth flow, that refresh token can be used to generate the access tokens multiple times till the time the refresh token is valid. For more information, see [OAuth Refresh Tokens](#)

To revoke all such related refresh and access tokens, the OAuth clients can additionally use the `chaining_level` parameter along with the `token` parameter in the `http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/revoked` API.

The following table shows the values supported in the `chaining_level` parameter and the corresponding behavior.



Note:

The `token` parameter is mandatory; however, `token_type` is optional. The token type is decided based on the token provided in the `token` parameter.

Table 37-6 `chaining_level` Values and Behavior

If the Specified Type of Token is ...	<code>chaining_level</code> values	Behavior
Refresh Token	NONE	Default. The refresh token specified in the <code>token</code> parameter is revoked.
	RELATED_TOKENS	In addition to the refresh token specified in the <code>token</code> parameter, all the access tokens that were generated by that refresh token is revoked.
	RELATED_CONSENT	The consent is deleted. In addition to the refresh token specified in the <code>token</code> parameter, all the access and refresh tokens created by the grant/consent (associated with the specified refresh token) is revoked.
Access Token	NONE	Default. The access token specified in the <code>token</code> parameter is revoked.
	RELATED_TOKENS	In addition to the access token specified in the <code>token</code> parameter, the parent refresh token, that was used to generate the specified access token, is revoked.
	RELATED_CONSENT	The consent is deleted. In addition to the access token specified in the <code>token</code> parameter, all the access and refresh tokens created by the grant/consent (associated with the specified access token) is revoked.

The following example shows a sample request to revoke refresh token and the access tokens generated using that refresh token.

Sample Request

```
curl --location --request POST
'<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/revoke' \
--header 'Content-Type: application/x-www-form-urlencoded' \
```

```

--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Authorization: Basic RGVtb0NsaWVudElkOndlbGNvbWUx' \
--header 'Cookie:
JSESSIONID=NysahaonLyc13HzjEh93gJmwbY4HnMetJQY33RE8_ZdBpZpw7kdr!-472705583' \
--data-urlencode
'token=LCbzQggeRM1EMqprtKrHuQ%3D%3D%7EcsLp2lL9J03orCX0dTvBySFAXG4Yi%2BI%2FOq8
0
ChZzVsz1BrME2GEg9Kuk6aShduv0K%2F8Yzhs6F4RCOdXg0luZilu3V544Hf%2FziaoJFZGDr4Umf
kLHByMTJYWTJXfR%2F
MUQkkDjffRALox1vVjztUbhb1uKmkZWE%2FhTYHCp1pkc2zNJC7j7KQaIF%2BkNfg8GPS%2FdjeLo
7i99%2B%2Bifb%2BKq
GTnaJWOr2JSm7XApoG1X9dwBzM8EHd04IQNPYDxkvtQLajVxlRhK5ZnL3F29wBD4yOuXqg%3D%3D'
\
--data-urlencode 'chaining_value=RELATED_TOKENS' \
--data-urlencode 'token_type=REFRESH_TOKEN'

```

Sample Response

```

{
  "status": "success"
}

```

The following example shows a sample request to revoke access token and its parent refresh token from which the access token was generated.

Sample Request

```

curl --location --request POST '<ManagedServerHost>:<ManagedServerPort>/
oauth2/rest/token/revoke' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Authorization: Basic RGVtb0NsaWVudElkOndlbGNvbWUx' \
--header 'Cookie: JSESSIONID=VKgaCGYiIiQ_3-
gTdIOymIqW4uMXGt2OPNg1GjTvJUaVyp4gkNY3!-472705583' \
--data-urlencode
'token=eyJraWQiOiJEZWlwRG9tYWluIiwieDV0IjoizG1HQi1zR1BlcHEzblE2ZWdyRnNmM3c4ZU
9
JIiwiYWxnIjoilMyNTYifQ.eyJpc3MiOiJodHRwOi8vc2xjMDdoYWYudXMub3JhY2xlLmNvbToyM
jIyL29hdXR0MiIsImF1
ZCI6WyJEZWlwUmVzU2VydmVyIiwiaWF0Ij0sImV4cCI6MTYxNTMlNTAwOSwianRpIjoizVetVHA1N
3JUTHFUTktPaERYbEpnd
yIsIm1hdCI6MTYxNTMlMTQwOSwic3ViIjoiaXNlcKEiLCJjdxN0b21BdHRyaXNlc3Npb25JZCI6Ij
gxOWY5ODc0NDdjMGE1Mzg
3Mzk1Mjg5OWYzZjUzMTYzMTUzZjRlODI5ZTNmN2FmMjg2OWFhNWY5M2YyMmMyYzMiLCJjdxN0b21B
dHRyaTIiOiJlSRVNPVJDR
UNPTlNUIiwiY2xpZW50IjoirGVtb0NsaWVudElkIiwic2NvcGUiOlsirGVtb1Jlc1NlcnZlci5EZW
ZhdWx0U2NvcGUxSWizG9
tYWluIjoirGVtb0RvbWVpbiIsInJ0X2lkIjoim2FmN2Q5NzEtYjAyZS00ZDI5LThlOTMtNWJhMTkz
MGJkN2Y2LjE2MTUzNTEzN
jciLCJncmFudCI6IkFVVEhPuklaQVRJT05fQ09ERSJ9.HBtENqB6nIAUA1cft84o5_tSfIXP_E-
tD7Ux6Wyc_n00D1m2x6l7sc
9oQ08ad1vgXV4KjSGSPnxL09pWLUPDxhwQqs15w_Pylq-SWQxrcpKqtCv-
vdz_zCS3_uMsaOLQTQwYyj94tnS9TEWAKQtKnDrV

```

```

4vOFhhDMVOOBPIo5h7BeDa91IshPj1B7wPAHJ2HX4rOhL5z2BfPS2v9QNUDuJbvKZ178Bw
HP80L8uDaSsh5a0XMcJGr1PQ9cd0
6bSWGTF3o9NBLWBiWnPq17XDxnr4rEcPrp3bV_iwuIvo0id91iu52L-
NTAts0FzTv7gEz2d61NMwgMAfBFQPFYRQ' \
--data-urlencode 'chaining_value=RELATED_TOKENS' \
--data-urlencode 'token_type=ACCESS_TOKEN'

```

Sample Response

```

{
  "status": "success"
}

```

The following example shows a sample request to delete the consent and also revoke all the access and refresh tokens generated by the consent associated with the specified refresh token.

Sample Request

```

curl --location --request POST
'<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/revoke' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Authorization: Basic RGVtb0NsaWVudElkOndlbnGNvbWUx' \
--header 'Cookie: JSESSIONID=VKgaCGYiIiQ_3-
gTdIOymIqW4uMXGt2OPNg1GjTvJUaVyP4gkNY3!-472705583' \
--data-urlencode
'token=LCbzQggeRM1EMgprtKrHuQ%3D%3D%7EcsLp21L9J03orCX0dTvBySFAXG4Yi%2BI
%2FOq80
ChZzVsz1BrME2GEg9Kuk6aShdUV0K%2F8Yzhs6F4RCodXg01uZilu3V544Hf%2FziaoJFZG
Dr4UmfkLHByMTJYWTJXfR%2F
MUQkkDjffRALox1vVjztUbhBluKmkZWE%2FhTYHCp1pkc2zNJC7j7KQaIF%2BkNfg8GPS%2
FdjeLo7i99%2B%2Bifb%2BKq
GTnaJWOr2JSm7XApoG1X9dwBzM8EHd04IQNPYDxkvtQLajVxlRhK5ZnL3F29wBD4yOuXqg%
3D%3D' \
--data-urlencode 'chaining_value=RELATED_CONSENT' \
--data-urlencode 'token_type=REFRESH_TOKEN'

```

Sample Response

```

{
  "status": "success"
}

```

Validating OAuth Token Revocation

After running the OAuth token revocation APIs, you can verify if the tokens have been successfully revoked in the following ways:

- To verify if the refresh token has been revoked, you can use the refresh token to generate a new access token. If the refresh token has been successfully revoked,

the access token generation fails. If the refresh token is still valid (not revoked) a new access token is generated.

Sample Request

```
curl --location --request POST '<ManagedServerHost>:<ManagedServerPort>/
oauth2/rest/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Authorization: Basic RGVtb0NsaWVudElkOndlbGNvbWUx' \
--header 'Cookie:
JSESSIONID=NysahaonLyc13HzjEh93gJmwbY4HnMetJQY33RE8_ZdBpzw7kdr!-472705583
' \
--data-urlencode 'grant_type=REFRESH_TOKEN' \
--data-urlencode
'refresh_token=LCbzQggeRM1EMgprtKrHuQ%3D%3D%7EcsLp2lL9J03orCX0dTvBySFAXG4Y
i%2BI%2FOq80
ChZzVsz1BrME2GEg9Kuk6aShdUV0K%2F8Yzhs6F4RCodXg01uZi1u3V544Hf%2FziaoJFZGDr4
UmfkLHByMTJYWTJXfR%2F
MUQkkDjffRALox1vVjztUbhB1uKMkZWE%2FhTYHCp1pkc2zNJC7j7KQaIF%2BkNfg8GPS%2Fdj
eLo7i99%2B%2Bifb%2BKq
GTnaJW0r2JSm7XApoG1X9dwBzM8EHdO4IQNPYDxkvtQLajVx1RhK5ZnL3F29wBD4yOuXqg%3D%
3D'
```

Sample Response

```
{
  "error": "invalid_grant",
  "error_description": "Invalid Refresh Token"
}
```

- To verify if the access token has been revoked, use `http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/info` REST API.

Sample Request

`http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/info`

```
curl --location --request POST '<ManagedServerHost>:<ManagedServerPort>/
oauth2/rest/token/info' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Cookie: JSESSIONID=VKgaCGYiIiQ_3-
gTdIOymIqW4uMXGt2OPNg1GjTvJUaVYP4gkNY3!-472705583' \
--data-urlencode
'access_token=token=eyJraWQioiJEZW1vRG9tYWluIiwieDV0IjoizG1HQilzR1BlcHEzbl
E2ZWdyRnNkM3c4ZU9
JIiwiYWxnIjoilUMyNTYifQ.eyJpc3MiOiJodHRwOi8vc2xjMDdoYWYudXMub3JhY2xlLmNvbT
oyMjIyL29hdXR0MiIsImF1
ZCI6WyJlZjZlVUmVzU2VydMvYiwiYWIl0sImV4cCI6MTYxNTM1NTAwOSwianRpIjoizVEtVH
A1N3JUTHFUTktPaERYbEpnd
yIsImIhdCI6MTYxNTM1MTQwOSwic3ViIjoiaXNlcEiLCJjdXN0b21BdHRyaXNlc3Npb25JZCI
6IjgOWY5ODc0NDdjMGE1Mzg
3Mzk1Mjg5OWYzZjUzMTYzMTUxZjRlODI5ZTNmN2FmMjg2OWFhNWY5M2YyMmMyZmIiLCJjdXN0b
21BdHRyaTIiOiJlSRVNPVJDR
UNPt1NUIiwiY2xpZW50IjoiaXNlcEiLCJjdXN0b21BdHRyaXNlc3Npb25JZCI6IjgOWY5ODc0NDdjMGE1Mzg'
```

```

EZWZhdWx0U2NvcGUixSwizG9
tYWluIjoiRGVtb0RvbWFpbiIsInJ0X2lkIjoiM2FmN2Q5NzEtYjAyZS00ZDI5LThlOTM
tNWJhMTkzMGJkN2Y2LjE2MTUzNTEzN
jciLCJncmFudCI6IkFVVEhPuklaQVRJT05fQ09ERSJ9.HBtENqB6nIAUA1cft84o5_ts
FiXP_E-tD7Ux6WYc_n00D1m2x617sc
9oQO8ad1vgXV4KjSGSPnxL09pWLUPDxhwQqs15w_Py1q-SWQxrcpKqtCv-
vdz_zCS3_uMsaOLQTQwYyj94tnS9TEWakQtknDrV
4vOFhhDMVOOBPIo5h7BeDa91iSiHPj1B7wPAHJ2HX4rOhL5z2BfPS2v9QNUDuJbvKZ17
8BwHP80L8uDaSsh5a0XMcJGr1PQ9cd0
6bSWGTF3o9NBLWBiWnPtyq17XDxnr4rEcPrp3bV_iwuIvo0id91iu52L-
NTAts0FzTv7gEz2d61NMwgMAfBFQPFYRQ'

```

Sample Response

```

{
  "error": "invalid_grant",
  "error_description": "Access Token Validation Failed"
}

```

37.11.2 Revoking OAuth Tokens for a User, Client and Resource Server

In addition to the Runtime API for revoking tokens by OAuth clients, OAM also provides an Administrator API to support revoking all 3-legged OAuth tokens for a specific user, or a specific user-client-resource server combination.

Use the `http://<AdminServerHost>:<AdminServerPort>/oam/services/rest/consent/revoke` REST API to revoke all the OAuth tokens for a specific user, or a specific user-client-resource server combination.

- OAM supports token revocation of only those access and refresh tokens that are generated through 3-legged OAuth flow (Authorization code flow). For tokens generated through 2-legged OAuth flow, there is no user consent and the client has all the information required to regenerate the tokens. If the client is compromised, it is recommended to delete the client.
- Consent Management must be enabled for the OAuth tokens to be revoked. If it is not enabled, OAM returns an empty consent:

```
[ ]
```

For details about how to enable consent management, see [Enabling Consent Management](#)

For details about how to enable consent management in an MDC setup, see [Enabling Consent Management on MDC](#)

- In an MDC setup, the time taken for the token revocation to propagate to all the clone data centers in an MDC setup is dependent on the value set in `pollInterval`. Recommended value is 5 seconds. That is, if the client revokes the OAuth tokens on the Master data center, it takes at least five seconds for the changes to get propagated to the Clone data centers in the MDC topology. For details about changing `pollInterval`, see [Modifying the Polling Interval in Clone Data Centers](#)

Also, it is recommended to set the `BatchSize` to 300. For details, see [Table 19-2](#)

In an MDC setup, this Administrator REST API must be run on the Master node.

Ensure the following headers are added in the request:

- `Authorization`: Base64 URL encoded `<Administrator>:<Secret>` combination.
- `X-OAUTH-IDENTITY-DOMAIN-NAME`: Identity Domain name.
- `Content-Type`: `application/x-www-form-urlencoded`: Add the following parameters as key-value pairs in the request body, as required.

<code>userId</code>	Mandatory.
<code>revoke_type</code>	Optional. Supported values: <code>REFRESH_TOKENS</code> , <code>ACCESS_TOKENS</code> , <code>TOKENS</code> . If you do not specify this parameter, or if you specify <code>TOKENS</code> as value then all the access and refresh tokens are revoked.
<code>timestamp</code>	Optional. If specified, the tokens issued before this timestamp are revoked.
<code>clientIdentifier</code>	Optional. If specified, <code>resServerId</code> also must be specified.
<code>resServerId</code>	Optional. Must be specified if <code>clientIdentifier</code> is specified.

For more information about the parameters, response codes, error codes, and so on, see [Revoke Token REST Endpoints](#) REST API documentation.

Examples

Example 37-1 Revoking All the OAuth Tokens for a User

Specify the `userId` whose tokens needs to be revoked.

Sample Request

```
curl --location --request POST '<AdminServerHost>:<AdminServerPort>/oam/
services/rest/consent/revoke' \
--header 'Authorization: Basic d2VibG9naWM6d2VsY29tZTE=' \
--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Cookie: JSESSIONID=yIwmL5y29hILQ44F1-
yt0t78Lgw00YNVY1Z_gaz4cqW8xS01ekuE!-1307751471' \
--data-urlencode 'userId=UserA'
```

Sample Response

```
{
  "consents": [
    {
      "clientId": "DemoClientId",
      "consentId": "30650989-8e53-3010-b06a-98b0ef42b65d",
      "createTimeStamp": "Fri Mar 12 03:24:08 PST 2021",
```

```

        "resourceId": "66ac1a16-ee37-4525-81f6-9062d69a743c",
        "scopes": [
            "DemoResServer.DefaultScope"
        ],
        "tokenRevokeTimestamp": "TOKENS=2021-03-12T03:30:31-0800",
        "valid": true
    }
}

```

Example 37-2 Revoking All Refresh Tokens for a Client

Specify the `userId`, `revoke_type`, `clientIdentifier`, and `resServerId` to revoke all `ACCESS_TOKENS` for the specified client.

Sample Request

```

curl --location --request POST
'<AdminServerHost>:<AdminServerPort>/oam/services/rest/consent/revoke' \
\
--header 'Authorization: Basic d2VibG9naWM6d2VsY29tZTE=' \
--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Cookie: JSESSIONID=yIwmL5y29hILQ44F1-
yt0t78Lgw00YNVY1Z_gaz4cqW8xS01ekuE!-1307751471' \
--data-urlencode 'userId=UserA' \
--data-urlencode 'revoke_type=REFRESH_TOKENS' \
--data-urlencode 'clientIdentifier=DemoClientId' \
--data-urlencode 'resServerId=66ac1a16-ee37-4525-81f6-9062d69a743c'

```

Sample Response

```

{
  "consents": [
    {
      "clientId": "DemoClientId",
      "consentId": "30650989-8e53-3010-b06a-98b0ef42b65d",
      "createTimeStamp": "Fri Mar 12 03:55:31 PST 2021",
      "resourceId": "66ac1a16-ee37-4525-81f6-9062d69a743c",
      "scopes": [
        "DemoResServer.DefaultScope"
      ],
      "tokenRevokeTimestamp":
"REFRESH_TOKENS=2021-03-12T03:56:49-0800",
      "valid": true
    }
  ]
}

```

Example 37-3 Revoking Refresh Tokens for a User Based on Timestamp

Use the `timestamp` parameter to revoke refresh tokens generated before the specified timestamp.

Specify the timestamp in the following format: [yyyy]-[MM]-[dd] 'T' [HH]:[mm]:[ss]Z, where

- Z is the time offset from UTC in the format **+/-HHmm** and denotes +0000 numerically. For example, the UTC offset for New York on standard time can be specified as -0500.
- yyyy is the year
- MM is the month
- dd is the day
- HH is hour
- mm is minutes
- ss is seconds

The following sample request revokes all the tokens for UserA that were generated before the timestamp 2021-03-09T15:30:33+0800

Sample Request

```
curl --location --request POST '<AdminServerHost>:<AdminServerPort>/oam/
services/rest/consent/revoke' \
--header 'Authorization: Basic d2VibG9naWM6d2VsY29tZTE=' \
--header 'x-oauth-identity-domain-name: DemoDomain' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Cookie:
JSESSIONID=NysahaonLyc13HzjEh93gJmwbY4HnMetJQY33RE8_ZdBpzw7kdr!-472705583' \
--data-urlencode 'userId=UserA' \
--data-urlencode 'revoke_type=REFRESH_TOKENS' \
--data-urlencode 'timestamp=2021-03-09T15:30:33+0800'
```

Sample Response

```
{
  "consents": [
    {
      "clientId": "DemoClientId",
      "consentId": "30650989-8e53-3010-b06a-98b0ef42b65d",
      "createTimeStamp": "Tue Mar 09 21:12:06 PST 2021",
      "resourceId": "66ac1a16-ee37-4525-81f6-9062d69a743c",
      "scopes": [
        "DemoResServer.DefaultScope"
      ],
      "tokenRevokeTimeStamp":
"REFRESH_TOKENS=2021-03-08T23:30:33-0800",
      "valid": true
    }
  ]
}
```

37.12 Configuring Client Authentication



This content applies only to OAM Bundle Patch 12.2.1.4.210920 and later releases.

OAM supports the following client authentication methods:

- `client_secret_basic`: uses password for authentication.
- `private_key_jwt`: OAM generates JWT tokens for authentication.
- `tls_client_auth`: mTLS client authentication.
- `self_signed_tls_client_auth`: self-signed mTLS client authentication.

See the following sections for details about Client authentications:

- [Configuring Identity Domain for Client Authentication](#)
- [Configuring the Client for Client Authentication](#)

37.12.1 Configuring Identity Domain for Client Authentication

You can add the properties in the body of the `https://<admin-host>:<admin-port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/oauthidentitydomain` REST API while creating the Identity Domain. Use the `PUT` method to update the existing Identity Domain with the properties.



Note:

See the [Identity Domain REST Endpoints](#) documentation for details about the properties.

For example,

```
{
  "tokenEndpointAuthMethodsSupported": [
    "tls_client_auth",
    "self_signed_tls_client_auth",
    "private_key_jwt",
    "client_secret_basic"
  ],
  "issueTLSClientCertificateBoundAccessTokens": "true",
  "tlsClientAuthSubjectDN": "CN=%CLIENT_ID_PLACEHOLDER%, OU=OAM, O=Oracle, L=BLR, ST=KA, C=IN"
}
```

 **Note:**

- Do not replace `%CLIENT_ID_PLACEHOLDER%` with the client id while creating or modifying the identity domain. OAM Server replaces the `%CLIENT_ID_PLACEHOLDER%` during authentication.
- You must disable client certificate propagation from OHS to WLS if you intend to use the `client_secret_basic` authentication method. This is due to the fact that if a client certificate is present, mTLS validation will occur automatically.
- mTLS will be enforced if the client certificate is sent in the request, regardless of whether it is part of the `tokenEndpointAuthMethodsSupported` list. Therefore, if OAM needs to ignore client certificates, then you must configure OHS/LBR so that the certificates do not pass to the OAM server.

Identity Domain can include various type of clients, therefore, multiple authentication methods can be configured in the identity domain. The following is the order of precedence of the authentication methods:

1. mTLS authentication methods: `tls_client_auth` and `self_signed_tls_client_auth`
2. JWT Key authentication method: `private_key_jwt`
3. Client Secret authentication method: `client_secret_basic`

 **Note:**

If the client includes configuration for any of the authentication methods then only that authentication method is used.

37.12.2 Configuring the Client for Client Authentication

You can add the properties in the body of the `https://<admin-host>:<admin-port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/client` REST API while creating the client. Use the `PUT` method to update the existing client with the properties.

 **Note:**

See [Client REST Endpoints](#) documentation for details about the properties.

For example,

```
{
  "tokenEndpointAuthMethod":"tls_client_auth",
  "issueTLSClientCertificateBoundAccessTokens":"true",
  "tlsClientAuthSubjectDN":"C=IN, ST=KA, L=BLR, O=Oracle, OU=OAM,
  CN=client2, EMAILADDRESS=client2@oracle.com"
}
```

**Note:**

Ensure that the `tlsClientAuthSubjectDN` values match the values in the client certificate.

Authentication method is optional in client configuration. If the authentication method is not provided in the client then the domain level configuration is used. For details, see [rfc6749](#)

37.12.3 Managing Client Certificates

In case of `CONFIDENTIAL_CLIENT` (this is set using the parameter `clientType` while creating the client. For details, see REST API documentation), only the CA and intermediate CA certificates must be added to the OAM truststore.

For a `PUBLIC_CLIENT` or a self-signed client certificate, the client certificate must also be included with the client. Use the `https://<admin-host>:<admin-port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/clientartifacts` REST API to add the client certificates to the client.

For example:

```
curl --location --request POST 'https://<admin-host>:<admin-port>/oam/
services/rest/ssa/api/v1/oauthpolicyadmin/clientartifacts' \
--header 'Accept: application/json' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--header 'Content-Type: application/json' \
--data '{
  "certificateValue": "<CLIENT_CERTIFICATE>",
  "clientName": "MTLSClient",
  "identityDomainName": "MTLSDomain"
}'
```

**Note:**

Certificate that has the issuer matching the subject is treated as self-signed certificate.

37.13 Configuring mTLS Client Authentication



This content applies only to OAM Bundle Patch 12.2.1.4.210920 and later releases.

Topics:

- [About Mutual Transport Layer Security \(mTLS\) in OAM](#)

- [Configuring mTLS Endpoint](#)
- [Managing Trust Certificates for mTLS](#)
- [Configuring Additional Options to Support mTLS](#)
- [Sample 2-Legged mTLS Authentication Flow](#)
- [Sample 3-Legged mTLS Authentication Flow](#)

37.13.1 About Mutual Transport Layer Security (mTLS) in OAM

In TLS authentication, the server confirms its identity by producing a certificate (public key), which is then verified by the TLS verification process.

In mTLS (mutual-TLS), along with the server, the client's identity is also verified. The TLS handshake is utilized to validate the client's possession of the private key corresponding to the public key in the certificate and to validate the corresponding certificate chain.

OAM supports configuration to validate the certificate chain at the server and configurations that support termination of SSL at load balancer or proxy end. Load balancer or proxy end points, where SSL is terminated, need not validate the certificate chain as OAM server does the validation.

About Certificate Binding

Certificate binding can be enabled with or without mTLS authentication in OAM. This enables mutual TLS during protected resource access to serve as a proof-of-possession mechanism.

TLS Client certificate binding is not mandatory. Binding is done based on the `issueTLSClientCertificateBoundAccessTokens` configuration set in the Identity Domain and the Clients.

To configure certificate binding, you can set `issueTLSClientCertificateBoundAccessTokens` as `true` or `false` while creating or modifying the Identity Domain or Clients.

The certificate bound access token contains `cnf` (confirmation) entry, which contains the thumbprint of the client certificate to which the access token is bound.

You can use the introspect endpoint to validate the certificate binding.



Note:

The `cnf` entry is present only in the response of the introspect endpoint with certificate bound access token and is not present in access token that is not bound to a certificate.

37.13.2 Configuring mTLS Endpoint

Prerequisite: Ensure you have performed the following procedures before configuring the mTLS endpoint:

- [Configuring Identity Domain for Client Authentication](#)
- [Configuring the Client for Client Authentication](#)

To configure the mTLS endpoint, set the `hostname` and `port` for the mTLS endpoint using the `https://<admin-host>:<admin-port>/oam/services/rest/ssa/api/v1/hostalias/mtls` REST API.

 **Note:**

The `hostname` and `port` of the mTLS endpoint must match the endpoint, at which SSL gets terminated. For example, if SSL terminates at the loadbalancer then the `hostname` and `port` of the mTLS endpoint must match the `hostname` and `port` of the loadbalancer.

```
curl --location --request POST 'https://<admin-host>:<admin-port>/oam/
services/rest/ssa/api/v1/hostalias/mtls' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--header 'Content-Type: application/json' \
--data '{
  "hostname": "<HOST_NAME>",
  "port": "4443"
}'
```

After setting up the mTLS configuration, you can also view the details using the discovery endpoint (`.well-known` URL, for example: `http://<HOST_NAME>:7777/.well-known/openid-configuration`).

The response includes the new parameter `mtls_endpoint_aliases`

```
"mtls_endpoint_aliases": {
  "token_endpoint": "https://<MTLS_HOST>:<MTLS_PORT>/oauth2/rest/
token",
  "revocation_endpoint": "https://<MTLS_HOST>:<MTLS_PORT>/oauth2/
rest/token/revoke",
  "introspection_endpoint": "https://<MTLS_HOST>:<MTLS_PORT>/oauth2/
rest/token/introspect"
},
```

Also see, [Configuring OpenIDConnect Discovery Endpoint](#)

37.13.3 Managing Trust Certificates for mTLS

Clients must provide certificates for mTLS authentication. The certificates can be signed from a known Certificate Authority (CA), self-signed, or from a custom CA. In case of certificate chaining, the client must include the certificate chain.

Trusted CA certificates must be added to the OAM truststore. Use the `https://<admin-host>:<admin-port>/oam/services/rest/ssa/api/v1/security/trust/oauthClient/certificate` REST API to manage trusted CA certificates.


```
6P7nkwMfThns1yyHByoE7WPQd7ans4WBX/HTozfz1xLh50QFqpKDS4C6mD/
M9fffSuSHOvR4mVug8cWghORmwc1SSuLLzxX1+LFmnGLwJNtP1ffPCfZqPRrd01PcO/
ifGPz50zfU="
}
```

2. Enable OCSP certificate validation. For details, see [Enabling OCSP Certificate Validation](#)

37.13.4 Configuring Additional Options to Support mTLS

Perform the additional configurations to support mTLS authentication

- Set the WebLogic Plug-In Enabled to Yes.
 1. Login to the WebLogic Console and navigate to the server instance (for example, **oam_server1**, and **Advanced**)
 2. Check the **Client Cert Proxy Enabled** checkbox
 3. Set the **WebLogic Plug-In Enabled** option to **Yes**
- If SSL terminates at OHS, perform the following changes in the ohs ssl.conf file under `MIDDLEWARE_HOME/user_projects/domains/base_domain/config/fmwconfig/components/OHS/instances/ohs1`:
 - In the **SSL Wallet**, search for `"${ORACLE_INSTANCE}"` and in the next line, add `SSLOptions, StdEnvVars, and ExportCertData`.
 - Change `SSLVerifyClient none` to `SSLVerifyClient require`
- Configure the load balancer to support mTLS configuration. Configure 2 way SSL endpoint in load balancer and import CA and intermediate CA of client certificate to the load balancer of the trust store. For details, see [Configuring SSL for the Web Tier](#)
- To provide support for custom cert headers, add the parameter `"enableHeaderCertValue":true` under `CustomAttrs` in the Identity Domain REST API. For example:

```
"customAttrs":{"domainCertValidityInDays":"30",
"consentExpiryTimeInMinutes":"10","enableHeaderCertValue":true}"
```

Once enabled, the headers must be provided under the following two http headers:

- `SSL_CLIENT_CERT`: Client certificate.
- `SSL_CLIENT_ROOT_CERT`: Root CA or intermediate CA certificates

Note:

This is applicable only for load balancers other than OHS terminating the SSL connection

37.13.5 Sample 2-Legged mTLS Authentication Flow

The following example provides a sample 2-legged mTLS authentication flow:

1. Create the Identity Domain

```
curl --location --request POST 'https://<admin-host>:<admin-port>/oam/
services/rest/ssa/api/v1/oauthpolicyadmin/oauthidentitydomain' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--header 'Content-Type: application/json' \
--data '{
  "name": "MTLSDomain",
  "identityProvider": "UserIdentityStore1",
  "description": "MTLSDomain",
  "tokenSettings": [
    {
      "tokenType": "ACCESS_TOKEN",
      "tokenExpiry": 3600,
      "lifeCycleEnabled": false,
      "refreshTokenEnabled": false,
      "refreshTokenExpiry": 86400,
      "refreshTokenLifeCycleEnabled": false
    },
    {
      "tokenType": "AUTHZ_CODE",
      "tokenExpiry": 3600,
      "lifeCycleEnabled": false,
      "refreshTokenEnabled": false,
      "refreshTokenExpiry": 86400,
      "refreshTokenLifeCycleEnabled": false
    },
    {
      "tokenType": "SSO_LINK_TOKEN",
      "tokenExpiry": 3600,
      "lifeCycleEnabled": false,
      "refreshTokenEnabled": false,
      "refreshTokenExpiry": 86400,
      "refreshTokenLifeCycleEnabled": false
    }
  ],
  "errorPageURL": "/oam/pages/error.jsp",
  "consentPageURL": "/oam/pages/consent.jsp",
  "customAttrs": "{\"domainCertValidityInDays\":\"30\",
\"consentExpiryTimeInMinutes\":\"10\"}",
  "tokenEndpointAuthMethodsSupported": [
    "tls_client_auth",
    "self_signed_tls_client_auth"
  ],
  "issueTLSClientCertificateBoundAccessTokens":"true",
  "tlsClientAuthSubjectDN":"CN=%CLIENT_ID_PLACEHOLDER%, OU=OAM,
O=Oracle, L=BLR, ST=KA, C=IN"
}'
```

2. Create the Resource

```
curl --location --request POST 'https://<admin-host>:<admin-port>/oam/
services/rest/ssa/api/v1/oauthpolicyadmin/application' \
```

```

--header 'Content-Type: application/json' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--data '{
  "name": "MTLSResource",
  "description": "Oracle Cloud",
  "scopes": [
    {
      "scopeName": "viewRes",
      "description": "View registered resources"
    },
    {
      "scopeName": "editRes",
      "description": "Edit registered resources"
    },
    {
      "scopeName": "delRes",
      "description": "Delete registered resources"
    }
  ],
  "tokenAttributes": [
    {
      "attrName": "sessionId",
      "attrValue": "$sessionId",
      "attrType": "DYNAMIC"
    },
    {
      "attrName": "resSrvAttr",
      "attrValue": "RESOURCECONST",
      "attrType": "STATIC"
    }
  ],
  "idDomain": "MTLSDomain",
  "audienceClaim": {
    "subjects": [
      "user"
    ]
  }
}'

```

3. Create the Client:

```

curl --location --request POST 'https://<admin-host>:<admin-
port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/client' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--data '{
  "attributes": [
    {
      "attrName": "staticAttr",
      "attrValue": "CustomValue",
      "attrType": "static"
    }
  ],
  "secret": "welcome1",
  "id": "MTLSClient",

```

```

"scopes": [
  "MTLSResource.viewRes",
  "MTLSResource.editRes",
  "MTLSResource.delRes"
],
"clientType": "CONFIDENTIAL_CLIENT",
"idDomain": "MTLSDomain",
"description": "OAuth Client Description",
"name": "MTLSClient",
"grantTypes": [
  "PASSWORD",
  "CLIENT_CREDENTIALS",
  "JWT_BEARER",
  "REFRESH_TOKEN",
  "AUTHORIZATION_CODE",
  "IMPLICIT"
],
"defaultScope": "MTLSResource.viewRes",
"redirectURIs": [
  {
    "url": "{redirect_URL}",
    "isHttps": false
  }
],
"tokenEndpointAuthMethod": "tls_client_auth",
"issueTLSClientCertificateBoundAccessTokens": "true",
"tlsClientAuthSubjectDN": "C=IN, ST=KA, L=BLR, O=Oracle, OU=OAM,
CN=client2, EMAILADDRESS=client2@oracle.com"
}'

```

4. Add the CA certificate to the OAM truststore:

```

curl --location --request POST 'https://<admin-host>:<admin-port>/oam/
services/rest/ssa/api/v1/security/trust/oauthClient/certificate?
certID=MTLSClient' \
--header 'Accept: application/json' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--header 'Content-Type: application/json' \
--data '{
  "publicCert": "<CA_CERT>",
  "id": "MTLSClient"
}'

```

5. The 2-legged flow using mTLS endpoint starts here:

```

curl -k --key <PATH_TO_CLIENT_KEY> --cert <PATH_TO_CLIENT_CERTIFICATE> --
location --request POST 'https://<mTLS_HOST>:<mTLS_PORT>/oauth2/rest/
token' \
--header 'X-OAUTH-IDENTITY-DOMAIN-NAME: MTLSDomain' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=PASSWORD' \
--data-urlencode 'redirect_uri=<REDIRECT_URI>' \
--data-urlencode 'username=weblogic' \

```

```
--data-urlencode 'password=<PASSWORD>' \
--data-urlencode 'client_id=MTLSClient'
```

6. Run the introspect REST API to get details from the token.

```
curl --key <PATH_TO_CLIENT_KEY> --cert <PATH_TO_CLIENT_CERTIFICATE>
--location --request POST 'https://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/
introspect' \
--header 'X-OAUTH-IDENTITY-DOMAIN-NAME: MTLSDomain' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'token=<ACCESS_TOKEN>'
```

Following is the sample response received:

```
{
  "iss": "http://<HOST_NAME>:7777",
  "aud": [
    "MTLSResource",
    "http://<HOST_NAME>:7777"
  ],
  "exp": 1629898603,
  "jti": "gv4Ms01THuOWAE27d3lMLQ",
  "iat": 1629895003,
  "sub": "weblogic",
  "client": "MTLSClient",
  "scope": [
    "MTLSResource.viewRes"
  ],
  "domain": "MTLSDomain",
  "staticAttr": "CustomValue",
  "cnf": {
    "x5t#S256": "yxa_pacafklt_5mqjjc_shtb9qhbvdx0pzsgh3a5nxc"
  },
  "rem_exp": 3409,
  "active": true
}
```

37.13.6 Sample 3-Legged mTLS Authentication Flow

The following example provides a sample 3-legged mTLS authentication flow:

1. Create the Identity Domain

```
curl --location --request POST 'https://<admin-host>:<admin-
port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/
oauthidentitydomain' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--header 'Content-Type: application/json' \
--data '{
  "name": "MTLSDomain",
  "identityProvider": "UserIdentityStore1",
  "description": "MTLSDomain",
```



```

"tokenSettings": [
  {
    "tokenType": "ACCESS_TOKEN",
    "tokenExpiry": 3600,
    "lifeCycleEnabled": false,
    "refreshTokenEnabled": false,
    "refreshTokenExpiry": 86400,
    "refreshTokenLifeCycleEnabled": false
  },
  {
    "tokenType": "AUTHZ_CODE",
    "tokenExpiry": 3600,
    "lifeCycleEnabled": false,
    "refreshTokenEnabled": false,
    "refreshTokenExpiry": 86400,
    "refreshTokenLifeCycleEnabled": false
  },
  {
    "tokenType": "SSO_LINK_TOKEN",
    "tokenExpiry": 3600,
    "lifeCycleEnabled": false,
    "refreshTokenEnabled": false,
    "refreshTokenExpiry": 86400,
    "refreshTokenLifeCycleEnabled": false
  }
],
"errorPageURL": "/oam/pages/error.jsp",
"consentPageURL": "/oam/pages/consent.jsp",
"customAttrs": "{\"domainCertValidityInDays\":\"30\",
\"consentExpiryTimeInMinutes\":\"10\"}",
"tokenEndpointAuthMethodsSupported": [
  "tls_client_auth",
  "self_signed_tls_client_auth"
],
"issueTLSClientCertificateBoundAccessTokens":"true",
"tlsClientAuthSubjectDN":"CN=%CLIENT_ID_PLACEHOLDER%, OU=OAM,
O=Oracle, L=BLR, ST=KA, C=IN"
}'

```

2. Create the Resource

```

curl --location --request POST 'https://<admin-host>:<admin-port>/oam/
services/rest/ssa/api/v1/oauthpolicyadmin/application' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--data '{
  "name": "MTLSResource",
  "description": "Oracle Cloud",
  "scopes": [
    {
      "scopeName": "viewRes",
      "description": "View registered resources"
    },
    {
      "scopeName": "editRes",

```

```

        "description": "Edit  registered resources"
    },
    {
        "scopeName": "delRes",
        "description": "Delete  registered resources"
    }
],
"tokenAttributes": [
    {
        "attrName": "sessionId",
        "attrValue": "$session.id",
        "attrType": "DYNAMIC"
    },
    {
        "attrName": "resSrvAttr",
        "attrValue": "RESOURCECONST",
        "attrType": "STATIC"
    }
],
"idDomain": "MTLSDomain",
"audienceClaim": {
    "subjects": [
        "user"
    ]
}
}'

```

3. Create the Client

```

curl --location --request POST 'https://<admin-host>:<admin-
port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/client' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--data '{
    "attributes": [
        {
            "attrName": "staticAttr",
            "attrValue": "CustomValue",
            "attrType": "static"
        }
    ],
    "secret": "welcome1",
    "id": "MTLSClient",
    "scopes": [
        "MTLSResource.viewRes",
        "MTLSResource.editRes",
        "MTLSResource.delRes"
    ],
    "clientType": "CONFIDENTIAL_CLIENT",
    "idDomain": "MTLSDomain",
    "description": "OAuth Client Description",
    "name": "MTLSClient",
    "grantTypes": [
        "PASSWORD",
        "CLIENT_CREDENTIALS",

```

```

        "JWT_BEARER",
        "REFRESH_TOKEN",
        "AUTHORIZATION_CODE",
        "IMPLICIT"
    ],
    "defaultScope": "MTLSResource.viewRes",
    "redirectURIs": [
        {
            "url": "{redirect_URL}",
            "isHttps": false
        }
    ],
    "tokenEndpointAuthMethod": "tls_client_auth",
    "issueTLSClientCertificateBoundAccessTokens": "true",
    "tlsClientAuthSubjectDN": "C=IN, ST=KA, L=BLR, O=Oracle, OU=OAM,
CN=client2, EMAILADDRESS=client2@oracle.com"
}'

```

4. Add the CA certificate to the OAM truststore

```

curl --location --request POST 'https://<admin-host>:<admin-port>/oam/
services/rest/ssa/api/v1/security/trust/oauthClient/certificate?
certID=MTLSClient' \
--header 'Accept: application/json' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--header 'Content-Type: application/json' \

--data '{
    "publicCert": "<CA_CERT>",
    "id": "MTLSClient"
}'

```

5. The 3-legged flow starts here: Open a browser, and run the following from the browser:

```

https://<OHS_Host>:<OHS_Port>/oauth2/rest/authorize?
response_type=code&domain=MTLSDomain&client_id=MTLSClient&scope=MTLSResour
ce.viewRes%20openid&state=code1234&redirect_uri=<redirect_URL>&nonce=""

```

Login using your user name and password credentials. In the next Consent page, click Allow to get the authentication code from the page URL.

6. Use the authentication code from the previous step to fetch the access token

```

curl --key <PATH_TO_CLIENT_KEY> --cert <PATH_TO_CLIENT_CERTIFICATE> --
location --request POST 'https://<mTLS_HOST>:<mTLS_PORT>/oauth2/rest/
token' \
--header 'X-OAUTH-IDENTITY-DOMAIN-NAME: MTLSDomain' \
--header 'Authorization: Basic dGVzdDp0ZXN0=' \
--data-urlencode 'grant_type=AUTHORIZATION_CODE' \
--data-urlencode 'code=<CODE>' \
--data-urlencode 'redirect_uri=<redirect_URL>' \
--data-urlencode 'client_id=MTLSClient'

```

7. Run the introspect REST API to get details from the token.

```
curl --key <PATH_TO_CLIENT_KEY> --cert <PATH_TO_CLIENT_CERTIFICATE>
--location --request POST 'https://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/
introspect' \
--header 'X-OAUTH-IDENTITY-DOMAIN-NAME: MTLSDomain' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'token=<ACCESS_TOKEN>'
```

Following is the sample response received:

```
{
  "iss": "http://<HOST_NAME>:7777",
  "aud": [
    "MTLSResource",
    "MTLSClient",
    "http://<HOST_NAME>:7777",
  ],
  "exp": 1629894002,
  "jti": "cuWu5Mzae9Hn00cBUErzKw",
  "iat": 1629890402,
  "sub": "weblogic",
  "client": "MTLSClient",
  "scope": [
    "MTLSResource.viewRes",
    "openid"
  ],
  "domain": "MTLSDomain",
  "grant": "AUTHORIZATION_CODE",
  "sessionId": "5e63e6ba-7eca-43fa-b264-52f4e26aecbd=",
  "staticAttr": "CustomValue",
  "nonce": "%22%22",
  "resSrvAttr": "RESOURCECONST",
  "cnf": {
    "x5t#S256": "yxa_pacafklt_5mqjjc_shtb9qhbvdx0pzigth3a5nxc"
  },
  "rem_exp": 3523,
  "active": true
}
```

37.14 Proof Key for Code Exchange (PKCE) Support in OAM

This section provides details about the Proof Key for Code Exchange (PKCE) enhancement.

In OAuth 2.0, 3-legged flow, public clients that use the `authorization_code`, `client_id`, and `client_secret` parameters for requesting access token from the authorization server are vulnerable to interception attacks. Once the attacker gains access to the authorization code and the `client_secret`, it can also retrieve the access token, therefore compromising the entire security of the 3-legged flow.

To prevent this, OAM provides Proof Key for Code Exchange (PKCE) support for OAuth 2.0 authorization code grant flow.

In a typical PKCE flow, the clients use a temporary one-time dynamic credential called `code_verifier` instead of the static `client_secret`. A SHA256 hashed string is generated from the `code_verifier` called the `code_challenge`, which is then used to request the Authorization Code. The access token is then requested from the authorization server by sending the authorization code along with the `code_verifier` in the request. The authorization server (OAM) hashes the `code_verifier` and compares it with the previously received `code_challenge`. Only if these two values match, the access token is issued, thereby enhancing the security of the authorization code grant flow.

 **Note:**

OAM validates `code_challenge` for Base64 with padding. To fix this and ensure RFC compliant behavior for the `code_challenge` (Base64 without padding as described in <https://tools.ietf.org/html/rfc7636#appendix-A>), you must download and apply the OAM Patch **32406872**. For details, see the note [How To Enable OAuth Proof Key For Code Exchange \(PKCE\) in Oracle Access Manager \(OAM\) 12.2.1.4 \(Doc ID 2755209.1\)](#) at <https://support.oracle.com>. OAM will support only the RFC compliant behavior in all the subsequent releases.

See the following sections for more details about enabling PKCE and generating access token through PKCE flow.

- [Enabling PKCE](#)
- [PKCE Flow for Access Token Generation](#)

37.14.1 Enabling PKCE

If you need enhanced security with the 3-legged OAuth 2.0 code grant flow, you can enable PKCE by setting `UsePKCE` either at the domain or client level.

- The `UsePKCE` values are case-sensitive.
- The domain level values are applicable to all the clients under that specific domain. If you need to apply PKCE for a specific client, you can enable PKCE by setting the `UsePKCE` parameter only for that client.
- The client level values take precedence over the domain level values.

Enabling PKCE at Domain Level

At domain level, you can enable PKCE by adding `UsePKCE` in `customAttrs` and setting it to one of the following values:

Values	Behavior and Description
ALL_CLIENTS_TYPES	<p>PKCE is enabled for all client types.</p> <p>If the PKCE parameters <code>code_verifier</code> and <code>code_challenge</code> are not provided in the <code>authorization_code</code> and <code>token_access request</code>, the authorization flow switches to the non-PKCE 3-legged OAuth flow and completes.</p>
ALL_CLIENTS_TYPES_STRICT	<p>PKCE is enabled for all client types.</p> <p>If the PKCE parameters <code>code_verifier</code> and <code>code_challenge</code> are not provided in the <code>authorization_code</code> and <code>token_access request</code>, the authorization fails.</p>
PUBLIC_CLIENTS	<p>PKCE is enabled for <code>PUBLIC_CLIENTS</code> only.</p> <p>If the PKCE parameters <code>code_verifier</code> and <code>code_challenge</code> are not provided in the <code>authorization_code</code> and <code>token_access request</code>, the authorization flow switches to the non-PKCE 3-legged OAuth flow and completes.</p> <p>For non-public clients, that PKCE parameters are ignored if used.</p>
PUBLIC_CLIENTS_STRICT	<p>PKCE is enabled for <code>PUBLIC_CLIENTS</code> only.</p> <p>If the PKCE parameters <code>code_verifier</code> and <code>code_challenge</code> are not provided in the <code>authorization_code</code> and <code>token_access request</code>, the authorization fails.</p>

The following example shows a sample request to enable PKCE when creating the domain. To update an existing domain with PKCE parameters use the `PUT` method.

See [Identity Domain REST Endpoints](#) for the REST API documentation.

```
curl --request POST '<AdminServer>:<AdminPort>/oam/services/
rest/ssa/api/v1/oauthpolicyadmin/oauthidentitydomain' \
--header 'Authorization: Basic d2VibG9naWM6d2VibG9naWMx' \
```

```
--header 'Content-Type: application/json' \
--header 'Cookie:
JSESSIONID=OweshB5ejqWvx6wDrFbHD1sc67WsGbEWu4sBp3aeZ3Ki7kLPLEY7!1934427792' \
--data-raw '{
  "name": "DemoDomain",
  "identityProvider": "UserIdentityStore1",
  "description": "Test Domain",
  "tokenSettings": [
    {
      "tokenType": "ACCESS_TOKEN",
      "tokenExpiry": 3600,
      "lifeCycleEnabled": false,
      "refreshTokenEnabled": false,
      "refreshTokenExpiry": 86400,
      "refreshTokenLifeCycleEnabled": false
    },
    {
      "tokenType": "AUTHZ_CODE",
      "tokenExpiry": 3600,
      "lifeCycleEnabled": false,
      "refreshTokenEnabled": false,
      "refreshTokenExpiry": 86400,
      "refreshTokenLifeCycleEnabled": false
    },
    {
      "tokenType": "SSO_LINK_TOKEN",
      "tokenExpiry": 3600,
      "lifeCycleEnabled": false,
      "refreshTokenEnabled": false,
      "refreshTokenExpiry": 86400,
      "refreshTokenLifeCycleEnabled": false
    }
  ],
  "errorPageURL": "/oam/pages/servererror.jsp",
  "consentPageURL": "/oam/pages/consent.jsp",
  "customAttrs": "{\"usePKCE\": \"ALL_CLIENTS_TYPES\"}"
}'
```

Enabling PKCE at Client Level

At client level, you can enable PKCE by setting the `UsePKCE` parameter to one of the following values:

Values	Behavior and Description
STRICT	If the PKCE parameters <code>code_verifier</code> and <code>code_challenge</code> are not provided in the <code>authorization_code</code> and <code>token_access</code> request, the authorization fails.
NON_STRICT	If the PKCE parameters <code>code_verifier</code> and <code>code_challenge</code> are not provided in the <code>authorization_code</code> and <code>token_access</code> request, the authorization flow switches to the non-PKCE 3-legged OAuth flow and completes.

The following example shows a sample request to enable PKCE when creating the client. To update an existing client with the PKCE parameters use the `PUT` method.

See [Client REST Endpoints](#) for the REST API documentation.

```
curl --request POST '<AdminServer>:<AdminPort>/oam/services/
rest/ssa/api/v1/oauthpolicyadmin/client' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic d2VibG9naWM6d2VibG9naWMx' \
--header 'Cookie:
JSESSIONID=OweshB5ejqWvx6wDrFbHD1sc67WsGbEWu4sBp3aeZ3Ki7kLPLEY7!
1934427792' \
--data-raw '{
  "attributes": [
    {
      "attrName": "customeAttr1",
      "attrValue": "CustomValue",
      "attrType": "static"
    }
  ],
  "secret": "<client_secret>",
  "id": "DemoClientId",
  "scopes": [
    "DemoResServer.scope1"
  ],
  "clientType": "PUBLIC_CLIENT",
  "idDomain": "DemoDomain",
  "description": "Client Description",
  "name": "DemoClient",
  "grantTypes": [
    "PASSWORD",
    "CLIENT_CREDENTIALS",
    "JWT_BEARER",
    "REFRESH_TOKEN",
    "AUTHORIZATION_CODE"
  ],
  "defaultScope": "DemoResServer.scope1",
  "usePKCE": "NON_STRICT",
  "redirectURIs": [
    {
      "url": "http://localhost:8080/Sample.jsp",
      "isHttps": true
    }
  ]
}'
```

37.14.2 PKCE Flow for Access Token Generation

This section provides the PKCE flow details for Access Token generation.

1. Create or update your Identity Domain by adding `UsePKCE` in `customAttrs`. For details, see [Enabling PKCE at Domain Level](#)
2. If the resource server already exists, skip this step. If not, add a resource server as described in the section [Creating a Resource](#).

3. Create or update the OAuth Client by adding the `UsePKCE` parameter. For details, see [Enabling PKCE at Client Level](#)
4. If the OAuth 3-legged flow is already setup, skip this step. If not, perform the manual side steps on OAM server and WebGate as described in the section **3 - Legged Flows — process** under [Runtime REST APIs for OAuth 12c](#)
5. Generate a cryptographically random key called `code_verifier`.
`code_verifier` refers to a cryptographically random URL safe string generated using the characters A-Z, a-z, 0-9, and the punctuation characters `-._~` (hyphen, period, underscore, and tilde). The string must be between 43 and 128 characters long.

 **Note:**

A unique `code_verifier` must be created for each authorization request.

For example,

```
Y1lgBSx8gRsruplrdpGiG-9Lkv~kna1q2pgwXY7UYKc~~jTgMkmUlZkZkapJGT6X.m12.ZUBDj24
qWuPGHl21x3vyFEC.m_XDH_JTw4Qk6_a62Qw~e0sVp3I-AHYhTzn
```

This `code_verifier` will be used to request the Access Token later.

6. Generate `code_challenge` string from the `code_verifier`.
`code_challenge` refers to the transformed BASE64-URL-encoded string of the SHA256 hash of the `code_verifier`.

For example, for **code_verifier**:

```
Y1lgBSx8gRsruplrdpGiG-9Lkv~kna1q2pgwXY7UYKc~~jTgMkmUlZkZkapJGT6X.m12.ZUBDj24
qWuPGHl21x3vyFEC.m_XDH_JTw4Qk6_a62Qw~e0sVp3I-AHYhTzn, the transformed SHA256
hashed string, code_challenge, is Ec-YfJRRibqf_myiWqObZfT-MlHthBUTygBH73zEHbc.
```

7. Through the browser, request the Authorization Code. Send the `code_challenge` along with `code_challenge_method` in the `authorization_code` request to the authorization server (OAM).

`code_challenge_method` value can be one of the following:

- Plain: Sets the `code_verifier` as `code_challenge` instead of the hashed value.
- S256 (Recommended): Indicates the hashed value of the `code_verifier` is set as `code_challenge`.

 **Note:**

A unique Authorization Code can be exchanged for a single Access Token only. Authorization Code replay is rejected by OAM server.

Authorization Code Sample Request

```
http://<OHS Hostname>:<OHS Port>/oauth2/rest/authz?response_type=code
&client_id=TestClient&domain=TestDomain&scope=TestResourceServer.scopel&st
ate=xyz
&redirect_uri=http://localhost:8080/Sample.jsp
```

```
&code_challenge=Ec-YfJRRibqf_myiWqObZfT-M1HthBUTygBH73zEHbc
&code_challenge_method=S256
```

Authorization Code Sample Response

```
http://localhost:8080/Sample.jsp?
code=U1F6aVY3eFVIemNTVWhVcWlMm0o1UT09fkJ5b0J4M1RVdFhwVFYySXhKR1RUS1N
nOWV4UE1oUGRKMfYxR3RmbGt
FYjJYK3lMUUxDRGxyZXNH2VrRXRtQVp1S25MYVY1YXVkbFFGNmx5MUtCcXFZYWJhR1h
pRENIcjI0Vm1YZjNtRE1hTzFDL1lmbEhKK0wvK2pBeUdTZWNYbEMza2dBSnMQ0ZrZ1R
NNmQ3SUVuSmZJNEXScEI4ZUJySVpMT1hEZ1p5alp5eHVxSkd2R1lCdVFqCfPEQTJEaCt
5Z1JxN21ZWFQyc3JKQi9EY0JrTDhiUVB1b01aNkFzdVhJL1JURjBpWFVvK2VPelVxaVo
zSjBMMjJncmZlcEhxMzh1N0hkZ0pqaEZZZmgyRWJHakxMnlBeE9nV1RxRXdxXhaazd
CdW5mQ3VORFkvVHBLUXRvNEJpUz1Eby9vcXlXaEJnZ08vMjExbE5NSFhHUVJFTHkrTlp
4aGNWUzBVmF2cFVJcWpUd3ZHVjBEZmlaOVBQbWI5b3FhT3gvUFpXdVdrYjczRDJBeUw
4RVJjdjMyQ2NnTEZRMXV0aDE2enVOYm9aYVNVb1pjOTJsrGltGJxRWYrOTZtbXlVN1l
SNkxRWEVqd2ovTVBpaHc2TzRrbFZXMyrM2kyaGxuYmhoNFdOeENSL1NvLysrTDBXQkt
UQVhrb2Y0NzF6WTY1VXhIcFg0bG56REoxcURjTFFlSnNMUUE1aVntMwtocDlzYW1CRFd
0YWRpcEhsT1JzK0hydDk4K0pFb0ZaSDYwaWJ1Q1ZMZGc0dzJ2ekh2ZWPWLU1ldnBIUWd
jRV1BeDVAENmOVg4dloyb0hhV0J6ekMyMExCQTdDnFudyt1ZVV6Nkp3cWZON1N6VGJ
0
UnA3UGk2U1JablVBK0k1bTduR1haOWwzVklQeHRKWWIwNHVObXY5NkFxn2tUMEMwb0dj
b2FlK1c0NGFTekFvKzdlbFdfbnFuZz09&state=xyz
```

- Request the access token from the authorization server (OAM) using the `authorization_code` and the `code_verifier`.

Access Token Sample Request

```
curl --request POST 'http://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token' \
--header 'X-OAUTH-IDENTITY-DOMAIN-NAME: DemoDomain' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Cookie:
JSESSIONID=OweshB5ejqWvx6wDrFbHD1sc67WsGbEWu4sBp3aeZ3Ki7kLPLEY7!
1934427792' \
--data-urlencode 'grant_type=AUTHORIZATION_CODE' \
--data-urlencode
'code=U1F6aVY3eFVIemNTVWhVcWlMm0o1UT09fkJ5b0J4M1RVdFhwVFYySXhKR1RUS1
NnOWV4UE1oUGRKMfYxR3RmbGt
FYjJYK3lMUUxDRGxyZXNH2VrRXRtQVp1S25MYVY1YXVkbFFGNmx5MUtCcXFZYWJhR1h
pRENIcjI0Vm1YZjNtRE1hTzFDL1lmbEhKK0wvK2pBe
UdTZWNYbEMza2dBSnMQ0ZrZ1RNNmQ3SUVuSmZJNEXScEI4ZUJySVpMT1hEZ1p5alp5e
HVxSkd2R1lCdVFqCfPEQTJEaCt5Z1JxN21ZWFQyc3J
KQi9EY0JrTDhiUVB1b01aNkFzdVhJL1JURjBpWFVvK2VPelVxaVozSjBMMjJncmZlcEh
xMzh1N0hkZ0pqaEZZZmgyRWJHakxMnlBeE9nV1RxR
XdxXhaazdCdW5mQ3VORFkvVHBLUXRvNEJpUz1Eby9vcXlXaEJnZ08vMjExbE5NSFhHU
VJFTHkrTlp4aGNWUzBVmF2cFVJcWpUd3ZHVjBEZml
aOVBQbWI5b3FhT3gvUFpXdVdrYjczRDJBeUw4RVJjdjMyQ2NnTEZRMXV0aDE2enVOYm9
aYVNVb1pjOTJsrGltGJxRWYrOTZtbXlVN1lSNkxRW
EVqd2ovTVBpaHc2TzRrbFZXMyrM2kyaGxuYmhoNFdOeENSL1NvLysrTDBXQktUQVhrb
2Y0NzF6WTY1VXhIcFg0bG56REoxcURjTFFlSnNMUUE1
aVntMwtocDlzYW1CRFd0YWRpcEhsT1JzK0hydDk4K0pFb0ZaSDYwaWJ1Q1ZMZGc0dzJ2
ekh2ZWPWLU1ldnBIUWdjRV1BeDVAENmOVg4dloyb0h
```


37.15 Token Exchange Support in OAM

This section provides details about the token exchange support in OAM.

Token Exchange provides a mechanism to an OAuth client to exchange a token for impersonation or delegation use cases. For more information, see [RFC 8693](#).

37.15.1 Enabling Token Exchange Support

To enable token exchange support several identity domain custom attributes, resource server token attributes and client grant_type TOKEN_EXCHANGE must be used.

The Identity domain custom attributes to be used are:

Name	Description	Default
TokenExchange_DelegationEnabled, TokenExchange_ImpersonationEnabled	set to true to enable delegation/impersonation	false
TokenExchange_TokenExpirationSeconds	Expiry of issued_token	Expiry is set to tokenExpiry value for ACCESS_TOKEN (Existing setting)
TokenExchange_DelegationClaimsToBeCopied, TokenExchange_ImpersonationClaimsToBeCopied	' ' separated value to indicate the list of claims to be copied from subject_token to issued_token. For example, mygroups,name	null (No claims apart from sub is copied from subject_token)
TokenExchange_EnforceMayActClaim	If may_act claim is present in the subject_token, sub claim from may_act claim will be matched with sub claim from actor_token. Set to false to disable the check.	true
TokenExchange_OAMIssuers	' ' separated list to indicate the list of issuers to be considered as OAM issuers. Consents are evaluated for OAM tokens only. By default, tokens are considered to be issued by OAM instance if the issuer field matches the OAM server LBR configured. If the issuer is different, this property can be used. If the property is set to ALL_ISSUERS, all tokens are considered as issued by the same OAM instance. This is a ' ' separated property. Can be needed in MDC environments with consent management enabled.	None

The resource server token attributes to be used are:

Name	Description	Default
TokenExchange_AllowedClientsForDelegation, TokenExchange_AllowedClientsForImpersonation	';' separated value to indicate the list of clientids allowed for creating an issued_token with the resource server scopes	null(No client is allowed)



Note:

The scopes expected in the issued token must be assigned to the client.

Client must have the following grant_type:

Name	Value	Default
grant_type	TOKEN_EXCHANGE	Null

Following are the request and response parameters to be used to generate exchange token.

Table 37-7 Request Parameters

Name	Requirement	Support type/values
grant_type	REQUIRED	TOKEN_EXCHANGE or urn:ietf:params:oauth:grant-type:token-exchange
resource	OPTIONAL if scope parameter is provided present. Else, REQUIRED.	ResourceServerName configured in OAM. Multiple parameters can be specified as defined at https://datatracker.ietf.org/doc/html/draft-ietf-oauth-resource-indicators-08
audience	OPTIONAL	Any string. Multiple parameters can be specified as defined at https://datatracker.ietf.org/doc/html/draft-ietf-oauth-resource-indicators-08
scope	OPTIONAL if resource parameter is provided. Else, REQUIRED	Space-separated scope values.
requested_token_type	OPTIONAL	urn:ietf:params:oauth:token-type:jwt
subject_token	REQUIRED	JWS token
subject_token_type	REQUIRED	urn:ietf:params:oauth:token-type:jwt
actor_token	OPTIONAL	JWS token
actor_token_type	OPTIONAL, REQUIRED only if actor_token is passed	urn:ietf:params:oauth:token-type:jwt

Table 37-8 Response Parameters

Name	Supported type/values
access_token	JWS token
issued_token_type	urn:ietf:params:oauth:token-type:jwt
token_type	Bearer
expires_in	Validity in seconds
scope	space separated scopes

CURL Commands

Following are a few example CURL commands to create or modify the identity domain, target resource server, OAuth Client, delegation request, and impersonation request.

1. Create identity domain with TokenExchange attributes

```
curl --location --request POST 'http://oamadminhost:oamadminport/oam/
services/rest/ssa/api/v1/oauthpolicyadmin/oauthidentitydomain' \
--header 'Authorization: Basic YWRtaW46cGFzc3dvcmQ=' \
--header 'Content-Type: application/json' \
--data '{
  "name": "CompanyDomain",
  "identityProvider": "oid",
  "description": "Updated Domain",
  "tokenSettings": [
    {
      "tokenType": "ACCESS_TOKEN",
      "tokenExpiry": 3600,
      "lifeCycleEnabled": false,
      "refreshTokenEnabled": true,
      "refreshTokenExpiry": 86400,
      "refreshTokenLifeCycleEnabled": false
    },
    {
      "tokenType": "AUTHZ_CODE",
      "tokenExpiry": 3600,
      "lifeCycleEnabled": false,
      "refreshTokenEnabled": true,
      "refreshTokenExpiry": 86400,
      "refreshTokenLifeCycleEnabled": false
    }
  ],
  "errorPageURL": "/oam/pages/servererror.jsp",
  "consentPageURL": "/oam/pages/consent.jsp",
  "keyPairRolloverDurationInHours": "24",
  "customAttrs":
  "{\"TokenExchange_DelegationEnabled\": \"true\", \"TokenExchange_DelegationClai
msToBeCopied\": \"mygroups\", \"TokenExchange_TokenExpiryInSeconds\": \"300\"
}" \
}'
```

2. Create a target resource server with clientid allowed for delegation (Re-create a resource server if an update to an existing client id is needed)

```
curl --location --request POST 'http://oamadminhost:oamadminport/oam/
services/rest/ssa/api/v1/oauthpolicyadmin/application' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic YWRtaW46cGFzc3dvcmQ=' \
```

```

--data '{
  "name": "PublishService_ResourceServer",
  "description": "Resource server for publish service",
  "scopes": [
    {
      "scopeName": "publish",
      "description": "Publish the user profile"
    },
    {
      "scopeName": "backup",
      "description": "Backup the user profile"
    }
  ],
  "tokenAttributes": [
    {
      "attrName": "TokenExchange_AllowedClientsForDelegation",
      "attrValue": "PublishServiceClientId",
      "attrType": "static"
    }
  ],
  "idDomain": "CompanyDomain",
  "audienceClaim": {
    "subjects": [
      "http://abc.publishservice.com"
    ]
  }
} }'
```

3. Create a OAuth Client with TOKEN_EXCHANGE grant type

```

curl --location --request POST 'http://oamadminhost:oamadminport/oam/services/
rest/ssa/api/v1/oauthpolicyadmin/client' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic YWRtaW46cGFzc3dvcnQ=' \
--data '{
  "attributes": [
  ],
  "secret": "password",
  "id": "PublishServiceClientId",
  "scopes": [
    "PublishService_ResourceServer.publish"
  ],
  "clientType": "CONFIDENTIAL_CLIENT",
  "idDomain": "CompanyDomain",
  "description": "Publish service client",
  "name": "PublishServiceClientName",
  "grantTypes": [
    "CLIENT_CREDENTIALS",
    "TOKEN_EXCHANGE"
  ],
  "defaultScope": "PublishService_ResourceServer.publish",
  "redirectURIs": [
    {
      "url": "https://oauthdebugger.com/debug",
      "isHttps": true
    }
  ]
}
```

4. The TOKEN_EXCHANGE endpoint is:

```
http://<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/exchange
```

5. Delegation request

```
curl --location --request POST 'http://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/exchange' \
--header 'X-OAUTH-IDENTITY-DOMAIN-NAME: CompanyDomain' \
--header 'Authorization: Basic UHVibGlzaFNlcnZpY2VDbGllbnRjZDp3ZWxjb211MQ==' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'charset: UTF-8' \
--data-urlencode 'grant_type=TOKEN_EXCHANGE' \
--data-urlencode
'subject_token=eyJraWQiOiJDb2lwYW55RG9tYWluIiwieDV0IjoiSuliLTdDcnBxaDBGY1VseD
VNekJZLWVXUUVViiWYxNjoiUlMyNTYifQ.eyJpc3MiOiJodHRwOi8vYWJiaGF0b2QtaWRtc2Uu
cmVmMi5zdXNlbnmdkZXlzcG1zZS5yZWcyLnNlc2VuZ2RldjFwaHhub3JhY2xldmNuLmNvbToxND
EwMC9vYXV0aDIiLCJodHRwOi8vYWJjLlVzZXJQcm9maWxlQXBwbGljYXRpb24uY29tIiwiaXNlcl
Byb2ZpbGVhbnRlbnRlc2VzZXNvdXJzZVNlcnZlciJdLCJleHAiOiJlbnRlc2VzZXNvdXJzZVN
lc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci51cGRhdGU
iXSwiZG9tYWluIjoiQ29tcGFueURvbWVpbiIsImdyYW50IjoiQVVUSE9SVpB
VELPTl9DT0RFIiwibXlncm91cHMlOiJvcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcn
Rlc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIi
wifQ.eyJpc3MiOiJodHRwOi8vYWJiaGF0b2QtaWRtc2UuY29tIiwiaXNlcnRlc2VzZXNvdXJzZ
VNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZX
NvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcn
Rlc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIi
wifQ.fV9aW9a1aocYdRvxZN_XX2AGfEuHz6aCnsc84wmV0dmq7MJhDHY0u70a7_yKjgN2
ZNPap_atSYDf3yo48gdDS4-60UgqUeXLgyrJyo-
eNBG1QEkkQfZxm_lRdAWGaeAk0c5Wzk6Yj2NHmMBEFQ8cHx8tysz-
CW21C42dy24g0rwwgSek_9GXj7Nlf2smZZge9asIogCGDQPIUsPdTFuGmrc0NX7u82kMf12bZ
IdnlpBy9216tRRnlaOLuoE025uhZSury36ArYDGKumI4PXv5I07RBFc5VbrxfgnvCaMlzkwcBgCb
_ei8BM7vz9TgOycKbHeg7-AiSchVbY2s8ldEA' \
--data-urlencode 'subject_token_type=urn:ietf:params:oauth:token-type:jwt' \
' \
--data-urlencode
'actor_token=eyJraWQiOiJDb2lwYW55RG9tYWluIiwieDV0IjoiSuliLTdDcnBxaDBGY1VseD
VNekJZLWVXUUVViiWYxNjoiUlMyNTYifQ.eyJpc3MiOiJodHRwOi8vYWJiaGF0b2QtaWRtc2Uu
cmVmMi5zdXNlbnmdkZXlzcG1zZS5yZWcyLnNlc2VuZ2RldjFwaHhub3JhY2xldmNuLmNvbToxND
EwMC9vYXV0aDIiLCJodHRwOi8vYWJjLlVzZXJQcm9maWxlQXBwbGljYXRpb24uY29tIiwiaXNlcl
Byb2ZpbGVhbnRlbnRlc2VzZXNvdXJzZVNlcnZlciJdLCJleHAiOiJlbnRlc2VzZXNvdXJzZVN
lc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIi
wifQ.eyJpc3MiOiJodHRwOi8vYWJiaGF0b2QtaWRtc2UuY29tIiwiaXNlcnRlc2VzZXNvdXJzZ
VNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZX
NvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcn
Rlc2VzZXNvdXJzZVNlcnZlci5yZWZkIiwiaXNlcnRlc2VzZXNvdXJzZVNlcnZlci5yZWZkIi
wifQ.yU60VR9Yyk5ukc4KdfFVDBAympu35s00nZG4kz-uh4f5rm_R94fajS3aJaZvA_d7v-
axisKwngGthBWjhwalIPJcgtS3lpGSjQfsxShl9Or67aFQ1ZbTi9uHPm8ehUot9FKbrYhmcVzaj_M
0pld4Scoqoc6_eZ4iaU3Jf4RiO_Jr-BLd5t6dfcdzbcUi-Q' \
--data-urlencode 'actor_token_type=urn:ietf:params:oauth:token-type:jwt' \
' \
--data-urlencode 'scope=PublishService_ResourceServer.publish'
```

6. Impersonation request

```
curl --location --request POST 'http://
<ManagedServerHost>:<ManagedServerPort>/oauth2/rest/token/exchange' \
--header 'X-OAUTH-IDENTITY-DOMAIN-NAME: CompanyDomain' \
--header 'Authorization: Basic UHVibGlzaFNlcnZpY2VDbGllbnRjZDp3ZWxjb211MQ==' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'charset: UTF-8' \
--data-urlencode 'grant_type=TOKEN_EXCHANGE' \
--data-urlencode
```


Understanding OpenIDConnect

OpenIDConnect implements authentication as an extension to the OAuth 2.0 authorization process. Use of this extension is requested by Clients by including the openid scope value in the Authorization Request.

**Note:**

OpenIDConnect with Detached Credential Collector (DCC) is not supported. Also see, [Overview of Access Manager Credential Collection](#).

OpenIDConnect provides information about the end-user in the form of an id_token. This token verifies the identity of the user and provides basic profile information about the end-user. OpenIDConnect is enabled by default as a part of the OAM 12cBP01 installation process.

This section describes the following topics:

- [About OpenIDConnect Tokens](#)
- [Claims](#)
- [OpenIDConnect Authentication Flows in Oracle Access Manager](#)

See Also:

- http://openid.net/specs/openid-connect-core-1_0.html

38.1 About OpenIDConnect Tokens

OpenIDConnect generates a token namely, OpenIDConnect ID Token.

In addition to OAuth Access and refresh tokens, OpenIDConnect considers an identity token (ID Token). The primary extension that OpenIDConnect makes to OAuth 2.0 to enable End-Users to be authenticated is the ID Token data structure.

See

- [About OAuth Tokens](#)

38.1.1 OpenIDConnect ID Token

An ID Token is a security token that contains Claims about the Authentication of an End-User by an Authorization Server when using a Client, and potentially other requested Claims.

The Claims of an ID Token include subject, issuer, audience, and timestamps. The following table lists out the Claims used within the ID Token for all OAuth 2.0 flows used by OpenIDConnect:

Table 38-1 Claims within the ID Token used by OpenIDConnect

Field	Description	Type	Required/Optional
iss	Issuer Identifier for the Issuer of the response.	String	Required
sub	Subject Identifier.	String	Required
aud	Audience that this ID Token is intended for.	String	Required
exp	Expiration time on or after which the ID Token MUST NOT be accepted for processing.	String	Required
iat	Time at which the JWT was issued.	String	Required
auth_time	Time when the End-User authentication occurred.	String	Required
Nonce	Value used to associate a Client session with an ID Token, and to mitigate replay attacks.	String (case-sensitive)	Required
acr	Value of the authentication level.	String (case-sensitive)	Optional
amr	Identifier for a family of closely related authentication methods. See Authentication Method Reference Values .	an array of case sensitive strings	Optional
azp	Identifier of the party that is intended to use the access token and to request resources.	String (case-sensitive)	Optional
sid	Value of the encrypted token that contains session identifier and details.	String (case-sensitive)	Optional

 **Note:**

nonce value is passed through unmodified from the Authentication Request to the ID Token. If present in the ID Token, Clients MUST verify that the *nonce* Claim Value is equal to the value of the *nonce* parameter sent in the Authentication Request. If present in the Authentication Request, Authorization Servers MUST include a *nonce* Claim in the ID Token with the Claim Value being the *nonce* value sent in the Authentication Request. Authorization Servers SHOULD perform no other processing on *nonce* values used.

Sample of a set of claims in an ID Token:

```
{
  "iss": "http://host1:14100/oauth2",
  "sub": "weblogic",
  "aud": ["MDCClient19", "http://host1:14100/oauth2"],
  "exp": 1509626702,
  "iat": 1509623102,
  "auth_time": "1509623099159",
  "jti": "_UC4Ew-NUTYQsMOXCoMo0g",
  "at_hash": "5CnkOBb_Mk28GYJlhC_Srg",
  "azp": "MDCClient19",
  "acr": "2",
  "sid": "gO5pDtJFt+7bH/YQC8QpUQ==~teJ01stvBcUXT8xXcmaIG1ppGMAMBLKqPuJUKnzLyX3spmDtWwgDm/
  qj5hhoyPhSiqAghOgFmE+kpsm8esEEsbZht+L5dkL27JUSUbAGBBmw1R/8Q1xLTE0cEoNJ+9aJ",
  "amr": ["pwd"]
}
```

38.2 Claims

The Client obtains Claims about the end-user and the authentication event. Standard claims can be requested to be returned either in the userinfo response or in the ID token.

Following are some of the standard claims used by OpenIDConnect. For a complete list of standard claims supported, see https://openid.net/specs/openid-connect-basic-1_0.html#StandardClaims

Table 38-2 Claims used by OpenIDConnect

Field	Description	Type	Required/Optional
sub	Identifier for the end-user at the issuer.	string	Required
name	Full name of the end-user in displayable form including all name parts, possibly including titles and suffixes, ordered according to the end-user's locale and preferences.	string	Required

Table 38-2 (Cont.) Claims used by OpenIDConnect

Field	Description	Type	Required/Optional
given_name	Given name(s) or first name(s) of the end-user. Separate multiple given names using space characters, as applicable for different cultures.	string	Required
family_name	Surname(s) or last name(s) of the end-user. Separate multiple family names using space characters, as applicable for different cultures.	string	Required
preferred_username	Shorthand name by which the end-user wishes to be referred to at the Client, such as janedoe or j.doe.	string	Required
email	Preferred e-mail address of the end-user.	string	Required
email_verified	True if the end-user's e-mail address has been verified. False if the end-user's e-mail address has been verified.	boolean	Required
gender	Gender of the end-user, male or female or other.	string	Required
Locale	Location of the end-user.	String	Required
phone_number	Preferred telephone number of the end-user.	String	Required
phone_number_verified	True if the end-user's phone number has been verified. False if the end-user's phone number has been verified.	boolean	Required

Table 38-2 (Cont.) Claims used by OpenIDConnect

Field	Description	Type	Required/Optional
address	Preferred postal address of the end-user.	JSON object	Required
updated_at	Time the end-user's information was last updated. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.	number	Required

38.3 Custom Claims



This content applies only to OAM Bundle Patch 12.2.1.4.210920 and later releases.

This section describes the following topics:

Topics

- [About Custom Claims](#)
- [Defining Custom Claims Using Templates](#)
- [Using Claims Parameter in Authorization Request](#)
- [Examples for Template-Based Claims](#)

38.3.1 About Custom Claims

OAM extends the ability to define the custom claims using templates that can be configured at client or domain level. The custom claims can be included in all the access tokens, ID tokens and userinfo. You can also perform value transformation as well as value filtering of the custom claim.

Custom claims are defined using templates and the usage details can be configured at client or domain level



Note:

If the custom claims are defined at both the client and domain level, the claims at the client level takes precedence.

You can specify if the custom claim name needs to be included in one or all of the following: access tokens, ID tokens, and userinfo response, for a specific client or all the clients under a specific domain.

You can set the following attributes while creating client or domain:

Access Tokens	"accessTokenCustomClaims": ["<Claim_Name_For_AccessToken>"]
ID Token	"idTokenCustomClaims": ["<Claim_Name_For_IdToken>"]
UserInfo	"userInfoCustomClaims": ["<Claim_Name_For_UserInfo>"]

For example, when creating an identity domain, if you specify a custom claim named "ClientIP" for access tokens ("accessTokenCustomClaims":["ClientIP"]) then all the access tokens that are created using any of the clients under that identity domain will include the ClientIP custom claim.

 **Note:**

You must define the Custom Claim before using it in the Client or Domain request. For details, see [Defining Custom Claims Using Templates](#)

38.3.2 Defining Custom Claims Using Templates

Administrators must define the custom claim that needs to be used in the client or the domain request.

Use the `<admin-host>:<admin-port>/oam/services/rest/ssa/api/v1/template/{name}` REST API to create a custom claim. The API must include the name of the custom claim that needs to be created. For example, to create a custom claim named ClientIP, you must use `<admin-host>:<admin-port>/oam/services/rest/ssa/api/v1/template/ClientIP`

 **Note:**

The following names are not supported for creating templates. Their value cannot be overwritten.

- OAuth registered claim names. Refer to [rfc7519#section-4.1](#)
- IDToken claims. Refer to https://openid.net/specs/openid-connect-basic-1_0.html#IDToken
- sid
- address

The custom claim needs to be defined by the following attributes:

Table 38-3 Attributes and Values for Custom Claim Definition

Custom Claim Attributes	Value	Description
name	<CustomClaim_Name>	Name of the custom claim. For example ClientIP
valueMapping	<\${namespace.var_name}>	Claim value can be mapped to any of the following: <ul style="list-style-type: none"> static attribute dynamic attribute from the UserStore, Session, or Requests using \$ For example, a claim can be mapped to dynamic value using UserStore attribute using \$user.attr.<attributeName> For more information about dynamic value mapping, see Namespace and Variable Names for Policy Responses
defaultValue	<default_Claim_Value>	Optional. This default value is used if the template processing fails. For more information, see About defaultValue
transformFirst	false or true	Optional. By default the value is false If the value is set to true then valueTransformation is applied before valueFiltering
description		Optional. Details related to this template.
dynamicParams		Optional. Parameters must be defined as list of entries in dynamicParams For details, see About dynamicParams
valueFiltering	Java string methods that return boolean.	Optional. Filter the values of the claim. The value filters must be defined using Java methods whose return type is boolean. For example, startsWith, contains, endsWith, and so on. For more information, see About valueFiltering Also refer to Class String in <i>Java™ Platform, Standard Edition 8 API Specification</i>

Table 38-3 (Cont.) Attributes and Values for Custom Claim Definition

Custom Claim Attributes	Value	Description
valueTransformation	Java string transformation methods	String transformations on the custom claim value. The transformation operations must be defined using Java methods. Optional. For example, <code>concat</code> , <code>replace</code> , <code>split</code> , and so on For more information, see About valueTransformation Also refer to Class String in <i>Java™ Platform, Standard Edition 8 API Specification</i>

For the full list of attributes supported for defining custom claims, see the [REST API Documentation](#).

The custom claims JSON must be similar to the following structure:

```
{
  "valueMapping": "value of claim",
  "defaultValue": "defaultValueForClaim",
  "transformFirst": "true or false",
  "description": "string defining the template",
  "dynamicParams": [
    "$user.attr.userStoreattributeName"
  ],
  "valueFiltering": {
    "populateIf": "java string method with boolean return type",
    "params": [
      "param3",
      "param4"
    ],
    "type": [
      "param3_type",
      "param4_type"
    ]
  },
  "valueTransformation": [
    {
      "operation": "java String method name",
      "params": [
        "param1",
        "param2"
      ],
      "type": [
        "param1_type",
        "param2_type"
      ]
    },
    {
      "operation": "java String method name",
```

```
        "params": [
            "param1",
            "param2"
        ],
        "type": [
            "param1_type",
            "param2_type"
        ]
    }
]
}
```

38.3.2.1 About valueTransformation

You can use `valueTransformation` to transform the values of the claim.

You can perform string transformations on the custom claim value. The transformation operations must be defined using Java methods.

Refer to [Class String](#) in *Java™ Platform, Standard Edition 8 API Specification*

Examples

The following examples illustrate the use of `valueTransformation` on custom claims.

Example 38-1

In this example, the claim value `sampleText` is transformed to `sampleData`

```
{
  "valueMapping": "sampleText",
  "valueTransformation": [
    {
      "operation": "replace",
      "params": [
        "Text",
        "Data"
      ],
      "type": [
        "CharSequence",
        "CharSequence"
      ]
    }
  ]
}
```

Example 38-2

You need not specify the `type`, if all parameters are of type string. For example:

```
{
  "valueMapping": "sampleText",
  "valueTransformation": [
    {
      "operation": "replaceFirst",
```

```

        "params": [
            "param1",
            "param2"
        ]
    }
]
}

```

Example 38-3

You can chain transformations, if the result of each transformation is also a string. The result of the first transformation is taken as the input for the next one.

In the following example, the claim value `sampleText` is transformed to `SAMPLETEXT.STRING1.STRING2`

```

{
  "valueMapping": "sampleText",
  "valueTransformation": [
    {
      "operation": "concat",
      "params": [
        "string1"
      ]
    },
    {
      "operation": "concat",
      "params": [
        "string2"
      ]
    },
    {
      "operation": "toUpperCase"
    }
  ]
}

```

Example 38-4

In the following example, the claim value `sampleText1:sampleText2` is transformed to `["sampleText1", "sampleText2"]`

**Note:**

The `"operation": "toUpperCase"` is not applied because the result of first operation (`split`) is not a string.

```

{
  "valueMapping": "sampleText1:sampleText2",
  "valueTransformation": [
    {
      "operation": "split",

```

```
        "params": [
            ":"
        ]
    },
    {
        "operation": "toUpperCase"
    }
]
}
```

38.3.2.2 About valueFiltering

You can use `valueFiltering` to filter the value/values of the claim.

The value filters must be defined using Java methods whose return type is boolean.

Refer to [Class String](#) in *Java™ Platform, Standard Edition 8 API Specification*.

Note:

A filter can contain either `populateIf` tag or `populateIfNot` tag, but not both.

- `populateIf`: returns claim value only if the result of this filter java method is true.
- `populateIfNot`: returns claim value only if the result of this filter java method is false.

Examples

The following examples illustrate the use of `valueFiltering` on custom claims.

Example 38-5

In the following example, the filtered value returns `null` as it does not match the `populateIf` filter

```
{
  "valueMapping": "sampleText",
  "valueFiltering": {
    "populateIf": "contains",
    "params": [
      "orcl"
    ],
    "type": [
      "CharSequence"
    ]
  }
}
```

Example 38-6

You need not specify the `type`, if all parameters are of type string. For example:

```
{
  "valueMapping": "sampleText",
  "valueFiltering": {
    "populateIf": "endsWith",
    "params": [
      "Text"
    ]
  }
}
```

38.3.2.3 About `defaultValue`

If there is an error while processing the template, the claim value is set to `null` and the claim is not returned back. You can choose to default to a particular claim value in case of error by setting the `defaultValue`.

Example 38-7

In this example, the claim value after processing the template is `defaultSampleText`.

In this case, invalid filter is defined as `"populateIf": "invalidMethodName"`. Therefore, the claim is assigned the configured default value and none of the further processing like transforms is applied.

```
{
  "valueMapping": "sampleText",
  "defaultValue": "defaultSampleText",
  "valueFiltering": {
    "populateIf": "invalidMethodName",
    "params": [
      "ingsss"
    ]
  },
  "valueTransformation": [
    {
      "operation": "toUpperCase"
    }
  ]
}
```

38.3.2.4 About `dynamicParams`

You can use dynamic parameters while defining `valueTransformation` and `valueFiltering`, in addition to the static parameters.

To use dynamic parameters, parameters must be defined as list of entries in `dynamicParams`

Example 38-8

In the following example, claim value is `sampleText` and the `dynamicParams` is defined as `"$user.attr.email" : email.com`. Therefore, the claim value returned after processing the template is `sampleTextemail.com`

```
{
  "valueMapping": "sampleText",
  "dynamicParams": [
    "$user.attr.email"
  ],
  "valueTransformation": [
    {
      "operation": "concat",
      "params": [
        "$user.attr.email"
      ]
    }
  ]
}
```

38.3.3 Using Claims Parameter in Authorization Request

You can add the `claims` parameter in the Authorization Request to request for individual claims.

```
http://<admin-host>:<admin-port>/oauth2/rest/authz?
response_type=<>&
client_id=<>&
domain=<>&
state=<>&
redirect_uri=<>&
scope=<>&
claims=New parameter
```

The `claims` Authentication Request parameter requests for specific Claims to be returned from the `UserInfo` endpoint and/or in the ID Token.


It is represented as a JSON object containing lists of Claims.

claims JSON Support Keys	Description
<code>userinfo</code>	Optional. Requests that the listed individual Claims be returned from the <code>UserInfo</code> Endpoint in addition to other claims that are being requested through scope or client/domain configuration.
<code>id_token</code>	Optional. Requests that the listed individual Claims be returned in the ID Token in addition to other claims that are being requested through scope or client/domain configuration.

The values for these keys must be one of the following:

claims JSON Supported Values	Description	Sample
null	Indicates that this Claim is being requested in the default manner	<pre>claims = { "userinfo": { "nickname": null, "picture": null, "customClaim": null }, "id_token": { "customClaim1": null, "email": null } }</pre>

claims JSON Supported Values	Description	Sample
JSON Object	Optional. Indicates whether the Claim being requested is an Essential Claim. If the value is <code>true</code> , this indicates that the Claim is an Essential Claim.	<pre>claims = { "userinfo": { "given_name": { "essential": true }, "email": { "essential": true }, "email_verified": { "essential": true }, "id_token": { "customClaim1": { "essential": true }, "email": { "essential": true } } } }</pre>

 **N**
O
t
e
:
O
n
l
y
e
s
s
e
n
t
i
a
l
i
s
s
u
p
p
o
r
t
e
d
.

For details, refer to https://openid.net/specs/openid-connect-core-1_0.html#ClaimsParameter

38.3.4 Examples for Template-Based Claims

This section provides examples that illustrate value transformations, value filtering, and creating custom claims in AccessToken, IDToken, and UserInfo.

Example 38-9 Transforming Claim Value by Composing Different User Store Attributes

In this example, a custom e-mail is composed using `<users given name>.<users family name>@<users domain name>.com`. The values are as follows:

- claim name is `CustomEmail`
- `$user.attr.given_name: "user"`
- `$user.attr.family_name: "lastname"`
- `$user.attr.domain_name: "domainName"`

The claim value after processing the template is: `user.lastname@domainName.com`



Note:

A variable-length argument must be defined as `Array`, for example, `CharSequence...` must be defined as `CharSequence[]`.

```
{
  "valueMapping": "$user.attr.given_name",
  "transformFirst": "true",
  "dynamicParams": [
    "$user.attr.family_name",
    "$user.attr.domain_name"
  ],
  "valueTransformation": [
    {
      "operation": "join",
      "params": [
        ".",
        "$user.attr.given_name",
        "$user.attr.family_name"
      ],
      "type": [
        "CharSequence",
        "CharSequence[]"
      ]
    },
    {
      "operation": "concat",
      "params": [
        "@"
      ]
    },
    {
      "operation": "concat",
      "params": [
        "$user.attr.domain_name"
      ]
    },
    {
      "operation": "concat",
```

```

        "params": [
            ".com"
        ]
    }
]
}

```

Example 38-10 Transforming Users Group Allocation to Array and Filter Groups based on a Regular Expression

In this example, users group allocation is transformed from colon-separated string to an array and a list of group values that belong to the admin set are returned, that is, 'admin' or 'Admin'

The values are as follows:

- claim name is Groups
- `$user.groups`: "HR:Finance:Admin:Manager:HRAdmin:Testadmin"

The claim value after processing the template is: ["Admin", "HRAdmin", "Testadmin"]

```

{
  "valueMapping": "$user.groups",
  "transformFirst": true,
  "valueTransformation": [
    {
      "operation": "split",
      "params": [
        ":"
      ]
    }
  ],
  "valueFiltering": {
    "populateIf": "matches",
    "params": [
      ".*[aA](dmin).*"
    ]
  }
}

```

Example 38-11 Mapping an OIDC Standard Claim in Userinfo to LDAP Attributes

In this example, `email_verified` reads from LDAP attribute named `emailVerified`

To achieve this,

1. Create a mapping using a template. The template name must match the claim name required.

```

{
  "valueMapping": "$user.attr.emailVerified"
}

```

2. Update the client or domain to read template-based claims

```
{
  "userInfoCustomClaims": [
    "email_verified"
  ]
}
```

Note:

Alternatively, you can also specify this per request in the `claims` parameter in the Authorization request. For details, [Using Claims Parameter in Authorization Request](#)

Example 38-12 Applying transformations and Filters to OIDC Standard Claim

In this example, the claim `website` is filtered and the transformations are being applied. That is, only if it starts with `https`, it is converted to lower case.

To achieve this,

1. Create mapping using a template, template name must match claim name required.

```
{
  "valueMapping": "$user.attr.website",
  "valueFiltering": {
    "populateIf": "startsWith",
    "params": [
      "https"
    ]
  },
  "valueTransformation": [
    {
      "operation": "toLowerCase"
    }
  ]
}
```

2. Update the client or domain to read template-based claims. Alternatively, request this claim as part of the authorization request claims parameter, for example:

```
http://<admin-host>:<admin-port>/oauth2/rest/authz?
response_type=<>&
client_id=<>&
domain=<>&
state=<>&
redirect_uri=<>&
scope=openid&
claims={"userinfo":{"website": null}}
```

Example 38-13 Adding Custom Claims in AccessToken , IDToken and UserInfo

To achieve this,

1. Create mapping using a template. Template name must match claim name required.

```
{
  "valueMapping": "customValue"
}
```

2. Update client or domain to specify the destination of custom claim

```
{
  "accessTokenCustomClaims": [
    "customClaim_accessToken"
  ],
  "idTokenCustomClaims": [
    "customClaim_idToken"
  ],
  "userInfoCustomClaims": [
    "customClaim_userInfo"
  ]
}
```

Example 38-14 Adding Custom Claims as Client Configurations and Request Parameters

To achieve this,

1. Create two templates, for example, `customClaim1` and `customClaim2`. Template name must match claim name required.
2. Update client or domain to specify the destination of custom claim

```
{
  "idTokenCustomClaims": [
    "customClaim1"
  ]
}
```

3. Request `customClaim2` as part of the authorization request claims parameter

```
http://<admin-host>:<admin-port>/oauth2/rest/authz?
response_type=<>&
client_id=<>&
domain=<>&
state=<>&
redirect_uri=<>&
scope=openid&
claims={"id_token":{"customClaim2": null}}
```

The `idToken` will include both the claims: `customClaim1` from client configuration and `customClaim2` from the request parameter.

38.4 OpenIDConnect Authentication Flows in Oracle Access Manager

OpenIDConnect server performs authentication to login as user or to verify the login status of the user and securely sends the result to the Client.

The authentication flows determine how the ID Token and Access Token are returned to the Client. Before processing a request, the client applications need to have an existing registration with the OAuth and OpenIDConnect servers. See [Clients](#) and [Creating a Client](#) .

Oracle Access Manager supports OpenIDConnect with the following 3-legged authentication flows:

- Authorization Code Grant Flow
- Implicit Grant Flow

The important parameters used in the curl command for OpenIDConnect Authentication Flows are:

Table 38-4 Parameters used in the curl command for OpenIDConnect Authentication Flows

Field	Description	Type	Required/Optional
client_id	Id of the client making the request.	string	Required
scope	OpenIDConnect requests MUST contain the "openid" scope value. If not present but other scopes are present, it is treated as a normal OAuth request flow where the scope parameter is not mandatory.	string	Required
redirect_uri	Redirect URI registered with the client, to which the response will be sent.	string	Required
response_type	Indicating the type of request (Valid values - code, token, id_token, token id_token)	string	Required

Table 38-4 (Cont.) Parameters used in the curl command for OpenIDConnect Authentication Flows

Field	Description	Type	Required/Optional
state	Not mandatory but recommended to maintain state between the request and callback.	string	Recommended
nonce	The value used to associate a Client session with an ID Token, and to mitigate replay attacks.	String	Required for implicit flows

See [OpenIDConnect Core 1.0](#) .

This section describes the following topics:

- [Understanding Authorization Code Grant Authentication Flow](#)
- [Understanding Implicit Grant Authentication Flow](#)

38.4.1 Understanding Authorization Code Grant Authentication Flow

When using the Authorization Code Grant Flow, the `response_type` parameter is set to `code` and all tokens are returned from the Token Endpoint. In this authentication flow, the `authZcode` is returned to the client. With the `authZcode`, the client makes a request to the token endpoint and receives the access and identity tokens.

The Authorization Code Grant Authentication Flow, at a high level, is described as follows:

1. Client prepares an authentication request containing the desired request parameters such as `scope` and `response_type` and sends it to the OAM authorization server.
2. OAM authorization server authenticates the user and obtains user's consent to access protected resources.
3. The OAM server sends the Authorization Code (`authZcode`) to the Client.
4. Client sends the `authZcode` to the Token Endpoint and exchanges it for an ID Token and an Access Token directly. The response body contains both the tokens.

 **Note:**

Note: For non-OpenIDConnect flows, only the Access Token is returned.

5. Client validates the ID token and retrieves the User's Subject Identifier.

 **Note:**

Identity token is returned only for the OpenIDConnect flow i.e., when the request contains "openid" keyword in the scope parameter. When the keyword is missing in the scope, it is considered as a normal OAuth - Authorization Code Flow that returns only the access token.

Mandatory parameters in a request include the following:

- `client_id`: Id of the client making the request
- `scope`: OpenIDConnect requests must contain the keyword `openid` in the scope parameter. When `openid` scope is not present and other scopes are defined, then the authentication flow is treated as a pure OAuth request flow.
 - **Consider a scenario where the scopes requested are not registered with the client.** In a 3-legged flow, when a client requests for valid scopes that were not registered with the client, `authZcode` is created if the user gives the consent. For example, a Resource Server has three scopes: `scopeA`, `scopeB`, and `scopeC` where only `scopeA` is registered with the client. During an Authorization code or Implicit grant authentication flow, the client requests for `scopeB` and `scopeC`. The consent page is displayed to the user as the scopes requested are valid but not registered with the client. Upon obtaining user's consent, the code is created for or the token is generated in Implicit flows.
 - **Consider a scenario where no scope is passed in the request:** When no scope is passed in the request, the default scope registered with the client is used to generate the `AuthZcode` and Access token eventually. Thus is a pure OAuth flow where the `scope` parameter is not mandatory.
- `redirect_uri`: Redirect URI registered with the client, to which the response will be sent.
- `response_type`: Indicating the type of request; valid values include `code`, `token`, `id_token`, and `token id_token`
- `state`: Not mandatory but recommended to maintain state between the request and callback.

For an Authorization Code Grant Authentication flow, the server returns the Access token where the `response_type` parameter value is `code` as shown in the following table:

Table 38-5 Authorization Code Grant Authentication flow: Parameters and Access tokens

Parameter value	Tokens returned	Sample Request (with openid scope)	Sample Response
response_type=code	Access Token Identity Token	http://host2:2222/oauth2/rest/authorize?response_type=code&domain=JANEDOEIDDOMAIN1&client_id=JANEDOEAOAUTHIDCLIENT1&scope=RSer0.searchsong openid profile&state=code1234&redirect_uri=http://host2:19528/app1/pages/Sample.jsp	http://host2:19528/app1/pages/Sample.jsp?code=NHN3WEtWazhjdGI0ZXhRcVh5bWtt09fnYwcitKZ3ZDdFc3aXJoUVJISVR0S3IWRIVrbjFKM0JFSzZLOGICZW13RkNBRzVUbzN1L3NOZUJLTVVFc3E1KzJnZlhVZGFXRNXJcEhtSkJNTI9BWWJhZ2VWUjZDZFB5SmtIM254VnRKC UdadVIGSjdzUzN4VktGTmFLVFhZYWQxR0NvYzVla2M4ajJaN1haWUFzZHBKekpzL0pEa3JjcXZkdzEybnlVZlBrRkhsZy8vTE14VXpMQUoxcThhaXRROU5VMURhNHkrRW9WeFE2bFhLWmR0NmFyTUVTLzlsTkZtQ1B6YTNoeGtFWHJNR2FUV05BaWZJZXFKSk9nNzdTeFZpME9zVi92dWFRalRhbHNRWWkwQnBZbFBHRmppNjJhU1R0S1NBaCtZaDBENzRXMFJuQ0R4dW1SQTlKbTVWZjhZRzQ0WGVnSE0yemc2U0JxbzIzQWwxd3lNNnFFS2FpLzZvaXJjSUdwZFI3anhXSDI2b09lZkJaamNHY1ZCTIYxOUcwY3JKUWE5RFVidDF0VG1PRDBWN DdOceFzZWFMbkFvRFc2QjZGUFFpd0drenEvSERjUVBpYWhoSUxHM1BOTW5CRFiyT2hzY3FJbzRvbWUxN0pONEVTR2ZiZC9sNFiyVUd2QXBPNFpWNVMxdGYrUEQ2cFJFLzZrRmVhYWk0cTV4T09jQnF1VytES2U4N0d2OXNpdIJPV3hUOFREajNSekFqeU5nWWNLc2Vktm51elBOeHlsYjAwQ3pTeWFNVVQ5R0toNWJCR2NYNGwzZkZUNIFJTUIHYVJDd3lKT1dITFhYRElqS3pTK29uZUxsSUdGZDByUUtHSmFrZGVNbGhmQktNT0hrcW8yY0xhNFQveTk4TW0zeXZOemNyc20zWnlIbWZOUHFuTFU4K2ZjPQ==&state=code1234

Table 38-5 (Cont.) Authorization Code Grant Authentication flow: Parameters and Access tokens

Parameter value	Tokens returned	Sample Request (with openid scope)	Sample Response
authzCode (sample snippet for reference)	NA	NA	<pre>http://localhost:8080/Sample.jsp? code=NDNIVnBOQ2MxWIZ4MHRmTW dHSm85QT09fnJBakNGU3FOOC9MZ HFyK0N3dDNDZG1uVWY4Sm05WXN EdDRRbE11cEEwQU0rb2lvVEpqS3Zj WDBnZSs0V1hka0NnYnhIRjlnKzNtekJI NysvMmlPekRpS09nOUFqbEVsUEFiR mY1VG4rbmFPbDg5OGtXc2hTdHITRU V5azM4U1k0NExHbWNKd0Fwa3YxZk w0VVZFdxBWS1dFWEtLWUR2T2FrTy t5VzNzdHhDaFp1b0NCOHBBWDFFN XZBZ1B4dEpEN2xmbm9vRzdIR0JpZE pid1ZvamR5NzFUvYtKVzZuaEE3WFIP b3NIUS9BK2pqVHo2OUNuWGdxajRM MWFwU1V6M256NG1RS3JoK3E5RG VDRDd6THJ5NW15U3RIWE14eDh1YU 9oUHJwSGpkVzJnPQ==&state=code1 234</pre>
access_token (sample snippet for reference)			<pre>"eyJraWQiOiJPSURDRG9tYWluliwiED V0ljoibndBTXM0cXVVZDV3emdKUTZS OTZaekRrRTYwliwiYWxnIjoiUIMyNTYif Q.eyJpc3MiOiJodHRwOi8vZGVuMDF0 cHMudXMub3JhY2xlcmlhbnVvbnVzNzc4Mi 9vYXV0aDliLCJhdWQiOiJPSURDU2Vy dmVyaWwiZXhwIjoxNTA5NjQwMjcwMjcwMjcw dGkiOiJUEFrcFlabVVtSjVuMlAyV0JfR DZBlwiaWF0IjoxNTA5NjM2NjcwMjcwMjcw WliOiJhYXJhdGhpliwIy2xpZW50IjoiT0l EQ0NsaWVudDliLCJzY29wZSI6WyJPSURDU2 VydmVyaWwIbnVzNjBIMSI6WyJRE NTZXJ2ZiJub3BlbnkIiwI0IEQ1NlcnZl ci5waG9uZSI6WyJRENTZXJ2ZiJub3BlbnkIiw RkcmVzcyJldLCJkb21haW4iOiJPSURD RG9tYWluliwiOiJ1IiwiaWF0IjoiMTUxOTI5 ODk5IiwiaXNjaWkiOiJ1IiwiaWF0IjoiMTUx OTI5ODk5IiwiaWF0IjoiMTUxOTI5ODk5Iiw I-DOS1- Yj7Pj6aTSXzCUUVkDTKdVzJb8e8BcjF 9FYiea8mE7zw9ulHtMO- gceSiVHQ4tc3sOYJ7_vek2gUfGj7urS_ h3oHQqGazWqQIQOHRxbrY7xbprj58 pFqiJdmVAp1eReYVNIImM7RpXKFRVy UZ43JBGEHONr94tuRC8H- Ss7Y9K_ND59Q0xmOmgznXjrNz7KN 2yWC3rIRuJySmPrzqG0bA0Pp30TAvX GqUpIlFn5H9DN0PnZKLeUWZZRB5G aXL- xWRmbzCS6eW110sMEqEdABVfdoOL A", "token_type": "Bearer", "expires_in": 3600,</pre>

Table 38-5 (Cont.) Authorization Code Grant Authentication flow: Parameters and Access tokens

Parameter value	Tokens returned	Sample Request (with openid scope)	Sample Response
			RDRG9tYWluIn0.ulizk3IEk2aWlc89zD7uf-rlQRz89RNfDGpdy_mrl8dMhSZme69BI-DOS1-Yj7Pj6aTSXzCUUVkDTKdVzJb8e8BcjF9FYiea8mE7zw9ulHtMO-gceSiVHQ4tc3sOYJ7_vek2gUfGj7urS_h3oHQqGazWqQIQOHRxbnYr7xbprj58pFqiJdmVAp1eReYVNIImM7RpXKFRVyUZ43JBGEHONr94tuRC8H-Ss7Y9K_ND59Q0xmOmgoznXjrNz7KN2yWC3rlRuJySmPrzqG0bA0Pp30TAvXGqUpIIFn5H9DN0PnZKLeUWZZRB5GaXL-xWRmbzCS6eW1I0sMEqEdABVfdoolA", "token_type": "Bearer", "expires_in": 3600,
	refresh_token (sample snippet for reference)		"LGHeL2ToSLNRsYN5rmvIQg==~2G6JvVBnwuNrTN/69fDPvzX2RZSVRptEbUF4mGVkf/bjCeeZFfaPEnyoqVu9hB5Gt4BZPMDeEjYQZx1imRdGNh/NOx/MoRt8oXkyMljf1g4Qo67n5RThoyNVNczkevmNUeBznSTjdW2HNOSzHBnv0WY8leimNkN3C77K2wAbSJtGWio8uX388jlrVXEa/5XdNe2LszA4c5nW5UgZzcjAi/69y9azViyEdtKP4ndo8CNS8O4cVAym2pryEr2zKPr+NfW7mwJ7gVeyjCaxcKmQ2zHKGpblaPHxak/iCyN0jew31XursBIRCP9PCKRwBc26qGX2Sf7W8Q3bxsUvImhvYw==",

Table 38-5 (Cont.) Authorization Code Grant Authentication flow: Parameters and Access tokens

Parameter value	Tokens returned	Sample Request (with openid scope)	Sample Response
id_token (sample snippet for reference)			"eyJraWQiOiJPSURDRG9tYWluliwieDV0ljoibndBTXM0cXVZDV3emdkUTZSOTZaekRrRTYwliwiYWxnIjoiUIMyNTYifQ.eyJpc3MiOiJodHRwOi8vZGVuMDF0cHMudXMub3JhY2xiLmNvbToxNzc4Mi9vYXV0aDIiLCJzdWIiOiJhYXJhdGhpliwiYXVkljoiT0lEQ1NlcnZlcilsmV4cCI6MlUwOTY0MDI3MCwiaWF0IjoxNTA5NjM2NjcwLlJhdXRoX3RpbWUiOiJAsImp0aSI6IkJKYTRlU3NhemhLa01Wd0pOT1F1NncifQ.NmUWTa0brKTV54ZlYUq8mxamATPO3xVHILALO22CAgdAYZs0ixKDSxMH-z-7-giNyAu8MXUwXmBfnwpd0ZuphIPVMVtr0YAEh_pzraR_bKSkglqtHOlpswFVa z69YXyFtvmIOU77m-zEndsUbsxuJ0oZ9BErtlqcnkiaQn5os_RKr4uz3ltZtSxzH-vF1nzbmOpHZYETNreji8kqg36gZHWxhFtbAE8hh_8UoNYpDCfxnpt1Du9kbFp2jXdeM9HVwQW_KH7Vv6rW8mChPefw9IAv3nfHxuiwcZ2GLC9NOwSJBVDM A94V1u9KfsXvR_bAlyaVE5zC-rpFEyXEG5cka"

38.4.2 Understanding Implicit Grant Authentication Flow

For an Implicit Grant Authentication Flow, the server returns all the tokens at the same time depending on the `response_type` parameter as follows:

Table 38-6 (Cont.) Implicit Grant Authentication Flow: Parameters and Access tokens

Parameter value	Tokens returned	Sample Request	Sample Response
response_type=id_token	Identity Token	<pre>http://host3:2222/ oauth2/rest/ authorize? response_type=id _token&domain=O IDCDomain&client _id=OIDCClient2& scope=OIDCServe r.scope2&redirect_ uri=http:// localhost:8080/ Sample.jsp&nonce =1234&state=abcd</pre>	<pre>http://localhost:8080/ Sample.jsp#id_token=eyJraWQiOiJPS URDRG9tYWluliwieDV0ljoibndBTXM0c XVVZDV3emdkUTZSOTZaekRrRRTYwli wiYWxnIjoiUIMyNTYifQ.eyJpc3MiOiJod HRwOi8vZGVuMDF0cHMudXMub3JhY 2xLmNvbToxNzc4Mi9vYXV0aDliLCJzd WliOiJhYXJhdGhpliwYXVkljoiT0IEQ1N lcnZlcilslmV4cCI6MTUwOTcyNTI0MCw iaWF0IjoxNTA5NzIxNjQwLCJhdXRoX3 RpbWUiOiAsIm5vbmNlljoiMTIzNCIsIm p0aSI6Ii12RThqMlpHOEhabXZTRnVXc TVPSmcfQ.2qxv4rdrv- YvWNUK31aphrWkQ0ilrxhK690X2FgW W1CSNImUWfYGsNolcHiXxyzwPT6sv 6koJ4m0jOII2HENluWTdbE7SbtNJQ8J SVcg2et6hQJXLUXB1ECJQGctw8FU Ysprg-dtki- gCW85eRH9_V1RJMv2XaHC7PNoHE jG4CfqKmjwubfletYRReXks_uJ9YYW8 TKnk87srG2Hh- uzK3C_GFLqUWaFrXp6XyM1yS-w4N- WyQ7UszuZTv8zd8SzX1xahjldY0nakM EA7My7O3PcpVM4RTh-6xVLR_jNqRG xwbg4jc8QfmoWzMwRrYyjtZBSyGg d0jB3PWp4zn5w&state=abcd</pre>
response_type=token	Access Token	<pre>http://host3:2222/ oauth2/rest/ authorize? response_type=to ken&domain=OID CDomain&client_id =OIDCClient2&sco pe=OIDCServer.sc ope1&state=code1 234&redirect_uri=h ttp:// localhost:8080/ Sample.jsp</pre>	<pre>http://localhost:8080/ Sample.jsp#access_token=eyJraWQi OiJPSURDRG9tYWluliwieDV0ljoibndB TXM0cXVVZDV3emdkUTZSOTZaekRr RTYwliwiYWxnIjoiUIMyNTYifQ.eyJpc3 MiOiJodHRwOi8vZGVuMDF0cHMudX Mub3JhY2xLmNvbToxNzc4Mi9vYXV0a DliLCJhdWQiOiJPSURDU2VydmVylwi ZXhwIjoxNTA5NzA5OTE0LCJqdGkiOiI 5S3BfLUZhT2pBTmxOOFE5S1J2T2d3 liwiaWF0IjoxNTA5NzA2MzE0LCJzdWli OiJhYXJhdGhpliwY2xpZW50ljoiT0IEQ 0NsaWVudDliLCJzY29wZSI6WyJPSU RDU2VydmVylmNjB3BIMSJdLCJkb21h aW4iOiJPSURDRG9tYWluIn0.QcCrRvl BEbK6- rVooKqkSzMP0RofgjeEhr74AbdJGnNQ ewITQ9zry_TY56oWrkkyNKqjRYN0qa1 _kXHoe562M-02L46A2qJSEEaaejrYhu vltmKbur0XIPpx8stUGCBhiv_vzSX_E5 dr2aZ7P9Xz6Eqjm- wMMzZngAKImPmrdMDR3eJLIK2AfX7 DFdcS9g4FIDZC3I8eoJ6sNBjBSVY37q GoT87NUh_d3y4Ga0Ga9Y9n5tHoxrq W1kuvtAAxHbB2IQBvBabSgDIGMGS9 Fp7o0pi6by176XFUg7iYb1UY5Dx1hr0 mlwPOHb3ndBX8Kg6wna_NplvvHrSc7 XTH3DpOYw&token_type=Bearer&stat e=code1234</pre>

**Note:**

When an authorization request is made with an authorization code grant or implicit grant type, and if `openid scope` is not included, the request is handled as a normal OAuth authorization request. An `id_token` is not issued through the response. Only `access_token` and `refresh_token` (in case of authorization code grant flow) are issued

38.4.3 Understanding OpenIDConnect UserInfo Endpoint

The UserInfo endpoint is an OAuth2.0 resource that returns claims or information about the authenticated user.

The access token is generated with all necessary scopes. The client passes this access token to the User endpoint of the server and requests for the claims about the end-user.

OpenIDConnect Clients use `scope` values to specify what access privileges are being requested for Access Tokens. The scopes associated with Access Tokens determine what resources will be available when they are used to access OAuth 2.0 protected endpoints.

OpenIDConnect defines the following scope values that are used to request Claims as shown in the table:

Table 38-7 scope values that are used to request Claims

Scope	Description	Mandatory/optional
profile	If requested, the Access token is generated with this scope. When the access token is sent to the server's UserInfo endpoint, the server responds with specific claims about the authenticated user's profile.	Optional
email	It requests access to the email and email_verified Claims of the user	Optional
address	It requests access to the address Claim of the end user.	Optional
phone	It requests access to the phone_number and phone_number_verified Claims of the end user.	Optional

See Also:

- http://openid.net/specs/openid-connect-core-1_0.html#ScopeClaims
- http://openid.net/specs/openid-connect-core-1_0.html#UserInfo

38.4.3.1 Retrieving Userinfo Attributes from OAM

When LDAP is used, the values for all the scope attributes or claims need to be retrieved from OAM. Configure OAM to get the UserInfo Claims.

Create a new IDStore using any one of the following methods:

- Using IDSPProfile
- OAM IDStore directly



Note:

The attribute or claim in the token are mapped to the physical attribute in the IDStore. The attribute or claims that are returned as part of the userinfo response cannot be modified. However, the values returned as part of these can be modified by editing the attribute in IDStore to Physical attribute mapping.

See Also:

- [Creating an Identity Directory Service Profile](#)
- [Editing or Deleting an Identity Directory Service Profile](#)

The following table captures the claims under each scope and the corresponding backend LDAP attribute.

Table 38-8 Claims under each scope and the corresponding backend LDAP attribute.

Scope	Attributes/ Claims	IDStore Attribute	IDSPProfile's physical attribute	Sample Response
Profile	Name	name	eg: cn	"profile": { "name": " jane", "family_name": " doe", "given_name": "jane" "preferred_username": "j.doe", "gender": "", "locale": "en/US" "updated_at": "1509709292632" }
	family_name	lastname	sn	
	given_name	firstname	givenname	
	given_name	firstname	givenname	
	preferred_username	This is returned as ""	cn	
	gender			
	locale	preferredlanguage		
updated_at	current time			
email	email_verified	"N"		
address	formatted	postaladdress	"address": { "formatted": "Aaccf Amar\$01251 Chestnut Street\$Panama City, DE 50369", "country": "US", "region": "DE", "postal_code": "50369" }	
	country	ss		
	region	country		
	postal_code	state postalcode		

Table 38-8 (Cont.) Claims under each scope and the corresponding backend LDAP attribute.

Scope	Attributes/ Claims	IDStore Attribute	IDSPProfile's physical attribute	Sample Response
phone	phone_number_verified	"N"	"phone": { "phone_number_verified": "false" }	

Following figure shows mapping between the entity attribute to physical attribute for an IDSPProfile:

When the `claim:given_name` is mapped to the `firstname` attribute in the IDStore, and the `firstname` is mapped to the `givenname` physical attribute in LDAP. This attribute to physical attribute can be changed to give the required value. We can change the Physical Attribute column, when `maidenname` should be returned instead of `givenname`.

As shown in the following figure, under `IAMSuite`, there are two resources seeded out-of-the-box for OAuth. While requesting for `UserAttributes`, the resource `/OAuth/UserAssertion` is used to assert the user against the new IDStore. Then, modify the originally created `IdentityDomain` to set the `identityProvider` to the new IDStore created through Admin OAuth APIs or curl commands.

[Access Manager >](#)

IAM Suite Application Domain

Application Domain provides a logical container for resources or sets of resources, and the associated policies that dictate who can access specific protected resources.

Summary **Resources** Authentication Policies Authorization Policies Token Issuance Policies Administration

Use the search tool to find an existing Resource or click the New Resource button to create a new one.

Search

Resource Type: HTTP Query String:

Host Identifier: Authentication Policy:

Resource URL: Authorization Policy:

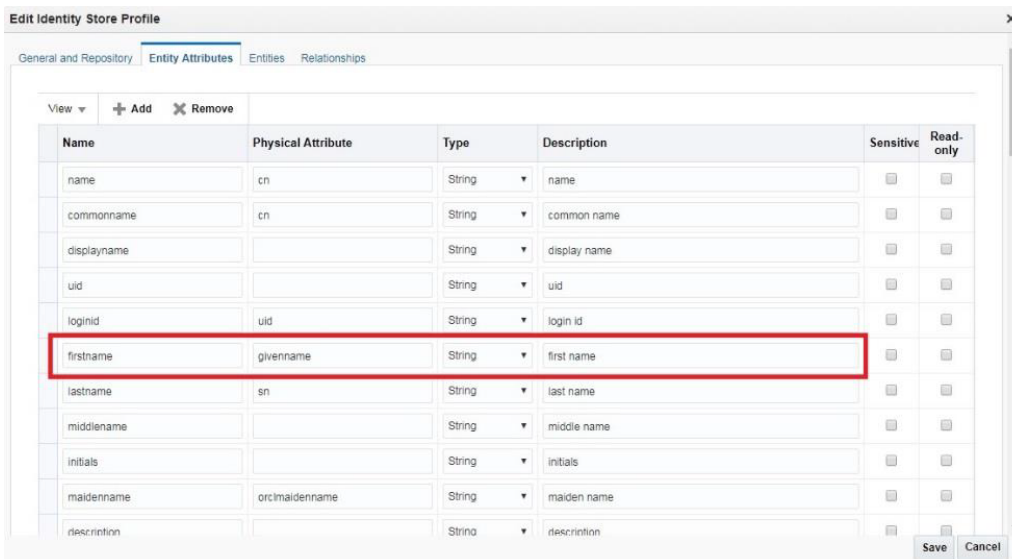
Search Reset

Search Results

Actions View Create Duplicate Edit Delete Detach

Row	Resource Type	Host Identifier	Resource URL	Query String	Authentication Policy	Authorization Policy
1	HTTP	IAMSuiteAgent	/OAuth/UserAssertion		OAuth Assertion Policy	Protected Resource Policy
2	HTTP	IAMSuiteAgent	/OAuth/UserAuthentication		OAuth Authentication Policy	Protected Resource Policy
3	HTTP	IAMSuiteAgent	/ucsr/**			
4	HTTP	IAMSuiteAgent	/reqsvc/**			

If the newly created IDStore is a DefaultStore, no changes required. The resource `/OAuth/UserAssertion` uses OAuth Assertion Policy that has the `LDAPNoPasswordValidationScheme`. Ensure that the scheme uses `LDAPNoPasswordAuthModule` that uses the new IDStore.



[Access Manager >](#)

LDAPNoPasswordAuthModule LDAP Authentication Module

Use the LDAP Authentication module for Basic and Form challenge methods. It matches the credentials (username and password) of the user who requests stored in an LDAP directory server.

* Name

* User Identity Store

If the newly created IDStore is not a DefaultStore, perform the following:

- Using UserIdentificationPlugin, create a new authentication module, UserInfoAuthModule
- Set the KEY_IDENTITY_STORE_REF to point to the new IDStore.
- Create a new AuthnScheme or update LDAPNoPasswordValidationScheme to use this new module.
- Update the OAuth Assertion Policy to point to this new Authentication scheme.

Table 38-9 Parameters to create new authentication module, UserInfoAuthModule

Endpoint	Sample Request	Sample Response
http://{{machine}}:{{mgdport}}/oauth2/rest/userinfo	curl -X GET http://host4:14100/oauth2/rest/userinfo -H 'authorization: Bearer <AccessToken>'	<pre>"guid": "6C9CF210194A11E99FB45DDD0C60B95A", "sub": "weblogic", "family_name": "weblogic", "preferred_username": "weblogic", "updated_at": "1548740667872", "email_verified": false, "phone_number_verified": false</pre>

38.4.3.2 Retrieving User Info Attributes Using Template-Based Mapping

The following claims are read from IDStore, if the IDStore attribute name matches with the OIDC standard claim names. For more information about the standard claims, see https://openid.net/specs/openid-connect-basic-1_0.html#StandardClaims

For few common claims there is default mapping to IDStore Attributes. However, for the claims that do not have a default mapping to IDStore attributes, use template-based mappings. For details, see [Defining Custom Claims Using Templates](#)

Table 38-10 OIDC Standard Claims Mapping

Scope	Attributes/Claims	Mapping for the IDStore Attributes	Sample Response
Profile	name family_name given_name middle_name nickname preferred_username profile picture website gender birthdate zoneinfo locale updated_at	Following common claims also have default mapping to IDStore Attributes: For example: name is mapped to name family_name is mapped to lastname given_name is mapped to firstname preferred_username is mapped to name locale is mapped to preferredlanguage updated_at is mapped to current time	{ "sub": "userSub", "name": "user1", "family_name": "userLastName", "given_name": "userGivenName", "middle_name": "userMiddleName" "nickname": "userNickname", "preferred_username": "userPreferredUsername", "profile": "userProfile", "picture": "userPicture", "website": "userWebsite", "gender": "Female", "birthdate": "2000-01-01", "zoneinfo": "Unknown", "locale": "IN", "updated_at": 1628759535, }

Table 38-10 (Cont.) OIDC Standard Claims Mapping

Scope	Attributes/Claims	Mapping for the IDStore Attributes	Sample Response
email	email email_verified	email is mapped to mail	{ "email": "user@example.co m", "email_verified" : false }
address	"address": { "formatted", "street_address" , "locality", "region", "postal_code", "country" }	formatted is mapped to postaladdress country is mapped to country region is mapped to state postal_code is mapped to postalcode	"address": { "formatted": "Aaccf Amar\$01251 Chestnut Street\$Panama City, DE 50369", "street_address" : "Panama City", "locality": "Panama City", "region": "DE", "postal_code": "50369", "country": "US" }
phone	phone_number phone_number_veri fied	phone_number is mapped to telephone	{ "phone_number": "[+1415555-0100] ", "phone_number_ve rified": false }



Note:

Claim values can be mapped to any physical attribute present in the User Store using Template-based mapping. For more information, see [Examples for Template-Based Claims](#).

38.4.4 Understanding OpenIDConnect Discovery Endpoint

OpenID Provider Issuer discovery is the process of determining the location of the OpenID Provider.

Using the Issuer location discovered, the OpenID Provider's configuration information can be retrieved. The response is a set of Claims about the OpenID Provider's configuration, including all necessary endpoints and public key location information.

38.4.4.1 Configuring OpenIDConnect Discovery Endpoint

The OpenID Provider's configuration details are exposed through `/.well-known/openid-configuration`.

As shown in the following figure, on the OAM console, under IAMSuite domain, OpenID Discovery endpoint is listed as a resource:

15	HTTP	IAMSuiteAgent	/.well-known/**		
16	HTTP	IAMSuiteAgent	/oauth2/rest/**		
17	HTTP	IAMSuiteAgent	/oauth2/rest/approval	OAuth Authentication Policy	Protected Resource Policy
18	HTTP	IAMSuiteAgent	/oam/pages/consent.jsp	OAuth Authentication Policy	Protected Resource Policy
19	HTTP	IAMSuiteAgent	/OAuth/UserAssertion	OAuth Assertion Policy	Protected Resource Policy
20	HTTP	IAMSuiteAgent	/OAuth/UserAuthentication	OAuth Authentication Policy	Protected Resource Policy

Locate `mod_wl_ohs.conf` file at `<ohs_domain_home>/config/fmwconfig/components/OHS/instances/<ohs_instance_name>` and add the following:

```
<Location /.well-known/openid-configuration>
SetHandler weblogic-handler
PathTrim /.well-known
PathPrepend /oauth2/rest
WebLogicHost <OAM_managed_server host>
WebLogicPort <OAM_managed_server port>
</Location>
```

Sample request:

```
GET /.well-known/openid-configuration HTTP/1.1
Host: host4:7777
```

Sample response:

```
{
  "issuer": "http://host4:7777/oauth2",
  "authorization_endpoint": "http://host4:7777/oauth2/rest/authorize",
  "token_endpoint": "http://host4:7777/oauth2/rest/token",
  "userinfo_endpoint": "http://host4:7777/oauth2/rest/userinfo",
  "introspect_endpoint": "http://host4:7777/oauth2/rest/token/info",
  "jwks_uri": "http://host4:7777/oauth2/rest/security",
  "end_session_endpoint": "http://host4:7777/oauth2/rest/userlogout",
  "scopes_supported": [
```

```

        "openid",
        "profile",
        "email",
        "address",
        "phone"
    ],
    "response_types_supported": [
        "code",
        "token",
        "id_token",
        "token id_token"
    ],
    "grant_types_supported": [
        "client_credentials",
        "password",
        "refresh_token",
        "authorization_code",
        "implicit",
        "urn:ietf:params:oauth:grant-type:jwt-bearer"
    ],
    "subject_types_supported": [
        "public"
    ],
    "id_token_signing_alg_values_supported": [
        "RS256"
    ],
    "userinfo_signing_alg_values_supported": [
        "none"
    ],
    "token_endpoint_auth_methods_supported": [
        "client_secret_basic",
        "client_secret_jwt"
    ],
    "token_endpoint_auth_signing_alg_values_supported": [
        "RS256"
    ],
    "claims_supported": [
        "aud",
        "exp",
        "iat",
        "iss",
        "jti",
        "sub"
    ],
    "ui_locales_supported": [
        "en"
    ]
}

```

Configuring `claims_supported`

OAM supports all OIDC standard claims (refer to https://openid.net/specs/openid-connect-basic-1_0.html#StandardClaims), provided they have a value in the configured UserStore or have template-based mappings.

`claims_supported` in the discovery URL can be configured to display all these standard claims by updating the `oam-config.xml`. The customized `claim_supported` list must be added under the following path in `oam-config.xml`:

```

DeployedComponent/Server/NGAMServer/Profile/STS/datastore/backend/
discoveryproviders/metadatadiscovery/claimsSupported

```

See [Updating OAM Configuration](#)

Following is a sample `claim_supported` list:

```
<Setting Name="claimsSupported" Type="htf:map">
  <Setting Name="aud" Type="xsd:string">aud</Setting>
  <Setting Name="exp" Type="xsd:string">exp</Setting>
  <Setting Name="iat" Type="xsd:string">iat</Setting>
  <Setting Name="iss" Type="xsd:string">iss</Setting>
  <Setting Name="jti" Type="xsd:string">jti</Setting>
  <Setting Name="sub" Type="xsd:string">sub</Setting>
  <Setting Name="acr" Type="xsd:string">acr</Setting>
  <Setting Name="name" Type="xsd:string">name</Setting>
  <Setting Name="given_name" Type="xsd:string">given_name</Setting>
  <Setting Name="family_name" Type="xsd:string">family_name</Setting>
  <Setting Name="middle_name" Type="xsd:string">middle_name</Setting>
  <Setting Name="nickname" Type="xsd:string">nickname</Setting>
  <Setting Name="preferred_username"
Type="xsd:string">preferred_username</Setting>
  <Setting Name="profile" Type="xsd:string">profile</Setting>
  <Setting Name="picture" Type="xsd:string">picture</Setting>
  <Setting Name="website" Type="xsd:string">website</Setting>
  <Setting Name="email" Type="xsd:string">email</Setting>
  <Setting Name="email_verified" Type="xsd:string">email_verified</
Setting>
  <Setting Name="gender" Type="xsd:string">gender</Setting>
  <Setting Name="birthdate" Type="xsd:string">birthdate</Setting>
  <Setting Name="zoneinfo" Type="xsd:string">zoneinfo</Setting>
  <Setting Name="locale" Type="xsd:string">locale</Setting>
  <Setting Name="updated_at" Type="xsd:string">updated_at</Setting>
  <Setting Name="address" Type="xsd:string">address</Setting>
  <Setting Name="phone_number" Type="xsd:string">phone_number</Setting>
  <Setting Name="phone_number_verified"
Type="xsd:string">phone_number_verified</Setting>
</Setting>
```

38.4.4.2 Configuring OIDC Discovery Endpoint

In addition to the OpenID Provider's configuration details, another endpoint, `/.well-known/oidc-configuration` is exposed. It provides information related to the access server such as authentication and logout endpoints.

Locate `mod_wl_ohs.conf` file at `<ohs_domain_home>/config/fmwconfig/components/OHS/instances/<ohs_instance_name>` and add the following:

```
<Location /.well-known/oidc-configuration>
SetHandler weblogic-handler
PathTrim /.well-known
PathPrepend /oauth2/rest
WebLogicHost <OAM_managed_server host>
WebLogicPort <OAM_managed_server port>
</Location>
```

Sample request:

```
GET /.well-known/oidc-configuration HTTP/1.1
Host: host4:7777
```

Sample response:

```
{
  "configuration": {
    "release-version": "12.2.1.3.0"
  },
  "access-configuration": {
    "http-direct-authentication-endpoint": "http://host4:7777/oam/server/authentication",
    "http-logout-endpoint": "http://host4:7777/oam/server/logout",
    "http-credential-submit-endpoint": "http://host4:7777/oam/server/auth_cred_submit"
  },
  "openid-configuration": {
    ← Same as openid discovery response value→
  }
}
```



Note:

Currently the back-channel logout through the `end_session_endpoint` is not implemented. The front-channel logout through `http-logout-endpoint` can be used to logout the user and end the session.

38.4.5 Fetching Identity Domain Certificate

Use the security endpoint, `http://<managed server host>:<managed server port>/oauth2/rest/security`, to fetch public certificate of given Identity domain. The output from this endpoint is `<identitydomain>.p7b` file .

Table 38-11 Fetch public certificate of given Identity domain: Parameters

Parameter Type	Parameter Name	Description	Sample
Header	X-OAUTH-IDENTITY-DOMAIN-NAME	Identity Domain Name	MDCDomain19
Header	Authorization	Basic <B64 encoded clientid:password>	Basic TURDQ2xpZW50MTk6d2VsY29tZTE=
Query	identityDomainName	Identity Domain Name Note: First preference will be given to X-OAUTH-IDENTITY-DOMAIN-NAME. Second preference is given to identityDomainName.	MDCDomain18

Following is a sample REST call where X-OAUTH-IDENTITY-DOMAIN-NAME has the preference over identityDomainName.

Sample Request:

```
curl -X GET 'http://host1:14100/oauth2/rest/security?
identityDomainName=MDCDomain18' -H 'authorization: Basic
TURDQ2xpZW50MTk6d2VsY29tZTE=' -H 'x-oauth-identity-domain-name: MDCDomain19'
```

Following is a sample REST call that results in MDCDomain18 public certificate chain in p7b format. i.e. MDCDomain18.p7b file:

Sample Request:

```
curl -X GET 'http://host1:14100/oauth2/rest/security?
identityDomainName=MDCDomain18' -H 'authorization: Basic
TURDQ2xpZW50MTk6d2VsY29tZTE='
```

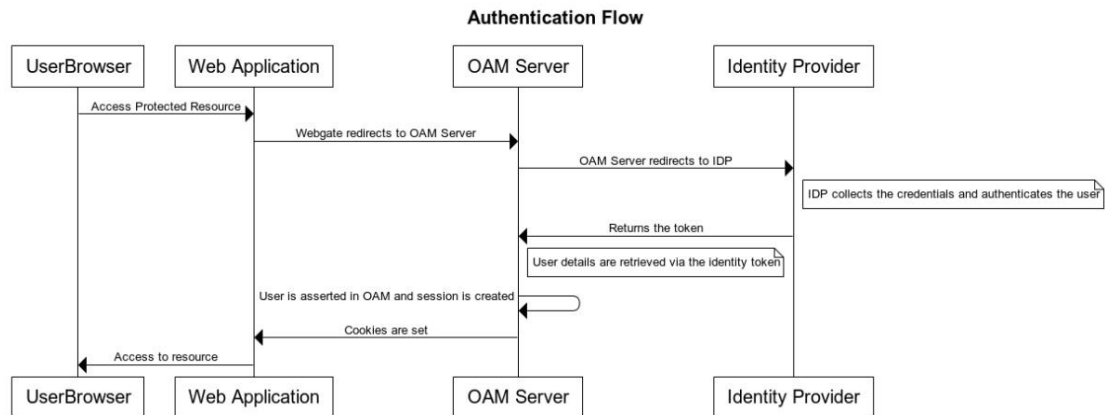
OIDC Client Integrations with Social Identity Providers

OAM ships an out-of-the box OIDC Client Authentication Plugin, `OpenIDConnectPlugin` that enables integration with Social Identity providers such as IDCS, Google and Facebook.

Apart from being an OAuth/OpenIDConnect2.0 Server, Oracle Access Manager (OAM) can also delegate authentication to OpenIDConnect-Social Identity Providers such as IDCS, Google, Facebook or even OAM itself, thus behaving like a relying party (service provider). After authenticating the user, the IDP redirects back to OAM where the user is asserted by OAM and an OAM Session is created.

The authentication process is described as follows:

1. The `OpenIDConnectPlugin` redirects the authentication request to any third-party Identity Provider using OIDC protocol.
2. The third-party Identity Provider (IDP) authenticates the user.
3. The IDP redirects the authentication request back to OAM.
4. The OAM asserts the user.
5. The OAM creates a session.




39.1 About the `OpenIDConnectPlugin`

`OpenIDConnectPlugin` is a generic plugin that you can integrate with any OpenId2.0 providers. This plugin redirects requests to the IDP it is integrated with. After authenticating the user at the IDP, the control is submitted back to the OAM server

The mandatory parameters for plugin configuration include `oauth_client_id`, `oauth_client_secret`, and `provider`.

Table 39-1 OpenIDConnectPlugin: Parameters for plugin configuration

Field	Sample Value	Mandatory / Optional	Description
id_domain		Optional	Enter the Identity domain. It is required for integration with OAM as an IDP since all the artifacts (client) are created under the Identity Domain in OAM.
oauth_client_secret	12312312312asdasdasd1231231sdsadasd	Mandatory	Enter OAuth client secret.
token_endpoint	https://graph.facebook.com/v2.11/oauth/access_token	Optional	Enter the access token endpoint, it is required ONLY if the IDP does not support Discovery URL.
authz_endpoint	https://www.facebook.com/v2.11/dialog/oauth	Optional	Enter the authorization endpoint, (required ONLY if the IDP does not support Discovery URL)
require_proxy		Optional	Set it to <code>false</code> , if the plugin does not use the proxy configuration and redirects directly to the IDP.

 **Note:**

If proxy is required to connect to the IDP, add this setting before starting the server:

```
Dhttp.proxy.Host=www-
proxy.example.com -
Dhttp.proxy.Port=80
```

Table 39-1 (Cont.) OpenIDConnectPlugin: Parameters for plugin configuration



Field	Sample Value	Mandatory / Optional	Description
provider	Facebook or oam or idcs	Mandatory	<p>For this patch release, use only Facebook or oam or idcs as provider values for the following reasons:</p> <ul style="list-style-type: none"> Facebook does not comply to OpenID2.0 specifications For OAM and IDCS, the default email attribute is not supported, must read the sub from the token for this provider. <div data-bbox="1101 646 1377 961" style="border: 1px solid #0070c0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>For other OpenID2.0 compliant providers such as google, you can use any key.</p> </div>
scope		Optional	<p>The default “scope” sent to the IDP is openid email. If this has to be overwritten, we can set the new scope via this plugin parameter.</p>
additional_parameters	attributes=profile,email;protocol=http;namespace=oidc	Optional	<p>Specify the following additional attributes and values. Use semicolon (;) to delimit the attributes and comma (,) to delimit the values.</p> <ul style="list-style-type: none"> attributes = profile, email protocol=http namespace=oidc <div data-bbox="1101 1392 1377 1654" style="border: 1px solid #0070c0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>By default, https is used if the protocol is not specified.</p> </div>
userinfo_endpoint	http://graph.facebook.com/me	Optional	<p>Enter the userinfo token endpoint, it is required ONLY if the IDP does not support Discovery URL.</p>

Table 39-1 OpenIDConnectPlugin: Parameters for plugin configuration

Field	Sample Value	Mandatory / Optional	Description												
discovery_url	https://accounts.google.com	Mandatory	<p>Enter the discovery URL at the IDP-end with the format, <code>http(s)://URL host</code>.</p> <p>It provides authorization, token, and userinfo endpoints information.</p> <p>The base URL, <code>/.well-known/openid-configuration</code> is appended to the discover URL for building the expected format of <code>http(s)://URL host/.well-known/openid-configuration</code>.</p> <p>The plugin fetches authorization, token and userinfo endpoints from this discovery URL and redirects to the same.</p>												
username_attr			<p>This parameter indicates, which attribute should be read from the Identity Token.</p> <table border="1"> <thead> <tr> <th>provider</th> <th>Username_attr value</th> <th>Reason</th> </tr> </thead> <tbody> <tr> <td>Facebook</td> <td>name</td> <td>Sends the display name of the user via “name” attribute. E.g.: {“name” : “John Doe”}</td> </tr> <tr> <td>Google</td> <td>email</td> <td>Sends the email of the user via “email” attribute. E.g.: {“email” : “John.Doe@gmail.com”}</td> </tr> <tr> <td>IDCS</td> <td>sub</td> <td>Sends the email of the user via “sub” attribute. E.g.: {“sub” : “John.Doe@oracle.com”}</td> </tr> </tbody> </table>	provider	Username_attr value	Reason	Facebook	name	Sends the display name of the user via “ name ” attribute. E.g.: {“name” : “John Doe”}	Google	email	Sends the email of the user via “ email ” attribute. E.g.: {“email” : “John.Doe@gmail.com”}	IDCS	sub	Sends the email of the user via “ sub ” attribute. E.g.: {“sub” : “John.Doe@oracle.com”}
provider	Username_attr value	Reason													
Facebook	name	Sends the display name of the user via “ name ” attribute. E.g.: {“name” : “John Doe”}													
Google	email	Sends the email of the user via “ email ” attribute. E.g.: {“email” : “John.Doe@gmail.com”}													
IDCS	sub	Sends the email of the user via “ sub ” attribute. E.g.: {“sub” : “John.Doe@oracle.com”}													
oauth_client_id	asdad11sdfasd a131231asdas	Mandatory	Enter OAuth client id.												

There is no dependency between the provider and the expected key value. The new plugin parameter, `username_attr` indicates the attribute to be read from the Identity Token.



Note:

Add `DUseSunHttpHandler=true` before starting the server. When it is set to `true`, the plugin connects to the external IDP through the backchannel and fetches token and userinfo details.

39.2 Authentication Module and Scheme and Policy Changes

Create a new authentication module that consists of two steps:

- OpenIDConnectPlugin
- UserIdentificationPlugin

Access Manager >

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** Steps Orchestration

View ▾ + Add ✕ Delete 🔗 Detach

Step Name	Description	Plug-in Name
OIDC		OpenIDConnectPlugin
UI		UserIdentificationPlugin

Access Manager >

Authentication Module Authentication Module Duplicate Apply

Custom Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function.

General **Steps** **Steps Orchestration**

You can specify the initial step

* Initial Step

Name	Description	On Success	On Failure	On Error
OIDC		UI ▾	failure ▾	failure ▾
UI		success ▾	failure ▾	failure ▾

UserIdentificationPlugin is required as the second Step. Before the OAM session creation, the user authenticated by the IDP needs to be asserted by OAM. The User Attribute returned in the Identity Token varies for providers. The user attribute could be `subject` which is the `uid` or `displayname` or `email` attribute.

In OAM, assert this user in the IDStore by modifying the `KEY_LDAP_FILTER` as described in the following table:

Assumption: The user authenticated at IDP must exist in the OAM-ID Store.

Step Name	Description	Plug-in Name
oidc	Open ID Connect	OpenIDConnectPlugin
useridentification		UserIdentificationPlugin

Step Details

Step Name: useridentification

Description:

Plug-in Name: UserIdentificationPlugin

KEY_IDENTITY_STORE_REF:

KEY_LDAP_FILTER:

KEY_SEARCH_BASE_URL:

Table 39-2 UserIdentificationPlugin: Parameters to modify filters

Provider	User Attribute returned by OpenIDConnectPlugin	Filter
Facebook	Sub returns displayname	&(objectclass=inetorgperson) (DISPLAYNAME={KEY_USERNAME}) It indicates that the LDAP search needs to be performed for a user with the given displayname.
OAM	Sub which returns uid, the default attribute configured	Default email attribute is not supported.
IDCS	Sub returns email	&(objectclass=inetorgperson) (MAIL={KEY_USERNAME})
Google	Email returns email	&(objectclass=inetorgperson) (MAIL={KEY_USERNAME})

39.3 Authentication Scheme Changes

Create a new Authentication scheme that uses the module created in the previous section

For this new Authentication scheme, set the challenge parameter `initial_command=NONE` so that the control is passed to the plugin

Launch Pad Authentication Modules × FB-AuthModule × Authentication Schemes × **FB-Scheme ×**

Access Manager >

FB-Scheme Authentication Scheme

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme

* Name	FB-Scheme
Description	scheme to authn with Facebook
* Authentication Level	2 ▲ ▼
Default	<input type="checkbox"/>
* Challenge Method	FORM ▼
Challenge Redirect URL	/oam/server/
* Authentication Module	FB-AuthModule ▼
* Challenge URL	/pages/login.jsp
* Context Type	default ▼
* Context Value	/oam
Challenge Parameters	initial_command=NONE

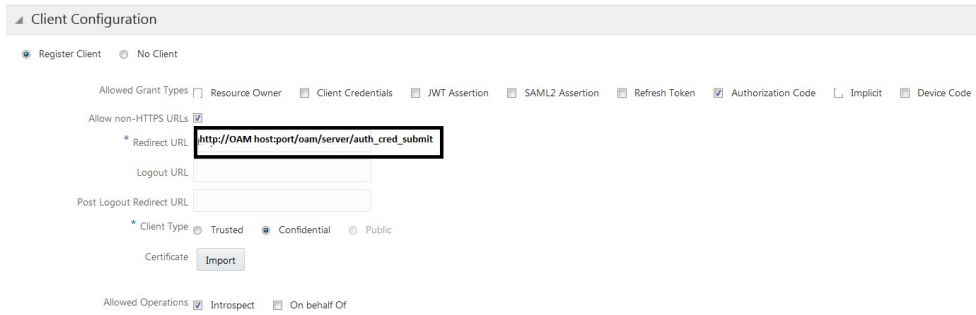
39.4 Policy Changes

Modify the authentication policy in the application domain of the protected resource to use the scheme created in [Authentication Scheme Changes](#). When you access a protected-resource, the control is given to this module. The OpenIDConnectPlugin redirects to the IDP endpoint. After authenticating at the IDP, when control is submitted back to the server, the UserIdentificationplugin asserts the user and the session are created in OAM.

39.5 Integration with IDCS

OAM ships an out-of-the box OIDC Client Authentication Plugin, OpenIDConnectPlugin that enables integration with Social Identity providers such as IDCS, Google and Facebook. Apart from being an OAuth/OpenIDConnect2.0 Server, Oracle Access Manager (OAM) can also delegate authentication to OpenIDConnect-Social Identity Providers such as IDCS, Google, Facebook or even OAM itself, thus behaving like a relying party (service provider). After authenticating the user, the IDP redirects back to OAM where the user is asserted by OAM and an OAM Session is created.

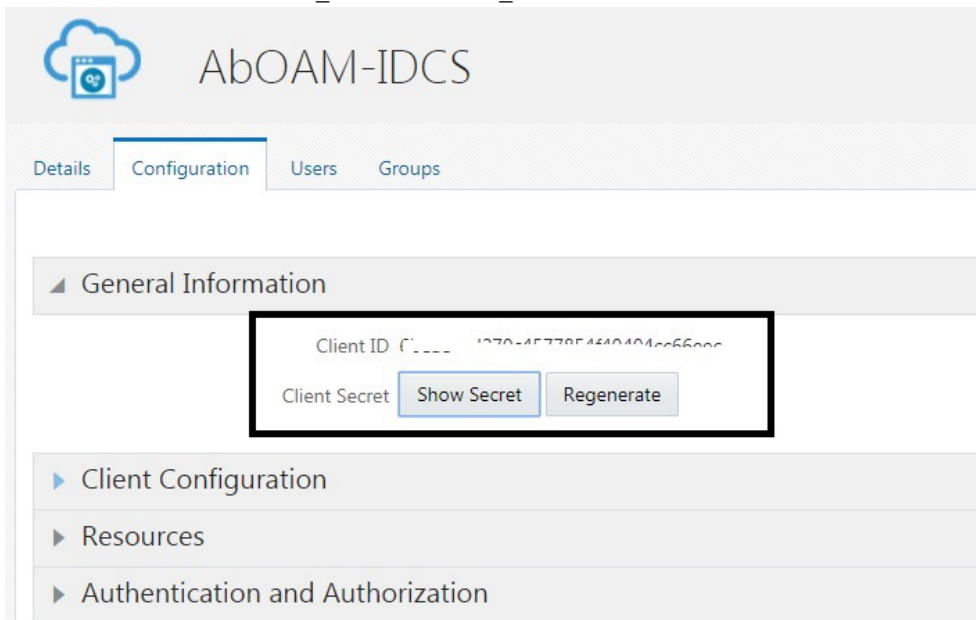
1. Create a client on the IDCS site.



 **Note:**

While creating the client, ensure to check `Authorization Code` grant type. The `redirect_uri` for the client needs to be the end point on the OAM server where credentials are submitted eg: `http(s)://OAMServer LBR Host:port/oam/server/auth_cred_submit`.

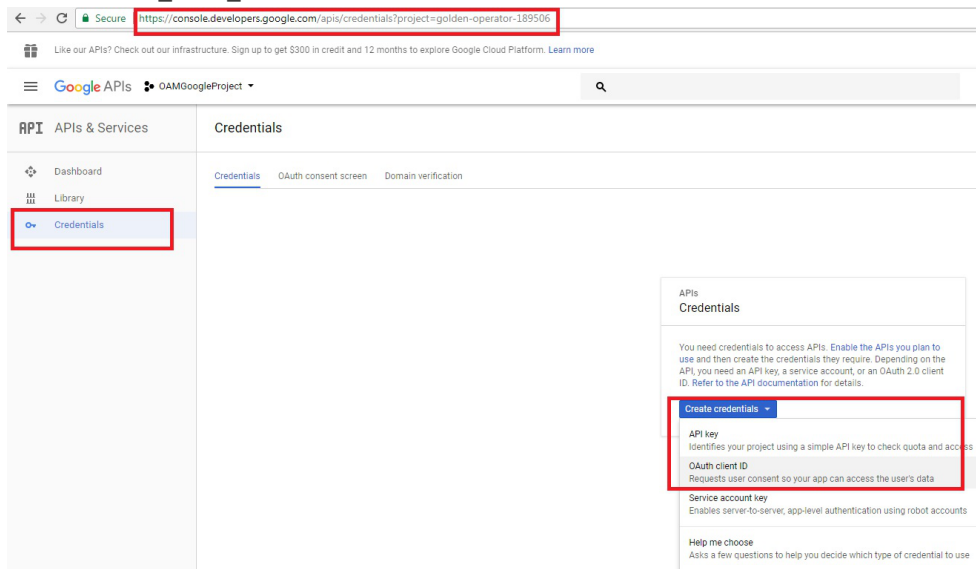
2. Create a new application under `Applications`.
3. Make a note of the `client_id` and `client_secret`.



39.6 Integration with Google

OAM ships an out-of-the box OIDC Client Authentication Plugin, OpenIDConnectPlugin that enables integration with Social Identity providers such as IDCS, Google and Facebook. Apart from being an OAuth/OpenIDConnect2.0 Server, Oracle Access Manager (OAM) can also delegate authentication to OpenIDConnect-Social Identity Providers such as IDCS, Google, Facebook or even OAM itself, thus behaving like a relying party (service provider). After authenticating the user, the IDP redirects back to OAM where the user is asserted by OAM and an OAM Session is created.

1. Create a client on `console.developers.google.com` as shown in the following figure:
2. Make a note of the `client_id` and `client_secret`.
3. Provide endpoint on the OAM server as `redirect_uri` for the client. Credentials are submitted at this endpoint, for example `http(s)://OAM Host:port/oam/server/auth_cred_submit`.



- Set the following configuration parameters for OpenIDConnectPlugin:

gp	OpenIDConnectPlugin
ui	UserIdentificationPlugin

Step Details Save Cancel

Step Name gp

Description

Plug-in Name OpenIDConnectPlugin

id_domain

oauth_client_secret

token_end_point

authz_end_point

require_proxy

provider google

scope

userinfo_end_point

additional_parameters

discovery_url https://accounts.google.com

username_attr email

oauth_client_id

- Verify that the user exists in the IDStore in OAM.

Step Name	Description	Plug-in Name
oidc	Open ID Connect	OpenIDConnectPlugin
userIdentification		UserIdentificationPlugin

Step Details

Step Name useridentification

Description

Plug-in Name UserIdentificationPlugin

KEY_IDENTITY_STORE_REF

KEY_LDAP_FILTER **&(objectclass=inetorgperson)(MAIL={KEY_USERNAME})**

KEY_SEARCH_BASE_URL

- Access the protected-resource.
- Get redirected to the Google login page for authentication.

39.7 Integration with Facebook

OAM ships an out-of-the box OIDC Client Authentication Plugin, OpenIDConnectPlugin that enables integration with Social Identity providers such as IDCS, Google and Facebook. Apart from being an OAuth/OpenIDConnect2.0 Server, Oracle Access Manager (OAM) can also delegate authentication to OpenIDConnect-Social Identity Providers such as IDCS, Google, Facebook or even OAM itself, thus behaving like a relying party (service provider). After

authenticating the user, the IDP redirects back to OAM where the user is asserted by OAM and an OAM Session is created.

1. Create a client on `developers.facebook.com` as shown in the following figure:

Service Provider Configuration Save Revert

*Required

Name:

Description:

Service Provider Java Class:

Attributes

Attributes View

Name	Value
OAM_VERSION	OAM_11G
DEBUG_VALUE	0
TRANSPORT_SECURITY	OPEN
OAM_SERVER_1	slc01mqd.us.oracle.com:5675
OAM_SERVER_1_MAX_CONN	4
OAM_SERVER_2	slc01mqd.us.oracle.com:5675
OAM_SERVER_2_MAX_CONN	4
colocated.oam	true

WebGate Agent

* WebGate ID:

Encrypted Password:

2. Make a note of the `client_id` and `client_secret`.

The screenshot shows the OAMPluginApp dashboard. The 'APP ID' field is highlighted with a red box, and the 'App Secret' field is also highlighted with a red box. A blue callout box says 'Click to add and configure developer products.' The dashboard also shows the API Version as 'v2.11' and a 'View Analytics' link.

3. Provide endpoint on the OAM server as `redirect_uri` for the client. Credentials are submitted at this endpoint, for example `http(s)://OAM Host:port/oam/`

server/auth_cred_submit.

The screenshot shows the Oracle OAM Admin console interface. On the left is a navigation menu with options like Dashboard, Settings, Roles, Alerts, and App Review. The main content area is titled 'Client OAuth Settings'. It includes several toggle switches for 'Client OAuth Login', 'Web OAuth Login', 'Embedded Browser OAuth Login', 'Force Web OAuth Reauthentication', 'Use Strict Mode for Redirect URIs', and 'Login from Devices'. A red box highlights the 'Valid OAuth redirect URIs' field, which contains the text 'http://OAM Server Host:Port/oam/server/auth_cred_submit'. There is also a warning message at the top regarding a security update in March.

4. Set the following configuration parameters for OpenIDConnectPlugin:

Step Name	Description	Plug-in Name
OIDC		OpenIDConnectPlugin
UI		UserIdentificationPlugin

Step Details

Step Name:

Description:

Plug-in Name:

id_domain:

oauth_client_secret:

token_end_point:

authz_end_point:

require_proxy:

provider:

scope:

additional_parameters:

userinfo_end_point:

discovery_url:

username_attr:

oauth_client_id:

Save Cancel

5. Verify that the user exists in the IDStore in OAM.

Step Name	Description	Plug-in Name
oidc	Open ID Connect	OpenIDConnectPlugin
useridentification		UserIdentificationPlugIn

Step Details

Step Name useridentification

Description

Plug-in Name UserIdentificationPlugIn

KEY_IDENTITY_STORE_REF

KEY_LDAP_FILTER **&(objectclass=inetorgperson)(DISPLAYNAME={KEY_USERNAME})**

KEY_SEARCH_BASE_URL

6. Access the protected-resource.
7. Get redirected to the Facebook login page for authentication.

OAuth Just-In-Time (JIT) User Provisioning

This section provides an overview of OAuth Just-in-Time user provisioning and configurations in OAM.

Just-In-Time user provisioning enables user identity to be provisioned dynamically when the user tries to login for the first time using any social identity providers. User account creation is done directly without the need to provision users in the system, in advance.

OAM provides the following ways to configure and use Just-In-Time user provisioning:

- [Using Self-Registration for Just-in-Time User Provisioning](#)
If you need more control over user provisioning, based on your proprietary flow, you can choose to redirect the user to your own self-registration page.
- [Configuring Just-In-Time User Auto-Provisioning with Password Prompt](#)
If you need the user provisioning to be done automatically during user authentication through OpenIDConnect protocol, then you can choose to configure for user auto-provisioning. The users are created in the configured identity store and automatically provisioned with the user attribute values such as userID, user name, first name, last name, e-mail, and so on. These values are retrieved from the IDToken received from the Identity Providers (Google, Facebook).

You can also choose to have a password prompt for more control over the newly created user account.
- [Configuring Just-In-Time User Auto-Provisioning \(No Password Prompt\)](#)
You can choose to have no password prompt during JIT Auto-Provisioning. The user created by this flow cannot be used for any other type of authentication.

40.1 Using Self-Registration for Just-in-Time User Provisioning

OAM enables you to have more control over user provisioning, based on your proprietary user-provisioning flow. Follow the steps in this section to use self-registration for Just-in-Time user provisioning.

- [Prerequisites for Just-in-Time Provisioning by Self-Registration](#)
- [About Self-Registration Page](#)
- [Configuring UserSelfRegistration Authentication Module and Scheme](#)
- [Protecting Resources with UserSelfRegistrationScheme](#)

40.1.1 Prerequisites for Just-in-Time Provisioning by Self-Registration

Ensure the following steps are followed before proceeding to the configurations for Just-in-Time user provisioning by self-registration.

1. Add the `OAuthUserSelfRegistration` plugin to the OAM console.

The Just-In-Time user provisioning is supported by an out-of-the-box `OAuthUserSelfRegistration` plugin.

If the plugin is not available on the console, you must run the `configurePluginMetadata` WLST command as described in the following steps:

- a. Create an XML properties file in your local directory. For example, `propFileName.xml`. Add the following plugin metadata content to the XML file you created. You can later change the values from the `oam-console`, if necessary.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/
properties.dtd">
<properties>
  <entry
key="implementation">oracle.security.am.plugin.authn.OAuthUserSel
fRegistrationPlugin</entry>
  <entry key="email">donotreply@oracle.com</entry>
  <entry key="source">System</entry>
  <entry key="status">ACTIVATED</entry>
  <entry key="description">User self registration plugin for
JIT</entry>
  <entry key="jarFileName"></entry>
  <entry key="version">10</entry>
  <entry key="author">uid=orcladmin</entry>
  <entry key="name">OAuthUserSelfRegistrationPlugin</entry>
  <entry key="checksum"></entry>
  <entry
key="interface">oracle.security.am.plugin.authn.AbstractAuthentic
ationPlugIn</entry>
  <entry key="type">Authentication</entry>
  <entry
key="initParameters.autoprovision_password_policy_regex">string~3
00~false</entry>
  <entry
key="initParameters.registration_url">string~300~true</entry>
  <entry key="initParameters.secret_key">string~300~true</
entry>
  <entry
key="initParameters.autoprovisioning">string~300~false</entry>
  <entry
key="initParameters.autoprovision_key_user_record_attribute_list"
>string~300~true</entry>
  <entry
key="initParameters.autoprovision_idtoken_to_user_attribute_mappi
ng">string~300~true</entry>
  <entry
key="initParameters.autoprovision_useridentitystore">string~300~f
alse</entry>
</properties>
```

- b. Run the following WLST command:

```
configurePluginMetadata('OAuthUserSelfRegistrationPlugin','/
path_to_file/propFileName.xml')
```

where, `OAuthUserSelfRegistrationPlugin` is the name of the plugin and `/path_to_file/propFileName.xml` is the directory, in which, the plugin metadata XML file was created in the previous step.

- c. Restart the Administration servers.
2. Enable the Identity Federation Service

Identity Federation service must be enabled for the Just in Time user provisioning to work.

- a. Log in to the Oracle Access Management Console

```
https://OAMAdminHost:OAMAdminPort/oamconsole/
```

- b. From the Welcome page, click **Configuration** and then click **Available Services**
 - c. Under Federation, click **Enable Service** beside **Identity Federation** (or confirm that the green status check mark displays).
A Confirmation window is displayed.
3. Set the `blobdiscovery` provider setting to `RDBMSBlobDiscoveryProvider`, using the following WLST command.

```
setDiscoveryProvider('blobdiscovery', 'oracle.security.fed.jvt.discovery.model.profilestate.RDBMSBlobDiscoveryProvider')
```

Verify the settings using the following WLST command:

```
displayDiscoveryProvider('blobdiscovery')
```

For more information about these WLST commands, see `setDiscoveryProvider` and `displayDiscoveryProvider` in *WebLogic Scripting Tool Command Reference for Identity and Access Management*.

4. Create a self-registration page and deploy the registration Url on the WebLogic Server. See [About Self-Registration Page](#) details.
5. Create an OAuth Client on your preferred Identity Providers.

For example:

- For creating OAuth client on Google, see [Integration with Google](#)
- For creating OAuth client on Facebook, see [Integration with Facebook](#)

40.1.2 About Self-Registration Page

This section provides guidelines for creating the self-registration page.

You can have more control over user provisioning, based on your proprietary user-provisioning flow, by redirecting the users to a self-registration page, owned by you.

The self-registration page must be able to perform the following:

 **Note:**

Ensure you have completed all the prerequisite steps before creating your self-registration page. See [Prerequisites for Just-in-Time Provisioning by Self-Registration](#) for details.

1. Call the user info endpoint to get the user attributes required to create the user. The user `registrationid` is received as a query parameter from OAM. For example:

```
curl -X GET 'http://<ManagedOAMHost>:<ManagedOAMPort>/oamfed/user/
regdata?registrationid=<registrationId>
```

The response must look similar to the following example:

```
{
  idtoken : "{\"iss\":\"https://
accounts.google.com\", \"azp\":\"288915398401-
etd60uissja597aed1veoc0hlhist1jg.apps.googleusercontent.com\", \"aud\
\": \"288915398401-
etd60uissja597aed1veoc0hlhist1jg.apps.googleusercontent.com\", \"sub\
\": \"104056721435364290436\", \"email\": \"johndoe@gmail.com\", \"email_
verified\": true, \"at_hash\": \"KVRh2eKbxnhN3fy0jESgNQ\", \"iat\": 15700
75783, \"exp\": 1570079383},
  hmac : "<value>"
}
```

2. HMAC Validation. You can calculate the HMAC of the user data received (contents of parameter `idtoken`), using the `secret_key` configured in the `OAuthUserSelfRegistrationPlugin` and do the `base64url` encoding of the calculated HMAC. This value must match with the `base64url` encoding of the HMAC value received in Step 1, confirming the message integrity. You can fail the requests if the HMAC values do not match.
3. Create a user account in the configured OAM user store. In this step, use your proprietary logic for user account creation in your configured user store.
4. Redirect back to OAM on the following URL with the user `registrationid`.

```
http(s)://<ManagedOAMHost>:<ManagedOAMPort>/oam/server/
auth_cred_submit
```

For information on how to deploy the self-registration page on OAM server, see [JSP/HTML Deployment](#)

 **Note:**

The Self-Registration page can reside on an external server.

Example 40-1 Sample Self-Registration Page

Your self-registration page must be similar to the following sample registration provided:

```
<!DOCTYPE html>
<%@ page contentType="text/html; charset=UTF-8"%>
<%@ page import="java.util.*, java.net.*, java.io.*"%>
<%@ page import="org.codehaus.jettison.json.JSONObject"%>
<%@ page import="javax.crypto.*, javax.crypto.spec.*"%>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Register</title>
  </head>
  <body>
    <h1>Creating User with following Attributes</h1>
    <%! JSONObject jsonBody=null; %>
    <%! String userdata="text"; %>
    <%! String managedHostName =
System.getProperty("managed.host.name"); %>
    <%! String managedPort = System.getProperty("managed.port"); %>
    <%! String managedprotocol = System.getProperty("managed.protocol");
%>
    <%
String statusCode ="RegistrationFailed";
try{

//Step 1 : Get the userinfo from OAM, using the following endpoint

//Verify and sanitise query parm to avoid XSS attack
String regRefId =
XSSFilter.sanitizeInput("registrationid",request.getParameter("registrationid
"));
String regDataURLPath = "/oamfed/user/regdata?registrationid=" +
regRefId;
URL regDataURL = new URL(managedprotocol + "://" + managedHostName
+ ":" + managedPort + regDataURLPath);
URLConnection connection =
(HttpURLConnection)regDataURL.openConnection();
String data=null;
String hmac=null;
if (connection.getResponseCode() == 200) {
statusCode ="RegistrationSuccess";
ByteArrayOutputStream bos = new ByteArrayOutputStream();
BufferedInputStream bis = new
BufferedInputStream(connection.getInputStream());
byte[] b = new byte[256];
int i;
while ((i = bis.read(b)) != -1)
bos.write(b, 0, i);
byte[] body = bos.toByteArray();
userdata=new String(body);
jsonBody = new JSONObject(new String(body));
//Save the userinfo as returned from OAM in parameter idtoken
```

```
data = jsonBody.getString("idtoken");
//Save the hmac of the data , to be used later.
hhmac = jsonBody.getString("hmac");
out.println(jsonBody.toString());
}else{
    out.println("got error");
    out.println(connection.getResponseMessage());
}

//End of Step 1

//Step 2 : (Optional) if you choose to compare the hmac of
received data for security purposes
//Calculate the hmac of the data received in idtoken using your
secret_key, as configured in OAuthUserSelfRegistrationPlugin.

String key= "any_data"; //secret_key value
String algo ="HmacSHA256";
SecretKeySpec signingKey = new SecretKeySpec(key.getBytes(),
algo);
    Mac mac;
    byte[] calculatedHmac = null;
    try {
        mac = Mac.getInstance(algo);
        mac.init(signingKey);
        calculatedHmac = mac.doFinal(data.getBytes());
    } catch (Exception e) {
        //handle error
    }
    String base64ofcalculatedHmac =
java.util.Base64.getUrlEncoder().encodeToString(calculatedHmac);
    if(hhmac.equals(base64ofcalculatedHmac)){
        out.println("/n hmac matches");
    }else{
        out.println("/n hmac verification Failed");
    }
    } catch (Throwable t)
{
    out.println(t.getMessage());
}

//End of Step 2

//Step 3: Add your proprietary logic for user account creation in
your configured UserStore, using the userinfo as retrieved in Step 1
/**
 * Add User Account creation Logic here
 *
 **/
//End of Step 3

%>

//Step 4: In this example the redirect is triggered by button
click action
```

```

        <button id="myButton" class="float-left submit-button">Create User</
button>
        <script type="text/javascript">
            var url_string = window.location.href;
            var managedprotocoll = "<%= managedprotocol %>";
            var managedHostName1 = "<%= managedHostName %>";
            var managedPort1 = "<%= managedPort %>";
            var redirect = managedprotocoll + "://" + managedHostName1 + ":" +
managedPort1 + "/oam/server/auth_cred_submit";
            document.getElementById("myButton").onclick = function () {
                location.href = redirect;

                //End of Step 4
            };
        </script>
    </body>
</html>

```

40.1.3 Configuring UserSelfRegistration Authentication Module and Scheme

- [Just In Time UserSelfRegistration Authentication Module](#)
- [JIT UserSelfRegistration Steps Orchestration](#)
- [Just In Time UserSelfRegistration Authentication Scheme](#)

40.1.3.1 Just In Time UserSelfRegistration Authentication Module

Create a new Authentication module and configure the following four steps: OpenIDConnectPlugin, UserIdentificationPlugin, OAuthUserSelfRegistrationPlugin, and UserIdentificationPlugin.

To configure the UserSelfRegistration authentication module in the Oracle Access Management Console:

1. Log into the Oracle Access Management Console as System Administrator.
2. From the Application Security Launch Pad, click **Authentication Modules** under **Plugins**.
3. From the Authentication Modules tab, click **Create Authentication Module** and then **Custom Authentication Module**.
4. Add a name and description for the authentication module under the General tab. For example, UserSelfRegistration.
5. Add the UserSelfRegistration authentication module steps as follows:
 - a. Add OpenIDConnectPlugin and UserIdentificationPlugin steps based on the Identity provider. For details, see [Authentication Module and Scheme and Policy Changes](#)
 - b. Add the OAuthUserSelfRegistrationPlugin step and set the following parameters:

Table 40-1 OAuthUserSelfRegistrationPlugin Step

Step Details	Description
registration_url	Mandatory. The absolute url of the customer owned self-registration page.
secret_key	Mandatory. This secret key is used to create hmac of the request payload to be sent to userInformation RestAPI (the custom self-registration page must use the same key value for HMAC verify). If encrypted key is to be used, add the prefix {AES} to encrypted key value.

6. Add the UserIdentificationPlugin step and set the parameters as described in [Authentication Module and Scheme and Policy Changes](#) .

40.1.3.2 JIT UserSelfRegistration Steps Orchestration

Configure the Orchestration steps for the Authorization flow.

1. Click the **Steps Orchestration** subtab
2. From the **InitialStep** list, choose the OpenIDConnectPlugin step.
3. Set **On Success**, **On Failure**, and **On Error** for each of the steps as shown in the following example:

Table 40-2 JIT Step Orchestration

Name	Description	On Success	On Failure	On Error
OpenID_Connect	OpenIDConnectPlugin Step	User_Identification1	Failure	Failure
User_Identification1	First UserIdentificationPlugin Step	Success	OAuth_Self_Registration	Failure
OAuth_Self_Registration	OAuthUserSelfRegistrationPlugin Step	User_Identification2	Failure	Failure
User_Identification2	Second UserIdentificationPlugin Step	Success	Failure	Failure

40.1.3.3 Just In Time UserSelfRegistration Authentication Scheme

Set the challenge parameter `initial_command=NONE` in the authentication scheme.

To create a new Authentication Scheme:

1. From the **Application Security** Launch Pad, click **Authentication Schemes** under **Access Manager**.
2. From the **Authentication Schemes** tab, click **Create Authentication Scheme**.
3. Set the parameters in the authentication scheme. For example, the following figure shows a Sample Authentication Scheme Page:

Figure 40-1 Sample Authentication Scheme Page

Access Manager >

Create Authentication Scheme

 Authentication Scheme Set As Default Apply

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme must also include a defined Authentication Module.

* Name: JIT_SelfRegistration

Description: JIT SelfRegistration

* Authentication Level: 2

Default:

* Challenge Method: FORM

Challenge Redirect URL: /oam/server

* Authentication Module: UserSelfRegistration

* Challenge URL: /pages/login.jsp

* Context Type: default

* Context Value: /oam

Challenge Parameters: initial_command=NONE

For details about the parameters, see [Authentication Schemes and Pages](#)

4. Under Challenge Parameters, add `initial_command=NONE`.

40.1.4 Protecting Resources with UserSelfRegistrationScheme

Complete the configuration for the Just in Time user provisioning by self-registration, by assigning the UserSelfRegistrationScheme authentication scheme to the protected resource policy.

1. From the **Application Security** Launch Pad, select **Application Domains** under **Access Manager**.
2. Search and open the required **Application Domain**.
3. Open the **Authentication Policies** tab and click **Protected Resource Policy**.
4. Select the **UserSelfRegistrationScheme** from the **Authentication Scheme** dropdown list and click **Apply**.

40.2 Configuring Just-In-Time User Auto-Provisioning with Password Prompt

Follow the steps in this section to configure Just-in-Time user auto-provisioning with password prompt.

- [Prerequisites for Just-in-Time User Auto-Provisioning](#)
- [Configuring AutoProvisioning Authentication Module and Scheme](#)
- [Protecting Resources with AutoProvisioningScheme](#)

40.2.1 Prerequisites for Just-in-Time User Auto-Provisioning

Ensure the following steps are followed before proceeding to the configurations for auto-provisioning.

1. Add the `OAuthUserSelfRegistration` plugin to the OAM console.

The Just-In-Time user provisioning is supported by an out-of-the-box `OAuthUserSelfRegistration` plugin.

If the plugin is not available on the console, you must run the `configurePluginMetadata` WLST command as described in the following steps:

- a. Create an XML properties file named `propFileName.xml` in your local directory. Add the following content to the XML file you created. You can later change the values from the `oam-console`, if necessary

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/
properties.dtd">
<properties>
  <entry
key="implementation">oracle.security.am.plugin.authn.OAuthUserSel
fRegistrationPlugin</entry>
  <entry key="email">donotreply@oracle.com</entry>
  <entry key="source">System</entry>
  <entry key="status">ACTIVATED</entry>
  <entry key="description">User self registration plugin for
JIT</entry>
  <entry key="jarFileName"></entry>
  <entry key="version">10</entry>
  <entry key="author">uid=orcladmin</entry>
  <entry key="name">OAuthUserSelfRegistrationPlugin</entry>
  <entry key="checksum"></entry>
  <entry
key="interface">oracle.security.am.plugin.authn.AbstractAuthentic
ationPlugIn</entry>
  <entry key="type">Authentication</entry>
  <entry
key="initParameters.autoprovision_password_policy_regex">string~3
00~false</entry>
  <entry
key="initParameters.registration_url">string~300~true</entry>
  <entry key="initParameters.secret_key">string~300~true</
entry>
  <entry
key="initParameters.autoprovisioning">string~300~false</entry>
  <entry
key="initParameters.autoprovision_key_user_record_attribute_list"
>string~300~true</entry>
  <entry
key="initParameters.autoprovision_idtoken_to_user_attribute_mappi
ng">string~300~true</entry>
  <entry
key="initParameters.autoprovision_useridentitystore">string~300~f
```

```

else</entry>
</properties>

```

- b. Run the following WLST command

```

configurePluginMetadata('OAuthUserSelfRegistrationPlugin','/
path_to_file/propFileName.xml')

```

where, `OAuthUserSelfRegistrationPlugin` is the name of the plugin and `/path_to_file/propFileName.xml` is the directory, in which, the plugin metadata XML file was created in the previous step.

- c. Restart the Administration servers.
2. Create an OAuth Client on your preferred Identity Providers.

For Example:

- For creating OAuth client on Google, see [Integration with Google](#)
- For creating OAuth client on Facebook, see [Integration with Facebook](#)

40.2.2 Configuring AutoProvisioning Authentication Module and Scheme

- [Just-In-Time User Auto-Provisioning Authentication Module](#)
- [JIT User Auto-provisioning Steps Orchestration](#)
- [Just-In-Time User AutoProvisioningScheme](#)

40.2.2.1 Just-In-Time User Auto-Provisioning Authentication Module

Create a new Authentication module and configure the following five steps: OpenIDConnectPlugin, UserIdentificationPlugin, OAuthUserSelfRegistrationPlugin, CredentialCollectorPlugin, and UserIdentificationPlugin.

To configure the Just in Time user Autoprovisioning authentication module in the Oracle Access Management Console:

1. Log into the Oracle Access Management Console as System Administrator.
2. From the Application Security Launch Pad, click **Authentication Modules** under **Plugins**.
3. From the Authentication Modules tab, click **Create Authentication Module** and then **Custom Authentication Module**.
4. Add a name and description for the authentication module under the General tab. For example, `AutoProvisioning_WithPasswd`.
5. Add the AutoProvisioning authentication module steps as follows:
 - a. Add the OpenIDConnectPlugin and UserIdentificationPlugin steps for the Identity Providers, such as Google and Facebook. For details, see [About the OpenIDConnectPlugin](#) and [Authentication Module and Scheme and Policy Changes](#).

 **Note:**

The `username_attr` configured in the OpenIDConnect Plugin indicates the attribute to be read from the identity token to get information about the user. This parameter is used as the `UserName` when creating the new user account

- b. Add the `OAuthUserSelfRegistrationPlugin` step and set the following parameters:

Table 40-3 OAuthUserSelfRegistrationPlugin Step for auto-provisioning

Description	
<code>autoprovisioning</code>	Set this to <code>True</code> (case insensitive) to enable auto-provisioning. If the value is not specified, or set to <code>False</code> auto-provisioning is not enabled, and user provisioning by self-registration is enabled.
<code>autoprovision_key_user_record_attribute_list</code>	Mandatory Specify a comma-separated list of attributes from the identity token that must be used to create the user record. Only the attributes listed here are set in the provisioned user record.
<code>autoprovision_idtoken_to_user_attribute_mapping</code>	Optional The attribute names fetched from the idtoken must be consistent with the attribute names specified in the identity store. If the attributes names are not consistent, specify a comma-separated list of <code>key:value</code> mapping, where <code>key</code> is the attribute name used in Idtoken and <code>value</code> is the attribute name used in the identity store. For example, <code>email:mail, firstName:first-name</code>
<code>autoprovision_useridentitystore</code>	Optional Specify the Identity store name. If not provided, the default value is used.
<code>autoprovision_password_policy_regex</code>	Optional Specify a valid regular expression that the password string must follow.

- c. Add the `CredentialCollectorPlugin` step and set the following parameters:

Table 40-4 CredentialCollectorPlugin Step for Auto-Provisioning

<code>actiontype</code>	REDIRECT
-------------------------	----------

Table 40-4 (Cont.) CredentialCollectorPlugin Step for Auto-Provisioning

CRED_PARAM_3	{ID=CONFIRM_PASSWORD}, {DISPLAY_NAME=CONFIRM_PASSWORD}, {TYPE=password}
CRED_PARAM_2	{ID=PASSWORD}, {DISPLAY NAME=PASSWORD}, {TYPE=password}
CRED_PARAM_1	{ID=USERNAME}, {DISPLAY NAME=USERNAME }, {TYPE=text_readonly},{VALUE=constant}
loginPageURL	/CustomReadServlet
NO_OF_CREDENTIALS	3

For details about all the remaining parameters in this step, see [Table 22-12](#)

- d. Add the UserIdentificationPlugin step and set the parameters as described in [Authentication Module and Scheme and Policy Changes](#) .

40.2.2.2 JIT User Auto-provisioning Steps Orchestration

Configure the Orchestration steps for the Authorization flow.

1. Click the **Steps Orchestration** subtab
2. From the **InitialStep list**, choose the OpenIDConnectPlugin step.
3. Set **On Success**, **On Failure**, and **On Error** for each of the steps as shown in the following example:

Table 40-5 JIT Step Orchestration

Name	Description	On Success	On Failure	On Error
OpenID_Connect	OpenIDConnectPlugin Step	User_Identification1	Failure	Failure
User_Identification1	First UserIdentificationPlugin Step	Success	Credential_Collector	Failure
Credential_Collector	CredentialCollectorPlugin Step	OAuth_Self_Registration	Failure	Failure
OAuth_Self_Registration	OAuthUserSelfRegistrationPlugin Step	User_Identification2	Failure	Failure
User_Identification2	Second UserIdentificationPlugin Step	Success	Failure	Failure

40.2.2.3 Just-In-Time User AutoProvisioningScheme

Create a custom HTML page and set the context type and context value in the authentication scheme.

To create a new Authentication Scheme:

1. From the **Application Security** Launch Pad, click **Authentication Schemes** under **Access Manager**.

2. From the **Authentication Schemes** tab, click **Create Authentication Scheme**.
3. Set the **Challenge URL** to `/CustomReadServlet`
4. Change the **Context Type** option from default to **customHTML**.
5. In **Context Value**, specify the absolute path to the customHTML page.
You must create the custom HTML file with code similar to the following sample:

```
<h4> Creating User Account with following details </h4>
<form id="loginData" action="/oam/server/auth_cred_submit"
method="post" name="loginData">
  <div id="oam_credentials" class="input-row">
    <span class="ctrl"></span>
  </div>
  <div class="button-row">
    <span class="ctrl">
      <input id="login_button" type="submit" value="Login"
class="formButton"
onclick="this.disabled=true;document.body.style.cursor = 'wait';
        this.className='formButton-
disabled';form.submit();return false;"/>
    </span>
  </div>
  <div id="ignore_oam_error_codes"></div>
</form>
```

For globalization, create a properties file in the same directory as the custom html file. The name of the file must be the same as the custom html file with `.properties` extension.

```
USERNAME=Username
PASSWORD=Password
CONFIRM_PASSWORD=Confirm password
```

6. Under **Challenge Parameters**, add `initial_command=NONE`

 **Note:**

The default number of retries for failed Authentication is 5. You can modify this using `OverrideRetryLimit` in the challenge parameters. For example, `OverrideRetryLimit=1`.

7. Set all the remaining parameters for this scheme. For example, the following figure shows a Sample Authentication Scheme Page:

Figure 40-2 Sample Authentication Scheme Page

Access Manager >

JIT_AutoProvisioning_WithPasswd Authentication Scheme Set As Default Duplicate Apply

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme must also include a defined Authentication Module.

* Name: JIT_AutoProvisioning_WithPasswd

Description: JIT AutoProvisioning With Password

* Authentication Level: 2

Default:

* Challenge Method: FORM

Challenge Redirect URL: /oam/server

* Authentication Module: AutoProvisioning_WithPasswd

* Challenge URL: /CustomReadServlet

* Context Type: customHtml

* Context Value: <domain_home>/config/fmwconfig/<filename>.html

Challenge Parameters: initial_command=NONE

For details about the parameters, see [Authentication Schemes and Pages](#)

40.2.3 Protecting Resources with AutoProvisioningScheme

Complete the configuration for the Just-in-Time Auto-provisioning, by assigning the AutoProvisioningScheme to the protected resource policy.

1. From the **Application Security** Launch Pad, select **Application Domains** under **Access Manager**.
2. Search and open the required **Application Domain**.
3. Open the **Authentication Policies** tab and click **Protected Resource Policy**.
4. Select the **AutoProvisioningScheme** from the **Authentication Scheme** dropdown list and click **Apply**.

40.3 Configuring Just-In-Time User Auto-Provisioning (No Password Prompt)

Follow the steps in this section to configure Just-in-Time user auto-provisioning with no password prompt.

- [Prerequisites for JIT User Auto-Provisioning \(No Password Prompt\)](#)
- [Configuring JIT AutoProvisioning Authentication Module and Scheme \(No Password Prompt\)](#)
- [Protecting Resources with AutoProvisioningScheme \(No Password Prompt\)](#)

40.3.1 Prerequisites for JIT User Auto-Provisioning (No Password Prompt)

Ensure the following steps are followed before proceeding with the configurations for auto-provisioning (no password prompt)

1. Add the `OAuthUserSelfRegistration` plugin to the OAM console.

The Just-In-Time user provisioning is supported by an out-of-the-box `OAuthUserSelfRegistration` plugin.

If the plugin is not available on the console, you must run the `configurePluginMetadata` WLST command as described in the following steps:

- a. Create an XML properties file named `propFileName.xml` in your local directory. Add the following content to the XML file you created. You can later change the values from the oam-console, if necessary

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/
properties.dtd">
<properties>
  <entry
key="implementation">oracle.security.am.plugin.authn.OAuthUserSel
fRegistrationPlugin</entry>
  <entry key="email">donotreply@oracle.com</entry>
  <entry key="source">System</entry>
  <entry key="status">ACTIVATED</entry>
  <entry key="description">User self registration plugin for
JIT</entry>
  <entry key="jarFileName"></entry>
  <entry key="version">10</entry>
  <entry key="author">uid=orcladmin</entry>
  <entry key="name">OAuthUserSelfRegistrationPlugin</entry>
  <entry key="checksum"></entry>
  <entry
key="interface">oracle.security.am.plugin.authn.AbstractAuthentic
ationPlugIn</entry>
  <entry key="type">Authentication</entry>
  <entry
key="initParameters.autoprovision_password_policy_regex">string~3
00~false</entry>
  <entry
key="initParameters.registration_url">string~300~true</entry>
  <entry key="initParameters.secret_key">string~300~true</
entry>
  <entry
key="initParameters.autoprovisioning">string~300~false</entry>
  <entry
key="initParameters.autoprovision_key_user_record_attribute_list"
>string~300~true</entry>
  <entry
key="initParameters.autoprovision_idtoken_to_user_attribute_mappi
ng">string~300~true</entry>
  <entry
key="initParameters.autoprovision_useridentitystore">string~300~f
alse</entry>
</properties>
```

- b. Run the following WLST command

```
configurePluginMetadata('OAuthUserSelfRegistrationPlugin','/
path_to_file/propFileName.xml')
```

where, `OAuthUserSelfRegistrationPlugin` is the name of the plugin and `/path_to_file/propFileName.xml` is the directory, in which, the plugin metadata XML file was created in the previous step.

- c. Restart the Administration servers.
2. Create an OAuth Client on your preferred Identity Providers.

For Example:

- For creating OAuth client on Google, see [Integration with Google](#)
- For creating OAuth client on Facebook, see [Integration with Facebook](#)

40.3.2 Configuring JIT AutoProvisioning Authentication Module and Scheme (No Password Prompt)

- [Just-In-Time User Auto-Provisioning Authentication Module \(No Password Prompt\)](#)
- [JIT User Auto-provisioning Steps Orchestration \(No Password Prompt\)](#)
- [Just-In-Time User AutoProvisioningScheme \(No Password Prompt\)](#)

40.3.2.1 Just-In-Time User Auto-Provisioning Authentication Module (No Password Prompt)

Create a new Authentication module and configure the following four steps: OpenIDConnectPlugin, UserIdentificationPlugin, OAuthUserSelfRegistrationPlugin, and UserIdentificationPlugin.

To configure the Just in Time user Autoprovisioning authentication module in the Oracle Access Management Console:

1. Log into the Oracle Access Management Console as System Administrator.
2. From the Application Security Launch Pad, click **Authentication Modules** under **Plugins**.
3. From the Authentication Modules tab, click **Create Authentication Module** and then **Custom Authentication Module**.
4. Add a name and description for the authentication module under the General tab. For example, `AutoProvisioning_NoPasswd`.
5. Add the AutoProvisioning authentication module steps as follows:
 - a. Add the OpenIDConnectPlugin and UserIdentificationPlugin steps for the Identity Providers, such as Google and Facebook. For details, see [About the OpenIDConnectPlugin](#) and [Authentication Module and Scheme and Policy Changes](#).

 **Note:**

The `username_attr` configured in the OpenIDConnect Plugin indicates the attribute to be read from the identity token to get information about the user. This parameter is used as the `UserName` when creating the new user account

- b. Add the `OAuthUserSelfRegistrationPlugin` step and set the following parameters:

Table 40-6 OAuthUserSelfRegistrationPlugin Step for auto-provisioning

Description	
<code>autoprovisioning</code>	Set this to <code>True</code> (case insensitive) to enable auto-provisioning. If the value is not specified, or set to <code>False</code> auto-provisioning is not enabled, and user provisioning by self-registration is enabled.
<code>autoprovision_key_user_record_attribute_list</code>	Mandatory Specify a comma-separated list of attributes from the identity token that must be used to create the user record. Only the attributes listed here are set in the provisioned user record.
<code>autoprovision_idtoken_to_user_attribute_mapping</code>	Optional The attribute names fetched from the idtoken must be consistent with the attribute names specified in the identity store. If the attributes names are not consistent, specify a comma-separated list of <code>key:value</code> mapping, where <code>key</code> is the attribute name used in Idtoken and <code>value</code> is the attribute name used in the identity store. For example, <code>email:mail, firstName:first-name</code>
<code>autoprovision_useridentitystore</code>	Optional Specify the Identity store name. If not provided, the default value is used.
<code>autoprovision_password_policy_regex</code>	Not applicable for the JIT User Auto-Provisioning (No Password Prompt) flow. This parameter is ignored.

- c. Add the `UserIdentificationPlugin` step and set the parameters as described in [Authentication Module and Scheme and Policy Changes](#) .

40.3.2.2 JIT User Auto-provisioning Steps Orchestration (No Password Prompt)

Configure the Orchestration steps for the Authorization flow.

1. Click the **Steps Orchestration** subtab
2. From the **InitialStep list**, choose the OpenIDConnectPlugin step.
3. Set **On Success**, **On Failure**, and **On Error** for each of the steps as shown in the following example:

Table 40-7 JIT Step Orchestration

Name	Description	On Success	On Failure	On Error
OpenID_Connect	OpenIDConnectPlugin Step	User_Identification1	Failure	Failure
User_Identification1	First UserIdentificationPlugin Step	Success	OAuth_Self_Registration	Failure
OAuth_Self_Registration	OAuthUserSelfRegistrationPlugin Step	User_Identification2	Failure	Failure
User_Identification2	Second UserIdentificationPlugin Step	Success	Failure	Failure

40.3.2.3 Just-In-Time User AutoProvisioningScheme (No Password Prompt)

Set the challenge parameter `initial_command=NONE` in the authentication scheme.

To create a new Authentication Scheme:

1. From the **Application Security** Launch Pad, click **Authentication Schemes** under **Access Manager**.
2. From the **Authentication Schemes tab**, click **Create Authentication Scheme**.
3. Set the parameters in the authentication scheme. For example, the following figure shows a Sample Authentication Scheme Page:

Figure 40-3 Sample Authentication Scheme Page

Access Manager >

JIT_AutoProvisioning_NoPasswd Authentication Scheme Set As Default Duplicate Apply

An Authentication Scheme defines the challenge mechanism required to authenticate a user. Each Authentication Scheme must also include a defined Authentication Module.

* Name: JIT_AutoProvisioning_NoPasswd

Description: JIT AutoProvisioning With No Password

* Authentication Level: 2

Default:

* Challenge Method: FORM

Challenge Redirect URL: /oam/server

* Authentication Module: AutoProvisioning_NoPasswd

* Challenge URL: /pages/login.jsp

* Context Type: default

* Context Value: /oam

Challenge Parameters: initial_command=NONE

For details about the parameters, see [Authentication Schemes and Pages](#)

- Under Challenge Parameters, add `initial_command=NONE`.

40.3.3 Protecting Resources with AutoProvisioningScheme (No Password Prompt)

Complete the configuration for the Just-in-Time Auto-provisioning (no password prompt), by assigning the AutoProvisioningScheme to the protected resource policy.

- From the **Application Security** Launch Pad, select **Application Domains** under **Access Manager**.
- Search and open the required **Application Domain**.
- Open the **Authentication Policies** tab and click **Protected Resource Policy**.
- Select the **AutoProvisioningScheme** from the **Authentication Scheme** dropdown list and click **Apply**.

Part X

Using Identity Context

Identity Context is considered as the environment and circumstances surrounding a user's request to access a particular protected resource.

It can be a sphere of activity, a geographical region, a communication platform, an application, or a logical or physical domain. Here is an introduction to Oracle Identity Context. This section contains the following chapter:

- [Using Identity Context](#)

41

Using Identity Context

Identity Context allows organizations to meet growing security threats by leveraging the context-aware policy management and authorization capabilities built into the Oracle Access Management platform.

Identity Context secures access to resources using traditional security controls (such as roles and groups) as well as dynamic data established during authentication and authorization (such as authentication strength, risk levels, device trust and the like).

This section describes the following topics:

- [Introducing Identity Context](#)
- [Understanding Identity Context](#)
- [Working With the Identity Context Service](#)
- [Identity Context API](#)
- [Configuring the Identity Context Service Components](#)
- [Validating Identity Context](#)

41.1 Introducing Identity Context

Identity Context is considered the environment and circumstances surrounding a user's request to access a particular protected resource. It can be a sphere of activity, a geographical region, a communication platform, an application, or a logical or physical domain.

Over the last decade, changes have been made to enterprise application infrastructures in order to web-enable the business applications that these infrastructures support. The changes allow for access by a greater number of users using different types of devices. To compensate for the additional risk associated with the greater number of users, the underlying security models used for access management have evolved from a silo-based implementation to a more dynamic one in which identity and risk data is shared across components of the entire application delivery process. This dynamic implementation relies on systems that offer Web single sign-on (SSO), fine-grained authorization, Web Services Security, Identity Federation and the like to aggregate security controls within a particular runtime deployment environment (web server or application server container) and provide policy-based security controls to manage access to application resources. Additionally, the identity and risk data provides a context for the user who is requesting access.

Initially, application security controls focused on unifying silos within a specific enterprise application deployment paradigm (for example, all web server applications, all web services applications, or all application server applications) but a growing presence of external and internal security threats now requires the unification of disparate security models in order to properly manage the greater amount of risk.

This requirement is further magnified by the advent of the cloud and mobile computing paradigm in which applications are no longer made up of components running neatly in the protected confines of a secure enterprise.

The ability of applications to leverage cloud services comes at the cost of having to account for the greater amount of risk stemming from those services being silos in their own way. With the number of threats to cloud deployments and mobile delivery channels growing steadily, it is required for the end-to-end application delivery process to implement the necessary policy controls for dealing with the greater range of threats. These policy controls require access to information about the requesting user on the basis of which security decisions must be made. Thus, a security policy management infrastructure must be context-aware to allow for an Administrator to create policy that controls the level of security imposed on a user who is requesting access to a protected application environment.

Previously, Identity Context was defined by the presence of an identity record in one or more identity stores (such as an LDAP directory or a SQL database). The identity record includes profile attributes, groups of which the user is a member, and enterprise roles. However, the constantly expanding reach of web, cloud, and mobile application delivery channels requires authorization policy controls that are aware of more dynamic information regarding the identity. This information is associated with the identity attempting to access the protected resource and may include some or all of the following:

- Presence (location, historical patterns)
- Authentication strength (weak, strong)
- Level of Assurance (NIST levels, X509 certificates)
- Risk Assessment (pattern analysis)
- Federation (partner attributes)
- Device characteristics (fingerprint, device health, device protection, trusted data)
- Assertions from trusted partners (SAML tokens, etc.)
- Single Sign On sessions (session time outs)

The following examples illustrate how Identity Context data might be used by an application. The application might:

- Disable a particular business function if the user is not authenticated using a strong credential such as smart card.
- Secure access to a transaction based on the identity data supplied by a business partner (via Identity Federation) with whom the organization does business.
- Request additional authentication credentials if it detects that access is originating from a location known for fraudulent activities.
- Limit the scope of administrative authority if the Administrator's industry certification (as maintained by a third party) has expired.
- Disable certain business functions if it detects that access is originating from an unknown device.

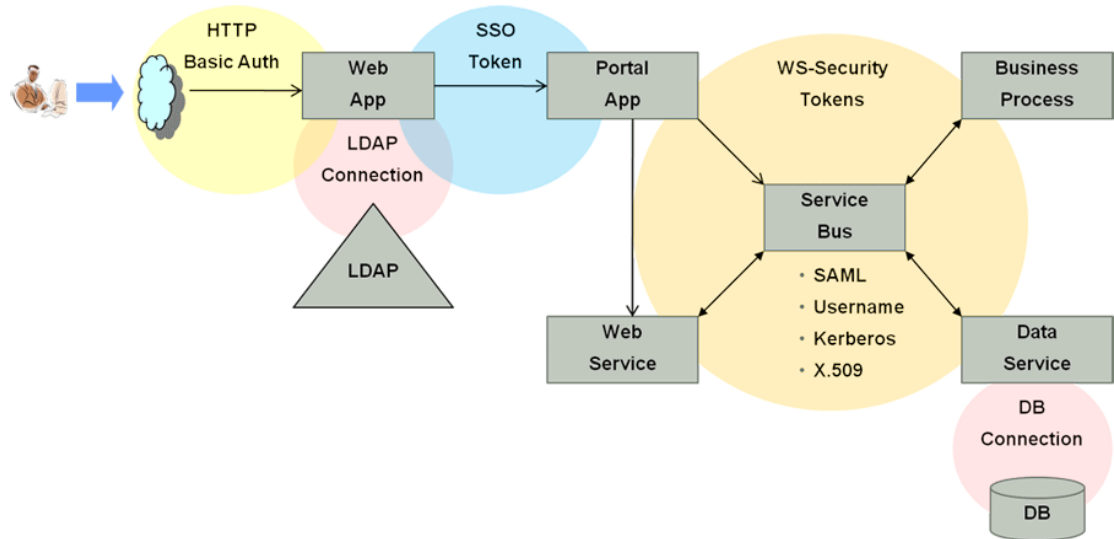
By incorporating the concept of Identity Context into access management, control can now be determined using dynamic data that is not necessarily contained in an identity profile (referred to as Identity Context attributes).

41.2 Understanding Identity Context

Access Manager enables context-aware access management by incorporating Identity Context as a built-in service of the Oracle Access Management platform.

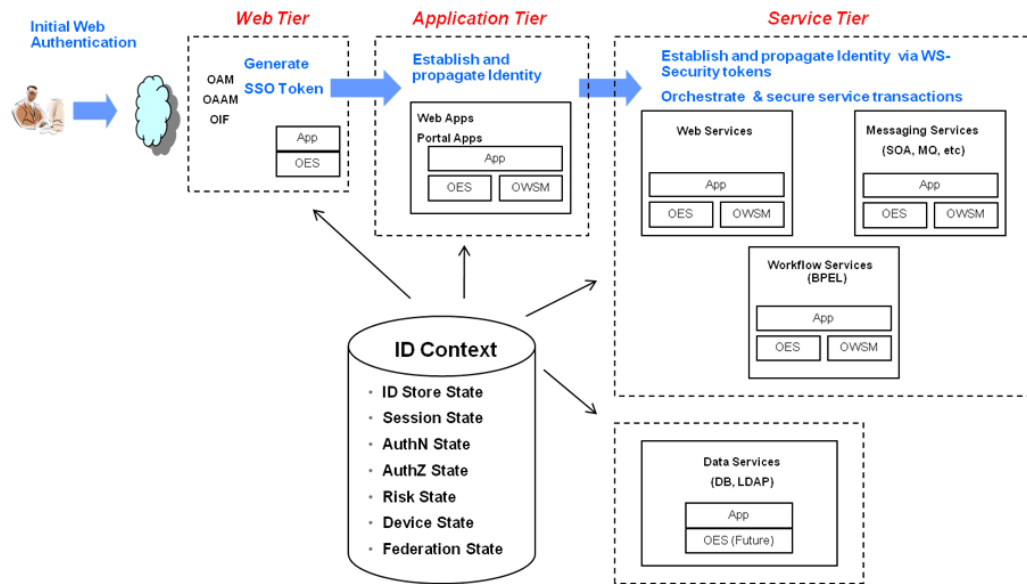
Figure 41-1 illustrates the flow of the Identity Context process, implemented by multiple system components. Each application delivery component has its own security policy infrastructure responsible for protecting its individual slice of the application. This specific use case involves the end user device, a Web Server running static GUI pages, an Application Server running the Portal Server rendering dynamic content, a Service Bus Server exposing the Web service endpoint, a database server containing transactional data, and an LDAP server containing identity profile data.

Figure 41-1 End to End Identity Context Process



Each component of the process has its own security infrastructure where the authorization policies governing access to protected resources are defined administratively and enforced at runtime. Additionally, some or all of the components may have externalized policy management to an external authorization server such as Oracle Entitlements Server - which is the case if the applications were built leveraging Oracle Platform Security Services. Figure 41-2 illustrates the functional architecture of Identity Context based on the Oracle applications of which it is comprised.

Figure 41-2 End To End Identity Context Process Components



As seen in the illustrations, context-aware security policy management is achieved by leveraging the Oracle Access Management platform. This platform contains native support for working with and enforcing Identity Context attributes (including risk score, trusted device data, authentication data, and the like) without changing end-user applications.

41.3 Working With the Identity Context Service

The Oracle Access Management platform enables Identity Context data to be collected, propagated across the involved components, and made available for granting or denying authorization to access protected resources.

For more information, see [Figure 41-2](#). The Identity Context Service allows access to the Identity Context Runtime through the Identity Context API. The Identity Context Dictionary schema specifies the Identity Context attributes.

This section describes the following topics:

- [Identity Context Dictionary](#)
- [Identity Context Runtime](#)

41.3.1 Identity Context Dictionary

At the core of the Identity Context architecture is the Identity Context Dictionary. The dictionary defines the Identity Context schema by specifying the identity context attributes as defined by the Oracle Access Management platform.

The Schema describes each attribute with a unique name that equals *namespace* : *attribute*. [Table 41-1](#) documents the Schema attributes.

 **Note:**

Virtual attributes (as documented in [Table 41-1](#)) represent an abstract class of identity information from which specific attributes are created. When publishing virtual attributes, the Identity Context API expects the attribute value to contain *attr-name=attr-value*. The actual attribute will be created using the name *namespace : attribute : attr-name* and a value of *attr-value*. This approach allows the publication of attributes whose value comes from a source not directly managed by the Oracle Access Management components.

Table 41-1 Identity Context Schema Attributes

Namespace	Attribute	Type	Virtual	Primary Publisher	Description
oracle:ldm:claims:nameid	value	string	no	OAM	Indicates a unique user identifier. Access Manager currently publishes User DN
oracle:ldm:claims:nameid	format	string	no	OAM	Indicates the type of user identifier. Access Manager currently publishes "urn:oasis:names:tc:SAML:1.1:nameid-format:x509SubjectName"
oracle:ldm:claims:nameid	qualifier	string	no	OAM	Indicates a logical Identity Domain to whom the user belongs. Access Manager currently publishes a logical name of the identity store, such as UserIdentityStore1.
oracle:ldm:claims:nameid	sprovidedid	string	no	OAM	Indicates unique identifier that can be used by any SP to locate the user in SP's own identity store(s). Access Manager currently publishes the value of the unique id attribute as configured in a registered identity store.
oracle:ldm:claims:client	firewallenabled	boolean	no	OESSO	Indicates client device has firewall enabled.
oracle:ldm:claims:client	antivirusenabled	boolean	no	OESSO	Indicates client device has antivirus enabled.
oracle:ldm:claims:client	fingerprint	string	no	OESSO, Oracle Access Management Mobile and Social (OMS)	Indicates fingerprint of the client device.
oracle:ldm:claims:client	ostype	string	no	OMS	Indicates client device's Operating System type.
oracle:ldm:claims:client	osversion	string	no	OMS	Indicates client device's operating system version.

Table 41-1 (Cont.) Identity Context Schema Attributes

Namespace	Attribute	Type	Virtual	Primary Publisher	Description
oracle:ldm:claims:c lient	jailbroken	boolean	no	OMS	Indicates if client device is Jailbroken (iOS) or Rooted (Android).
oracle:ldm:claims:c lient	macaddress	string	no	OMS	Indicates client device's Ethernet (MAC) Address.
oracle:ldm:claims:c lient	ipaddress	string	no	OMS	Indicates client device's Client IP Address.
oracle:ldm:claims:c lient	vpnenabled	boolean	no	OMS	Indicates if client's device has VPN enabled.
oracle:ldm:claims:c lient	geolocation	string	no	OMS	Indicates client device location's geographical coordinates in the form of "latitude,longitude".
<hr/> <hr/> <hr/>					
oracle:ldm:claims:s ession	authnlevel	integer	no	OAM	Indicates authentication level for Access Manager
oracle:ldm:claims:s ession	usercount	integer	no	OAM	Indicates number of sessions held by the users
oracle:ldm:claims:s ession	appdomain	string	no	OAM	Indicates name of the Access Manager Application Domain containing policies
oracle:ldm:claims:s ession	apppolicy	string	no	OAM	Indicates name of the Access Manager policy that allowed access
oracle:ldm:claims:s ession	appagent	string	no	OAM	Indicates the name of the agent from which the request came to Access Manager
oracle:ldm:claims:s ession	appclientip	string	no	OAM	Indicates the IP address of the client sending the request to Access Manager
oracle:ldm:claims:s ession	sessionid	string	no	OAM	Indicates the Access Manager session ID
oracle:ldm:claims:s ession	attributes	string	yes	OAM	Indicates session attributes as retrieved from the session store. For example, in Access Manager, select "oracle:ldm:claims:session:attributes" as the claim name and then specify the session attribute using the following notation: "attr-name=\$session.attr.name" where <i>name</i> is the name of the attribute stored in the session. The claim will be created with the name of "oracle:ldm:claims:session:attributes:attr-name" and value equal to session's <i>name</i> attribute.

Table 41-1 (Cont.) Identity Context Schema Attributes

Namespace	Attribute	Type	Virtual	Primary Publisher	Description
oracle:ldm:claims:f ed	partner	string	no	OAM--or IF?	Indicates partner ID as determined by Identity Federation
oracle:ldm:claims:f ed	nameidvalue	string	no	OAM--or IF?	Indicates user ID from a federation partner as determined by Identity Federation
oracle:ldm:claims:f ed	nameidformat	string	no	OAM--or IF?	Indicates format of the user ID from a federation partner as determined by Identity Federation
oracle:ldm:claims:f ed	attributes	string	yes	OAM	Indicates federation attribute as supplied by the partner and determined by Identity Federation. For example, in Access Manager, select "oracle:ldm:claims:fed:attributes" as the claim name and then specify the federation attribute using the following notation: " <i>attr-name</i> =\$session.attr.fed.attr. <i>name</i> ", where <i>name</i> is the name of the SAML attribute in the partner's SAML assertion. The claim will be created with the name of "oracle:ldm:claims:fed:attributes: <i>attr-name</i> " and value equal to the partner's assertion provided in the SAML's <i>name</i> attribute.
oracle:ldm:claims:i ds	attributes	string	yes	OAM	For example, in Access Manager, select "oracle:ldm:claims:ids:attributes" as the claim name, and then specify the ID Store attribute using the following notation: " <i>attr-name</i> =\$user.attr. <i>name</i> " where <i>name</i> is the name of the attribute on the user profile. The claim will be created with the name of "oracle:ldm:claims:ids:attributes: <i>attr-name</i> " and value equal to user profile's <i>name</i> attribute.
oracle:ldm:claims:t enant	tenantid	string	no	OAM	Currently reserved for future use. (Indicates tenant id.)
oracle:ldm:claims:t enant	attributes	string	yes	OAM	Currently reserved for future use. (Indicates tenant attributes as supplied by the Publisher. The claim value is meant to contain " <i>attr-name</i> = <i>attr-value</i> ". The claim will be created with the name of "oracle:ldm:claims:tenant: <i>attr-name</i> " and value of <i>attr-value</i> .)

41.3.2 Identity Context Runtime

Identity Context Runtime refers to a collection of Identity Context attributes (as defined in the Identity Context Dictionary) that is asserted by various trusted application components and/or security frameworks known to be authoritative for the attributes; this is the Oracle Access Management platform. Runtime context represents current surroundings, circumstances, environment, background, or settings which determine, specify, or clarify the meaning of an event for an identity in the runtime application environment.

The Oracle Access Management platform leverages a common infrastructure component called the Context Management Engine (CME). CME ensures that an Identity Context is generated for every transaction that is processed through the Oracle Access Management platform. The context data gathered by CME applies to transactions a user performs over the web channel or web service channel and using many of the software products available in the Oracle Access Management platform. Some transactions that are initiated on the back end may also require access to Identity Context, and may require Identity Context to be persisted for some duration of time.

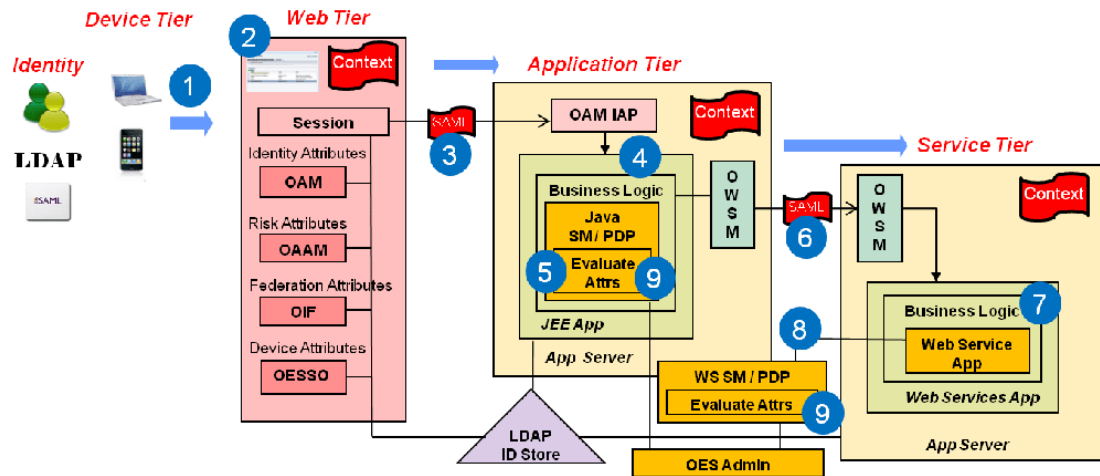
In a typical Oracle middleware deployment the Identity Context Runtime will be utilized primarily by the Oracle Access Management platform to perform policy-based decisions on behalf of protected applications. However, it is also possible for any applications running in the container to directly integrate with, and consume, the Identity Context Runtime by leveraging the Identity Context API. The amount of available Identity Context data will vary depending on what products have been deployed. There will be a default set of Identity Attributes that will be available out-of-the-box, which are mainly configured in the Access Manager by leveraging the Identity Assertion. [Table 41-1](#) documents these default attributes. The following list provides details on the end-to-end flow of the Identity Context Runtime. [Figure 41-3](#) below the list illustrates the flow.

Process overview: End-to-end flow of the Identity Context Runtime

1. User accesses a protected application from a device.
2. Access Manager asserts the identity, collects Identity Attributes from the participating Access Management publishing components and creates an Identity Context.
3. Access Manager generates an Identity Assertion (a SAML Session token) and incorporates the Identity Context attributes. The Access Manager Identity Asserter processes the Identity Assertion and publishes the Identity Context to the WebLogic Server container using the OPSS Attribute Service.
4. The protected application calls the OES PEP API to make an authorization decision. OES automatically propagates the Identity Context to the local OES PDP.
5. OES finds the appropriate Authorization Policy and evaluates it's Conditions (based on the Identity Context attributes). Evaluation can be done using a built-in Identity Context function or a custom function.
6. The protected application makes a JRF web service call in which the Oracle Web Service Manager (OWSM) client uses the SAML token to propagate Identity Context into the Web Service application environment.

7. OWSM (on the web service side) processes the SAML assertion with the Identity Context and publishes the Identity Context to the WebLogic Server container by using the OPSS Attribute Service.
8. Web Service application calls OES PEP API to make an authorization decision.
9. OES automatically propagates Identity Context to the remote OES PDP where conditions based on Identity Context attributes are evaluated using a built-in Identity Context function or a custom function.

Figure 41-3 Identity Context Process Flow



Once CME propagates Identity Context into the application tier and underlying Application Server container, the Identity Context is then made available to the container and applications running in it. Table 41-2 documents which Access Management platform products do what when working with Identity Context.

Table 41-2 Mapping Identity Context Operations

Role and Context Operation	Description	Components
Publisher - publishes Identity Context	Trusted security framework protecting an application component obtains from another trusted security framework, or derives from the information available to it, suitable facts about the identity and/or identity's access request. The information collected by the authoritative component is based on the environmental context available to component's runtime framework. For example, Access Manager determines the user's level of authentication strength and OESSO determines whether or not the client device has a firewall enabled.	<ul style="list-style-type: none"> • OAM – Session, Federation, and identity store attributes • OESSO – Device attributes • OMS Mobile SDK - Device attributes

Table 41-2 (Cont.) Mapping Identity Context Operations

Role and Context Operation	Description	Components
Propagator - propagates Identity Context	Trusted security framework propagates Identity Context attributes for use by another application security framework or directly by the application. For example, Access Manager propagates Identity Assertion (SAML token) for with the authenticated user's unique id and authentication level, and OWSM client propagates the current Identity Context over to the web service where OWSM agent will rebuild Identity Context in the web service application.	<ul style="list-style-type: none"> • OAM is between Web tier and container tier • OWSM is between web service client tier and web service tier • OPSS is between Access Manager Identity Asserter or OWSM agent and WebLogic Server container • OMS is between the OMS Mobile SDK and Access Manager
Evaluators - evaluate Identity Context	Trusted security framework or end-user application using Identity Context attributes to perform policy decisions or personalize application business logic. For example, when Identity Federation in Access Manager is configured, the application uses a partner-supplied assertion (available in the Identity Context) to authorize access to a transaction using OES.	<ul style="list-style-type: none"> • OAM – Web Perimeter Policy • OWSM – Web Service policy • OES – App-specific or WLS-specific policy for all PEP API calls made from the container where Identity Context exists. This includes all ADF apps, IAM apps, custom apps, etc.

41.4 Identity Context API

The Identity Context API is a set of Java classes designed to work with the Identity Context Dictionary and Identity Context Runtime.

The API is delivered as `IdentityContext.jar`, a part of Oracle Java Required Files (JRF). The following example illustrates an application working with Identity Context Dictionary.

Exmapple 55-1: Working with Identity Context Dictionary

```
// Display Identity Context Dictionary
try {
    ClaimDictionary idCtxDict = new ClaimDictionary();
    System.out.println
        ("IDC Dictionary :" + idCtxDict.getClaimCount() + "attributes");
    Iterator<String> iterNamespace = idCtxDict.getAllNamespaces();
    while (iterNamespace != null && iterNamespace.hasNext()) {
        String namespace = iterNamespace.next();
        System.out.println("Namespace : " + namespace);
        Iterator<ClaimSchema>
            iterClaimSchema=idCtxDict.getClaimsForNamespace(namespace);
        while (iterClaimSchema != null && iterClaimSchema.hasNext()) {
            out.println(iterClaimSchema.next().getUniqueName());
        }
    }
}
```

```

} catch (Exception e) {
    System.out.println("Unable to acquire IDC Dictionary. " + toString());
}

```

Applications work with the Identity Context Runtime to obtain the runtime state of the Identity Context as it currently exists in the application infrastructure. In order to work with the Identity Context Runtime, the protected application must be deployed to either a WebLogic Server domain built on Oracle Fusion Middleware PS5 with the OPSS Opatch for PS5, or Oracle Fusion Middleware PS6 or later.

Additionally, working with the Identity Context Runtime is a privileged operation that requires applications running in the WebLogic Server (with the required Identity Context support) to have proper source code grants. The privileged application, running in the WebLogic Server container, can then access the Identity Context Runtime by requesting it from the OPSS Attribute Service. The following example demonstrates how to use WLST to grant the OPSS Attribute Service permission to access an application (in this case, `ssofilter.jar`).

Using WLST To Grant Attribute Service Access To Application

```

# sh ../oracle_common/common/bin/wlst.sh
connect ('<username>', '<password>', 't3://localhost:7001')
grantPermission(codeBaseURL="file:${common.components.home}/
    modules/oracle.ssofilter_11.1.1/ssofilter.jar",
    permClass="oracle.security.jps.service.attribute.AttributeAccessPermission",
    permTarget="*", permActions="get, set, remove")
exit()

```

The following example illustrates an application working with Identity Context Runtime.

Working with Identity Context Runtime

```

import java.security.AccessController;
import java.security.PrivilegedAction;
import oracle.security.jps.internal.api.runtime.AppSecurityContext;
import oracle.security.idm.IdentityContext;

...

// get runtime ID Context from OPSS
private static Object getIDContext() {
    Object idc = AccessController.doPrivileged(new PrivilegedAction<Object>() {
        public Object run() {return
AppSecurityContext.getSecurityContext().getAttribute
    (oracle.security.idm.IdentityContext.Constants.IDC_API_ID); });
    return idc;
}

...

// Display runtime ID Context
try {
    Context idCtx = (Context)getIDContext();
    if (idCtx != null) {
        System.out.println("IDC Runtime :" + idCtx.getSize() + "attributes");
        Iterator<Claim> i = idCtx.getClaims();
        while (i != null && i.hasNext()) {
            Claim c = i.next();
            System.out.println(c.getName() + " : " + c.getValue());
        }
    } else {
        System.out.println("Identity Context Runtime is not available");
    }
}

```

```
    }  
  } catch (Exception e) {  
    System.out.println("Unable to acquire Identity Context Runtime. " +  
e.toString());  
  }  
  }  
  ...  
  
  // Obtain few attributes from Identity Context Runtime  
  Attr authnLevel = ctx.getAttr (Constants.ATTR_SESSION_AUTHN_LEVEL);  
  Attr isFirewallEnabled = ctx.getAttr(Constants.ATTR_CLIENT_FIREWALL_ENABLED);  
  Attr isTrustedDevice = ctx.getAttr(Constants.ATTR_RISK_TRUSTED_DEVICE);  
  
  // Use user's authentication strength established at login by OAM  
  int authLevel = new Integer(authnLevel.getValue()).intValue();  
  if (authLevel < 20) {  
    // do something  
  }  
}
```

41.5 Configuring the Identity Context Service Components

Each Identity Context Service component must be configured to accommodate business requirements. Support for Identity Context is pre-integrated into each participating Oracle Access Management component. A high-level overview of the necessary Identity Context configurations is provided here. However, detailed information can be found in documentation accompanying individual products. See [Table 41-2](#)

This section describes the following topics:

- [Configuring Oracle Fusion Middleware](#)
- [Configuring Access Manager](#)
- [Configuring Web Service Security Manager](#)
- [Configuring Oracle Entitlements Server](#)
- [Configuring Oracle Enterprise Single Sign On](#)
- [Configuring Secure Identity Context Propagation](#)

41.5.1 Configuring Oracle Fusion Middleware

The application to be protected must be deployed in a WebLogic Server domain built on Oracle Fusion Middleware 11.1.1 patch set 5 (PS5) with the Oracle Platform Security Services (OPSS) Opatch for PS5 or, Oracle Fusion Middleware PS6 or later.

The WebLogic Server domain in which the application is running must be protected by the Access Manager Identity Asserter component that will validate the Identity Assertion received from Access Manager and start the process of creating the Identity Context Runtime. The Access Manager Identity Asserter must be configured to detect the token type, OAM_IDENTITY_ASSERTION. Also, the protected application working with the Identity Context Runtime directly must be granted source code grants to work with the OPSS Attribute Service.

 **See Also:**

Securing Applications with Oracle Platform Security Services for more information on configuring Access Manager Identity Asserter, as well a source code grants.

41.5.2 Configuring Access Manager

As the main publisher and propagator of Identity Context, OAM serves as the central configuration point for collecting Identity Context data from its participating components.

The following sections describe key elements of the architecture behind Identity Context management.

- [Identity Assertion](#)
- [Federation Attributes](#)
- [Session Attributes](#)
- [Identity Store Attributes](#)

41.5.2.1 Identity Assertion

Oracle recommends that you define Asserted Attributes in Access Manager Authorization policies for proper enforcement of end-to-end security between the Web and application tiers. In addition to ensuring trust between the WebGate protecting a Web resource and the Application Server container, Identity Assertion (a SAML Session token) is used to publish the Identity Context data as SAML attributes.

Identity Assertion must be enabled and populated with Asserted Attributes as required by the business logic expecting specific attributes in the Identity Context. It is configured within the OAM Policy Responses tab and can be defined for both Authentication and Authorization policies.

 **See Also:**

Access Manager Identity Assertion and Asserted Attributes ([Table 25-25](#)).

41.5.2.2 Federation Attributes

Once a resource is protected by the Access Manager authentication scheme FederationScheme, Access Manager will act as the service provider and receive the SAML assertion as provided by the federation partner.

After the federation single sign on (SSO) operation, the following attributes will be present in the authenticated identity's Access Manager session:

- `$session.attr.fed.partner` (contains the partner name)
- `$session.attr.fed.nameidvalue` (contains the SAML NameID Value)
- `$session.attr.fed.nameidformat` (contains the SAML NameID Format)

- one `$session.attr.fed.attr.name` entry per SAML Attribute (contained in the SAML Assertion received from the partner)

These federation attributes can be used in configuring an Identity Assertion by selecting `oracle:idm:claims:fed:attributes` as the Asserted Attribute, and setting the value to "`attr-name=$session.attr.fed.attr.name`" where `attr-name` is the name given to the Identity Context attribute and `name` is the name of the SAML attribute in the partner's SAML assertion.

For example, defining `oracle:idm:claims:fed:attributes` with the value of `partner-role=$session.attr.fed.attr.role` will result in the creation of the Identity Context attribute `oracle:idm:claims:fed:attributes:partner-role` having a value of "manager" (assuming `$session.attr.fed.attr.role` contains "manager" as specified in the partner's SAML assertion for the SAML attribute "role").

41.5.2.3 Session Attributes

Access Manager session attributes can be used in configuring Identity Assertion by selecting `oracle:idm:claims:session:attributes` as the Asserted Attribute and setting the value to "`attr-name=$session.attr.name`" where `attr-name` is the name given to Identity Context attribute and `name` is the name of the Access Manager session attribute.

For example, defining `oracle:idm:claims:session:attributes` with the value of `authn-strength=$session.attr.authnlevel` will result in the creation of the Identity Context attribute `oracle:idm:claims:session:attributes:authn-strength` having a value as defined by the authentication scheme used during the login process.

41.5.2.4 Identity Store Attributes

Identity Store attributes can be used to configure an Access Manager Identity Assertion by selecting `oracle:idm:claims:ids:attributes` as the Asserted Attribute and setting the value to "`attr-name=$user.attr.name`" where `attr-name` is the name given to the Identity Context attribute and `name` is the name of the Identity Store attribute.

For example, defining `oracle:idm:claims:ids:attributes` with the value of `first-name=$user.attr.fname` will result in the creation of the Identity Context attribute `oracle:idm:claims:ids:attributes:first-name` having a value from the user's `fname` attribute as maintained in the identity store.

41.5.3 Configuring Web Service Security Manager

You can enable Oracle Web Service Security Manager (OWSSM) to propagate Identity Context.

To configure Web Service Security Manager for Identity Context:

1. Configure Security Policy by modifying the Identity Context supported OWSSM security policies to contain the `propagate.identity.context` element with a value of `true`.

 **Note:**

`propagate.identity.context` (by default, *false*) is a configuration override property on SAML related policies. To enable it globally, configure a global policy with the property set to *true*.

2. Configure the Keystore and Credential Store to sign the SAML assertion and messages: copy the updated Keystore and Credential Store to your `$DOMAIN_HOME/config/fmwconfig/` directory.

41.5.4 Configuring Oracle Entitlements Server

Runtime integration with Oracle Entitlements Server (OES) is fully automated.

When an application invokes the PEP API to make an authorization call, the PEP API automatically propagates the entire Identity Context Runtime to the OES PDP where Conditions (the policy objects that define the Identity Context) are evaluated.

 **Note:**

When making authorization calls, ensure that the last argument passed into the `newPepRequest()` method is not null, and is at least an empty hashmap as shown in this example:

```
PepRequestFactory requestFactory =
    PepRequestFactoryImpl.getPepRequestFactory();
PepRequest request = requestFactory.newPepRequest (subject,
    action, resource, new HashMap<String, Object>());
PepResponse response = request.decide();
boolean isAuthorized = response.allowed();
```

Conditions are built, based on the Identity Context schema, by a security Administrator using the OES Administration Console. The following built-in functions are used to specify Conditions using Identity Context attributes:

- `ASSERT_IDENTITY_CONTEXT`
- `GET_STRING_IDENTITY_CONTEXT`
- `GET_INTEGER_IDENTITY_CONTEXT`
- `GET_BOOLEAN_IDENTITY_CONTEXT`

Custom OES functions receive the full Identity Context Runtime information as a well-known request attribute. This data structure can be converted into Identity Context Runtime using the Identity Context API. The following example shows a custom OES function creating a context from the received parameter.

Custom Function Creating Identity Context:

```
public OpssString GET_STRING_IDENTITY_CONTEXT_V2 (
    RequestHandle requestHandle,
    Object[] args,
    Subject subject,
```

```
Map roles,
Resource resource,
ContextHandler contextHandler) throws RuntimeException {

// Obtain string representation of the runtime ID Context from the request
handle.
Context runtimeCtx = null;
try {
AttributeElement ctxAttr = requestHandle.getAttribute
(Constants.IDM_IDC_API_ID, false);
if (ctxAttr != null) {
String ctxStr = (String) ctxAttr.getValue();
runtimeCtx = new Context(ctxStr);
} else {
throw new RuntimeException ("Unable to acquire ID Context from request
handle");
}
} catch (Exception e) {
throw new RuntimeException (e.toString());
}

...

// start using Context which now contains the same exact Identity Context
Runtime as was present in the application that made the PEP API call
...
}
```

41.5.5 Configuring Oracle Enterprise Single Sign On

As part of the Identity Context Service, Oracle Enterprise Single Sign-on (OESSO) can publish and propagate client-based Identity Context attributes.

Once full integration has been configured, client-specific Identity Context attributes (as documented in [Identity Context Dictionary](#)) will be sent by OESSO to OAM in the session initiation request together with the user credentials submitted in the access request.

After the request has been received, OESSO makes a call to an SSL-protected OAM REST API (previously configured by the OESSO Administrator and included as part of the OESSO client distribution). This API returns the OAM_ID cookie to OESSO. OESSO then propagates the valid OAM_ID cookie to the client browsers (Internet Explorer and Firefox) which enables OESSO resources to be protected and enables single sign-on (SSO) with those resources that are protected by the OAM Embedded Credential Collector. (This does not include resources that are protected by the Distributed Credential Collector.) OESSO then provides OAM credentials that are acceptable to the OAM Embedded Credential Collector as well as client context information in the payload.

 **Note:**

The payload is secured by:

- Generating a 16 byte Random Salt
- Generating a SHA-256 Hash using the 16 Byte Random Salt
- Encrypting the claims using the OAM password protected by OESSO

To configure OESSO to get attributes for Identity Context

1. Refer to "Installing Logon Manager Client-Side Software" in the Oracle Enterprise Single Sign-On Suite Plus Installation Guide for details on integrating OAM and OESSO.
2. See additional details in the Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide section "Oracle Access Management Support in Logon Manager".

41.5.6 Configuring Secure Identity Context Propagation

The `OAM_IDENTITY_ASSERTION` header is used to securely propagate the Identity of the end user to the OAM protected application. The value of the `OAM_IDENTITY_ASSERTION` header is a Base64 encoded signed SAML token containing the Identity Context. By default, the token is signed using a Self Signed Certificate generated during bootstrap when the Server was started for the first time. The signing key and certificate are stored in the OAM Keystore at `$DOMAIN_HOME/config/fmwconfig/.oamkeystore`.

This section provides steps to replace the out of the box signing key and certificate. It is recommended that you use a key and certificate issued in compliance with your enterprise security practices.

 **Note:**

It is recommended that you take a backup of the latest `.oamkeystore` under the name `.oamkeystore_new` before updating the `.oamkeystore`.

For example, domain home: `<mw>/domains/oam_domain`

Run the following commands to create certificate and key:

1. `openssl req -new -keyout aaa_key.pem -out aaa_req.pem -utf8 -sha256`

 **Note:**

The key must be generated using the same password as the `.oamkeystore` file.

2. `openssl genrsa -aes256 -out rootCA.key 4096`
3. `openssl req -x509 -new -nodes -key rootCA.key -days 7300 -sha256 -out aaa_chain.pem`

4. `openssl x509 -req -in aaa_req.pem -CA aaa_chain.pem -CAkey rootCA.key -CAcreateserial -sha256 -out aaa_cert.pem -days 500`
5. `openssl x509 -in aaa_cert.pem -inform PEM -out aaa_cert.der -outform DER`
6. `openssl pkcs8 -topk8 -nocrypt -in aaa_key.pem -inform PEM -out aaa_key.der -outform DER`

Run the following keytool commands to update the `.oamkeystore`:

1. `keytool -delete -alias assertion-cert -v -keystore oamkeystore_new -storepass <pswd> -storetype JCEKS`
2. `keytool -delete -alias assertion-key -v -keystore oamkeystore_new -storepass <pswd> -storetype JCEKS`
3. `keytool -list -v -keystore oamkeystore_new -storetype JCEKS -storepass <pswd>`
4. `keytool -importcert -file aaa_chain.pem -trustcacerts -storepass <pswd> -keystore oamkeystore_new -storetype JCEKS -alias assertion-cert`
5. `keytool -list -v -keystore oamkeystore_new -storetype JCEKS -storepass <pswd>`
6. `<JDK>/bin/java -cp importcert.jar oracle.security.am.common.tools.import certs.CertificateImport -keystore oamkeystore_new -privatekeyfile aaa_key.der -signedcertfile aaa_cert.der -alias assertion-key -storetype JCEKS`
7. `keytool -list -v -keystore oamkeystore_new -storetype JCEKS -storepass <pswd>`
8. **Copy the `oamkeystore_new` to the location**
`<DOMAIN_HOME>/config/fmwconfig/.oamkeystore cp oamkeystore_new <DOMAIN_HOME>/config/fmwconfig/.oamkeystore`
9. `keytool -list -v -keystore <DOMAIN_HOME>/config/fmwconfig/.oamkeystore -storetype JCEKS -storepass <pswd>`
10. **WLST command to save artifacts:**
`saveAccessArtifacts (domainHome="<mw>/domains/oam_domain", propsFile="<path>/dbprop.properties")`

 **Note:**

Content of `<path>/dbprop.properties`:

```
oam.entityStore.schemaUser=OAM_Schema_userName
oam.entityStore.schemaPassword=<pswd>
oam.entityStore.ConnectString=jdbc:oracle:thin:@dbhost:dbport/ServiceID
```

Once you have updated your Signing Certificate, you should update your Clients to use the updated Certificate to validate the `OAM_IDENTITY_ASSERTION` header value. If

you are using the Weblogic OAM Identity Asserter (`OAMIdentityAsserter`), then you must replace the validation certificate in the configuration.

For more information, see [Integrating Oracle ADF Applications with Access Manager SSO](#).

41.6 Validating Identity Context

You can ensure correct operation of the Identity Context with Access Manager.

To validate:

To validate your Identity Context operations

1. Perform the following to validate the Identity Assertion response that Access Manager is constructing.
 - a. Configure Access Manager to protect the `/testidc` resource with a WebGate agent and return the Identity Assertion with the desired Asserted Attributes as part of the Authorization response.
 - b. Use the OAM Tester to validate that the Identity Assertion is returned as an `OAM_IDENTITY_ASSERTION` attribute in response to the authorization request for `/testidc`.
2. Perform the following to validate that WebGate is creating an HTTP header that contains the Identity Assertion.
 - a. Ensure the `/cgi-bin/printenv.pl` script is protected by the same policy that protects the `/testidc` resource.

 **Note:**

`printenv.pl` ships as part of OHS and must have permission to execute. Any script to display header information can be used instead.

- b. Access the `printenv.pl` to trigger a login and display the HTTP headers.
 - c. Ensure that the `HTTP_OAM_IDENTITY_ASSERTION` header contains a SAML token with Asserted Attributes.

Part XI

Using the OAM Snapshot Tool

The OAM Snapshot tool helps administrators manage, migrate and update OAM deployments. The tool provides a way of managing OAM deployments across various infrastructures in a uniform manner utilizing Oracle Database backup and cloning solutions.

This topic contains the following sections

- [Introduction to the OAM Snapshot Tool](#)
- [Customizing the Paths for the OAM Snapshot Tool](#)
- [Performing Backup and Restore and Migration of OAM System Using Snapshot Tool](#)
- [Using the OAM Snapshot Tool](#)

Introduction to the OAM Snapshot Tool

This section provides an overview of the OAM Snapshot Tool.

Snapshot refers to the OAM system backup comprising policies, configurations, certificates, and database schemas.

OAM Snapshot Tool refers to a set of customizable scripts: `createSnapshot.sh` and `applySnapshot.sh` that provides the ability to perform backup and restore operations of the entire OAM Domain configurations.

The scripts are available as part of the OAM 12c PS4 Bundle Patch 2 (12.2.1.4.200327) installation and located at `$MW_HOME/idm/oam/server/upgrade/script` directory.

Note:

Out-of-the-box, the scripts have default values for domain paths, DB paths, and so on. Before you run the scripts, ensure you review and update the paths based on your customized requirement. For more information, see [Customizing the Paths for the OAM Snapshot Tool](#)

The OAM Snapshot tool can be used, for example, to create another identical OAM system on a separate infrastructure, upgrade from the existing older version of OAM (12c PS3) to a newer version of OAM (12c PS4) without service disruptions, migrate OAM domain and database schemas to a different infrastructure, and so on. For details, see [Using the OAM Snapshot Tool](#)

Also see Doc ID 2663163.1 at <https://support.oracle.com> for additional details about the Snapshot tool.

Operation Types Supported by the OAM Snapshot Tool

The OAM Snapshot Tool supports the following operations:

Note:

The operation type is specified within the `<backup/restore>.properties` file, which is provided as an input to the customizable scripts.

Backup

- This operation enables you take a backup (snapshot) of the OAM Domain.
- Operation type is specified using the parameter `OPERATION=BACKUP` in the `backup.properties` file.
- You can run `createSnapshot.sh <path>/backup.properties` to create a backup of the OAM Domain.

- See [Backing Up an OAM Deployment](#) for detailed instructions.

Restore

- This operation enables you to restore the OAM Domain on a new infrastructure using the backup (snapshot) created.
- Operation type is specified using the parameter `OPERATION=RESTORE` in the `restore.properties` file.
- You can run `applySnapshot.sh <path>/restore.properties` to restore the OAM domain on the new infrastructure.
- See [Rebuilding an OAM Deployment - Domain Only](#)

Migrate

- This operation enables you to take a backup of OAM Domain and Database Schemas from the source infrastructure and restore them on the target infrastructure.
- The operation type is specified using the parameter `OPERATION=MIGRATE` in the `backup.properties` file (on the source infrastructure) and `OPERATION=MIGRATE` in the `restore.properties` file (on the target infrastructure).
- You can run `createSnapshot.sh <path>/backup.properties` on the source infrastructure. This internally calls the following scripts in sequence:
 - `moveToDB.sh`
 - `datapump_exp.sh`

In addition to the OAM Domain backup, the scripts create the database `*.dmp` files on the source infrastructure. These database `*.dmp` files must be manually copied from the source `DATA_PUMP_DIR` to the target `DATA_PUMP_DIR`.

- You can run `applySnapshot.sh <path>/restore.properties` on the target infrastructure. This internally calls the following scripts in sequence:
 - `datapump_imp.sh`
 - `extract_from_db.sh`
- See [Migrating OAM Domain and Database Schemas](#) for detailed instructions.

43

Customizing the Paths for the OAM Snapshot Tool

Out-of-the-box, the scripts have default values for domain paths, DB paths, and so on. Before you run the scripts, ensure that you review and update the paths based on your customized requirement.

- The default path for the DOMAIN_HOME (where the OAM domains reside) is `<MW_HOME>/user_projects/domains/<DOMAIN_NAME>`. If your DOMAIN_HOME is different than the default path, you must update the path before you run the scripts. For details, see [Adding Custom DOMAIN_HOME Path in the OAM Snapshot Tool](#)
- The scripts assume that the Database binaries are available on the same infrastructure that the scripts are being run. If your Database is on a different infrastructure, you must install the DB binaries and update the path before you run the scripts. For details, see [Configuring OAM Snapshot Tool when the DB is on a Different Infrastructure](#)

43.1 Configuring OAM Snapshot Tool when the DB is on a Different Infrastructure

The OAM Snapshot Tool assumes that the database binaries are available on the same infrastructure from where the scripts are being run.

If your database binaries are on a different infrastructure, you must also install the database binaries on the source and the target infrastructures where the `createSnapshot.sh` and the `applySnapshot.sh` scripts are required to be run. After the DB installation, update the scripts and the properties file to point to the DB_HOME.



Note:

The binaries are required only to invoke the scripts, and has no direct co-relation with the DB Connect String.

The following steps provide instructions for the script files to work correctly:

1. Download the required versions of the database installation files.



Note:

If your database version is 12.2 or higher, you can use Instant Clients. If you are on any other database version, use the full client.

- For Full Client, download the installation files from [Oracle Database Software Downloads](#).
 - For Instant Client, download the `Basic`, `SQL*Plus`, and the `Tools` Instant Client Packages from [Instant Client for Linux x86-64 \(64-bit\) site](#). For example, download `instantclient-basic-linux.x64-12.2.0.1.0.zip`, `instantclient-sqlplus-linux.x64-12.2.0.2.0.zip`, and `instantclient-tools-linux.x64-12.2.0.1.0.zip`.
2. Install the database binaries on the source and the target Infrastructures where the scripts need to run.
 - For Full Client instructions, navigate to [Oracle Database Software Downloads](#) and under **Note** click **See All** against the version of the database that you need to install. A new page is displayed containing the link to the Installation guides.
After installation, skip to Step 5.
 - For Instant Client, unzip the downloaded files to a directory, for example, `/usr/local/oracle/dbclient`.
Unzipping the files creates a new folder under the specified directory, for example, `/usr/local/oracle/dbclient/instantclient_12_2`
 3. Create two folders `bin` and `lib` under `/usr/local/oracle/dbclient/instantclient_12_2`
 4. From `/usr/local/oracle/dbclient/instantclient_12_2` copy all the `*.so.*` files to the `lib` folder and the rest of the files to the `bin` folder.
 5. Navigate to `$MW_HOME/idm/oam/server/upgrade/script` directory on the source infrastructure and open the `datapump_exp.sh` in your preferred editor.
 6. Add `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib` line below the existing `export ORACLE_HOME=$ORACLE_HOME` line.

```
export ORACLE_HOME=$ORACLE_HOME
//New line specifying the library path
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

7. Similarly, open the `datapump_imp.sh` on the target infrastructure and add the library path.

```
export ORACLE_HOME=$ORACLE_HOME
//New line specifying the library path
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

8. Edit the `backup.properties` file on the source infrastructure and set `DB_HOME`. For example,

```
DB_HOME=/usr/local/oracle/dbclient/instantclient_12_2
```

9. Similarly, edit the `restore.properties` file on the target infrastructure and set `DB_HOME`. For example,

```
DB_HOME=/usr/local/oracle/dbclient/instantclient_12_2
```

The `createSnapshot.sh` and the `applySnapshot.sh` scripts run correctly.

Performing Backup and Restore and Migration of OAM System Using Snapshot Tool

The OAM Snapshot tool provides tooling to create a snapshot of the Oracle Access Management (OAM) Domain with all its configurations, persist it, and use it for creating fully functional OAM IDM Domain clones.

You can backup your entire OAM configuration so that it can be restored in case of a server or data-center crash. You can also move the OAM domain to a different infrastructure, leaving the OAM schemas in the same database. Alternatively, you can move both the OAM domain and the database schemas to a different infrastructure.

Topics

- [Backing Up an OAM Deployment](#)
- [Rebuilding an OAM Deployment - Domain Only](#)
- [Migrating OAM Domain and Database Schemas](#)

44.1 Backing Up an OAM Deployment

Backup an OAM deployment using the OAM Snapshot Tool.

The OAM Snapshot Tool allows you to create a backup of the following configurations:

- The OAM domain configuration only
- The OAM domain configuration along with the OAM database schemas.



Note:

At present it is only supported to backup and restore database schemas stored in an Oracle Database.

Follow these steps to create a backup of your OAM deployment:

Topics

- [Prerequisites for Backing up an OAM Deployment](#)
- [Creating a backup.properties file](#)
- [Running the createSnapshot.sh script](#)
- [Backing up Keystores and Custom Applications](#)

44.1.1 Prerequisites for Backing up an OAM Deployment

Follow the prerequisites prior to backing up an OAM deployment.

If your OAM is used with third party sites, for example with federation, then the OAM server ports on the source server should be load balanced using a load balancer. This is a requirement for deployment migration where the source will be replaced by the target or where the source will be load balanced with the target as a standby server. The load balancer should have a different `hostname.domain` to the OAM `hostname.domain`. Once the load balancer is configured, modify the settings in OAM as follows:

1. Launch a browser and access the OAM console (For example, `http://localhost:7001/oamconsole`). Login with the weblogic username and password.
2. In the OAM homepage navigate to **Configuration**.
3. In the **Configuration** page, from the **Settings** tile, select **View** and from the drop down menu select **Access Manager**.
4. In the **Access Manager Settings** tab, under **Load Balancing** modify the **OAM Server Host** and **OAM Server Port** fields to match the load balancer HTTP endpoint `hostname.domain` and port.
5. Under **WebGate Traffic Load Balancer** modify the **OAM Server Host** and **OAM Server Port**, to point to the Loadbalancer HTTP endpoint `hostname.domain` and port.
6. Click **Apply**.

 **Note:**

If using OAP (TCP) rather than OAP over HTTP then you must also update the SSO Agents with the load balancer `hostname.domain` and port:

1. Navigate to **Application Security**.
2. In the **Application Security** Launch Pad select **Agents**.
3. In the **Search SSO Agents** tab, under **Webgates** click **Search**.
4. Click on a WebGate to update.
5. In the webgate tab, under **Server Lists**, **Primary Server List** select **Add**.
6. Under **Access Server** select **Other** and enter the load balancer **Host Name** and **Host Port** respectively.
7. Click **Apply**.
8. Repeat the above steps for any other WebGates that need updating.

44.1.2 Creating a backup.properties file


Create a `backup.properties` file that contains the OAM deployment settings.

On the server where the OAM domain resides, create a `backup.properties` file that contains the OAM deployment settings. The entries to add to the properties file

depend on whether you are backing up the OAM domain only or also backing up the OAM database schemas. The following table describes all the parameters that can be added to the `backup.properties` file.

Parameter	Description	Sample Value
ORACLE_HOME	The path to the directory where OAM binaries are installed; same as MW_HOME.	/u01/oracle/product/OAM
DOMAIN_NAME	The name of the OAM WebLogic Server domain.	oam_domain
JAVA_HOME	The path to the JDK installed.	/u01/oracle/product/jdk8
CONNECT_STRING	The Oracle database connect string <code><hostname.domain>:<port>/<service_name></code> where the OAM schemas reside. Database where the domain is backed up.	sourcedb.example.com:1521 /oampdb.example.com
OAM_SCHEMA_USER	The OAM <code><schema_prefix>_OAM</code> schema.	DEV_OAM
OAM_SCHEMA_PASSWORD	The OAM <code><schema_prefix>_OAM</code> schema password.	<schema_password>
DB_SYS_USER	The database SYS user. This parameter is only required if you are also exporting the OAM database schemas.	SYS
DB_SYS_PASSWORD	The database SYS user password. This parameter is only required if you are also exporting the OAM database schemas.	<sys_password>

Parameter	Description	Sample Value
DATA_PUMP_DIR	<p>The alias of the Data Pump directory configured in the Oracle Database.</p> <p>This parameter is only required if you are also exporting the OAM database schemas.</p> <p>To view the current Data Pump alias and directory, or to configure a new one see Understanding Dump_ Log_ and SQL File Default Locations</p>	DATA_PUMP_DIR

 **Note:**

Make sure the directory is empty and no *.dmp files exist prior to running the create snapshot command in [Running the](#)

Parameter	Description	Sample Value
		<pre>create Snaps hot.sh script</pre>
DB_VERSION	<p>The version of the database to export the data to.</p> <p>For example, if both the source and target databases are running Oracle Database 12c Release 1 (12.1), then the Data Pump <code>DB_VERSION</code> parameter must be set to at least 12.0 or the <code>COMPATIBLE</code> database initialization parameter must be set to at least 12.0 or later.</p> <p>This parameter is only required if you are also exporting the OAM database schemas.</p>	12.1
OPERATION	<p>The operation to perform. If backing up the OAM domain only then the value is <code>BACKUP</code>. If backing up the OAM domain and database schemas then the value is <code>MIGRATE</code>.</p>	BACKUP
OAM_SCHEMA_PREFIX	<p>The OAM <code><schema_prefix></code>.</p>	DEV

Parameter	Description	Sample Value
DB_HOME	The path to the directory where the Database binaries are located. This parameter is only required if you are also exporting the OAM database schemas. The Database version must be same as the Database used in the CONNECT_STRING	/u01/oracle/product/database/12.2.1/db_1

 **Note:**

Database binaries must be on the same infrastructure from where the create Snapshot and applySnapshot scripts need to

Parameter	Description	Sample Value
		<p>be run . For mo re info rm atio n, see Co nfig urin g OA M Sn aps hot Too l wh en the DB is on a Diff ere nt Infr astr uct ure</p>
LOG_DIR	The path where the log output will be stored. Ensure that the directory exists and can be written to.	/tmp

An example backup.properties to back up the OAM domain only will look as follows:

```
ORACLE_HOME=/u01/oracle/product/OAM
DOMAIN_NAME=oam_domain
JAVA_HOME=/u01/oracle/product/jdk8
CONNECT_STRING=sourcedb.example.com:1521/oampdb.example.com
OAM_SCHEMA_USER=DEV_OAM
OAM_SCHEMA_PASSWORD=<schema_password>
OPERATION=BACKUP
OAM_SCHEMA_PREFIX=DEV
LOG_DIR=/tmp
```

An example `backup.properties` to back up the OAM domain and database schemas will look as follows:

```
ORACLE_HOME=/u01/oracle/product/OAM
DOMAIN_NAME=oam_domain
JAVA_HOME=/u01/oracle/product/jdk8
CONNECT_STRING=sourcedb.example.com:1521/oampdb.example.com
OAM_SCHEMA_USER=DEV_OAM
OAM_SCHEMA_PASSWORD=<schema_password>
DB_SYS_USER=sys
DB_SYS_PASSWORD=<sys_password>
DATA_PUMP_DIR=DATA_PUMP_DIR
DB_VERSION=12.1
OPERATION=MIGRATE
OAM_SCHEMA_PREFIX=DEV
DB_HOME=/u01/oracle/product/database/12.2.1/db_1
LOG_DIR=/tmp
```

44.1.3 Running the `createSnapshot.sh` script

Take a backup of the OAM deployment using the `createSnapshot.sh` script.

To run the backup script:

1. Shutdown the OAM AdminServer and Managed Servers on the source infrastructure.
2. Launch a terminal window on the OAM source server and navigate to the `$MW_HOME/oam/server/upgrade/script` directory.
3. Run `$ sh createSnapshot.sh <path>/backup.properties`.

The output for an OAM domain only backup will look similar to the following:

```
bash-4.1$ sh createSnapshot.sh backup.properties
Invoking moveToDB command : sh moveToDB.sh /u01/oracle/product/OAM
oam_domain /u01/oracle/product/jdk8
jdbc:oracle:thin:@sourcedb.example.com:1521/oampdb.example.com DEV_OAM
*****
Apr 08, 2020 7:34:55 AM oracle.security.am.artifacts.DBConfigCommand
moveToDBOperation
INFO: move to DB for Context : STATELESS_CONTEXT
Apr 08, 2020 7:34:55 AM
oracle.security.am.artifacts.common.DBConnection createConnection
INFO: Opening Connection
Apr 08, 2020 7:34:56 AM
oracle.security.am.artifacts.common.DBTableValidator validate
WARNING: during db-table validation, context_name: STATELESS_CONTEXT
.....
.....
.....
Apr 08, 2020 7:35:02 AM
oracle.security.am.artifacts.store.FileArtifactStore storeArtifacts
INFO: Successfully written file with name : opss-audit-jdbc.xml in
folder /tmp/tempDir/user_projects/domains/base_domain/config/jdbc/
Apr 08, 2020 7:35:02 AM
```

```
oracle.security.am.artifacts.store.ArtifactStore cleanup
INFO: Closing the stream for file opss-audit-jdbc.xml
Apr 08, 2020 7:35:02 AM
oracle.security.am.artifacts.processor.DBToFileArtifactProcessor postProcess
INFO: Successfully validated the checksum for artifact opss-audit-jdbc.xml
in folder /tmp/tempDir/user_projects/domains/base_domain/config/jdbc/
Apr 08, 2020 7:35:02 AM
oracle.security.am.artifacts.processor.DBToFileArtifactProcessor postProcess
INFO: Calculated Checksum [WR/Mjp9ls92UGZIEnmigAYDifPt9IwMhJ9xIMJtCa/U=].
Checksum in DB [WR/Mjp9ls92UGZIEnmigAYDifPt9IwMhJ9xIMJtCa/U=]
Apr 08, 2020 7:35:02 AM oracle.security.am.artifacts.DBConfigCommand
moveToDBOperation
INFO: Closing the DB Connection.
Apr 08, 2020 7:35:02 AM oracle.security.am.artifacts.common.DBConnection
close
INFO: Closing Connection
STATUS=SUCCESS
bash-4.1$
```

The output for an OAM domain and database schema backup will look similar to the following:

```
bash-4.2$ sh createSnapshot.sh backup.properties
Invoking moveToDB command : sh moveToDB.sh /u01/oracle/product/OAM
oam_domain /u01/oracle/product/jdk8
jdbc:oracle:thin:@sourcedb.example.com:1521/oampdb.example.com DEV_OAM
*****
Apr 02, 2020 4:16:45 AM oracle.security.am.artifacts.DBConfigCommand
moveToDBOperation
INFO: move to DB for Context : STATELESS_CONTEXT
Apr 02, 2020 4:16:45 AM oracle.security.am.artifacts.common.DBConnection
createConnection
INFO: Opening Connection
Apr 02, 2020 4:16:46 AM oracle.security.am.artifacts.common.DBTableValidator
validate
WARNING: during db-table validation, context_name: STATELESS_CONTEXT
Apr 02, 2020 4:16:46 AM oracle.security.am.artifacts.common.DBTableUtility
prepareSqlStatment
INFO: SCHEMA_NAME:DEV_OAM
Apr 02, 2020 4:16:46 AM DBTableValidator validate
INFO: Record Found in the database table.cohstore.jks
Apr 02, 2020 4:16:46 AM DBTableValidator validate
INFO: Successfully able to validate the table in the database
Apr 02, 2020 4:16:46 AM oracle.security.am.artifacts.DBConfigCommand
processRecord
INFO: Moving file ORACLE_HOME@/user_projects/domains/oam_domain/servers/
AdminServer/security/boot.properties with mandatory flag as :false for
context:STATELESS_CONTEXT
Apr 02, 2020 4:16:46 AM oracle.security.am.artifacts.handler.FileHandler
handle
INFO: File Handler: versionnull
Apr 02, 2020 4:16:46 AM
oracle.security.am.artifacts.processor.ArtifactProcessor execute
SEVERE: Request to process artifacts failed. Context=STATELESS_CONTEXT
```

```

.....
.....
.....
Connected to: Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
Starting "SYS"."SYS_EXPORT_SCHEMA_01": "sys/
*****@sourcedb.example.com:1521/oampdb.example.com AS SYSDBA"
schemas=DEV_WLS_RUNTIME directory=DATA_PUMP_DIR
dumpfile=DEV_WLS_RUNTIME.dmp logfile=DEV_WLS_RUNTIME.log Version=12.1
Estimate in progress using BLOCKS method...
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 0 KB
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/TABLESPACE_QUOTA
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/
INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
. . exported "DEV_WLS_RUNTIME"."ACTIVE"          0 KB
0 rows
Master table "SYS"."SYS_EXPORT_SCHEMA_01" successfully loaded/unloaded
*****
*****
Dump file set for SYS.SYS_EXPORT_SCHEMA_01 is:
/u01/oracle/product/database/12.2.1/db_1/dpdump/DEV_WLS_RUNTIME.dmp
Job "SYS"."SYS_EXPORT_SCHEMA_01" successfully completed at Thu Apr 2
11:37:56 2020 elapsed 0 00:00:19

bash-4.2$

```

 **Note:**

The `createSnapshot.sh` backs up the OAM domain configuration to a table in the `<schema_prefix>_OAM` schema. If you choose to also backup the database schemas, then all the OAM schemas are exported as dump files (*.dmp) into the directory on the database server referenced by the `DATA_PUMP_DIR` parameter.

44.1.4 Backing up Keystores and Custom Applications

Manually backup keystores and custom applications.

The OAM Snapshot tool backs up the OAM domain and optionally the database schemas. There are some extra manual steps that may be required depending on your configuration. If your configuration contains any of the following items then manually back up these files:

- Any java keystores or wallets you have configured on the source server that reside in any directory other than `$DOMAIN_HOME/config/fmwconfig`.
- Any custom war files deployed for custom login pages, custom error pages or custom UI pages.

44.2 Rebuilding an OAM Deployment - Domain Only

Rebuild an OAM Deployment (Domain Only) using the OAM Snapshot Tool.

Topics



Note:

Before following the steps below make sure you have followed the instructions in [Backing Up an OAM Deployment](#) to backup the domain only.

Follow these steps to rebuild your OAM deployment (Domain Only):

- [Installing the OAM Software in the New Infrastructure](#)
- [Creating a restore.properties file](#)
- [Running the applySnapshot.sh script](#)
- [Restoring Keystores and Custom Applications](#)

44.2.1 Installing the OAM Software in the New Infrastructure

Install the OAM software in the new target infrastructure.

Prior to installation refer to the `backup.properties` used to backup the OAM domain and note the PATH for the `JDK_HOME` and `ORACLE_HOME`. For the restore to be successful on the target infrastructure, the following OAM software must be installed in the same paths as the source infrastructure:

- Oracle JDK
- Oracle Identity and Access Management 12cPS4 Infrastructure
- Oracle Identity and Access Management 12cPS4

 **Note:**

The version of the software installed on the target has to be the exact same version as the source, or a later version. It is not possible to restore to a version of the software earlier than used on the source.

To install the software refer to Roadmap for Installing and Configuring a Standard Installation Topology

44.2.2 Creating a restore.properties file

Create a `restore.properties` file that contains the OAM deployment settings for the rebuild.

On the target server where the OAM domain will be restored, create a `restore.properties` file that contains the OAM deployment settings. The table below describes all the parameters that can be added to the properties file for restoring the OAM domain only.

Parameter	Description	Sample Value
ORACLE_HOME	The path to the directory where OAM binaries are installed; same as MW_HOME.	/u01/oracle/product/OAM
DOMAIN_NAME	The name of the OAM WebLogic Server domain.	oam_domain
JAVA_HOME	The path to the JDK installed.	/u01/oracle/product/jdk8
CONNECT_STRING	The Oracle database connect string <hostname.domain>:<port>/<service_name> where the OAM schemas reside.	sourcedb.example.com:1521/oampdb.example.com
OAM_SCHEMA_USER	The OAM <schema_prefix>_OAM schema.	DEV_OAM
OAM_SCHEMA_PASSWORD	The OAM <schema_prefix>_OAM schema password.	<schema_password>
DB_SYS_USER	The database SYS user.	SYS
DB_SYS_PASSWORD	The database SYS user password.	<sys_password>
OPERATION	The operation to perform. In this case for domain only, the operation is RESTORE.	RESTORE
OAM_SCHEMA_PREFIX	The OAM <schema_prefix>.	DEV
DB_HOME	The path to the directory where the Database binaries are located.	/u01/oracle/product/database/12.2.1/db_1

Parameter	Description	Sample Value
LOG_DIR	The path where the log output will be stored. Ensure that the directory exists and can be written to.	/tmp

An example restore.properties will look as follows:

```
ORACLE_HOME=/u01/oracle/product/OAM
DOMAIN_NAME=oam_domain
JAVA_HOME=/u01/oracle/product/jdk8
CONNECT_STRING=sourcedb.example.com:1521/oampdb.example.com
OAM_SCHEMA_USER=DEV_OAM
OAM_SCHEMA_PASSWORD=<schema_password>
DB_SYS_USER=sys
DB_SYS_PASSWORD=<sys_password>
OPERATION=RESTORE
OAM_SCHEMA_PREFIX=DEV
DB_HOME=/u01/oracle/product/database/12.2.1/db_1
LOG_DIR=/tmp
```

44.2.3 Running the applySnapshot.sh script

Restore the OAM Deployment (domain only) using the `applySnapshot.sh` script.

To run the restore script:

1. Launch a terminal window on the target server and navigate to the OAM `$MW_HOME/oam/server/upgrade/script` directory.
2. Run `$ sh applySnapshot.sh <path>/restore.properties`.

The output for an OAM domain only backup will look similar to the following:

```
bash-4.1$ sh applySnapshot.sh restore.properties
Invoking extract_from_db command : sh extract_from_db.sh /u01/oracle/
product/OAM oam_domain /u01/oracle/product/jdk8 sourcedb.example.com:1521/
oampdb.example.com /tmp DEV DEV_OAM ***** sys *****
Apr 08, 2020 7:56:27 AM oracle.security.am.artifacts.DBConfigCommand
extractFromDBOperation
INFO: extract from DB. Context=STATELESS_CONTEXT
Apr 08, 2020 7:56:27 AM oracle.security.am.artifacts.common.DBConnection
createConnection
INFO: Opening Connection
Apr 08, 2020 7:56:28 AM oracle.security.am.artifacts.common.DBTableValidator
validate
WARNING: during db-table validation, context_name: STATELESS_CONTEXT
Apr 08, 2020 7:56:28 AM oracle.security.am.artifacts.common.DBTableUtility
prepareSqlStatment
INFO: SCHEMA_NAME:DEV3_OAM
Apr 08, 2020 7:56:28 AM DBTableValidator validate
INFO: Record Found in the database table.cohstore.jks
Apr 08, 2020 7:56:28 AM DBTableValidator validate
INFO: Successfully able to validate the table in the database
```

```
.....  
.....  
.....  
SQL>  
0 rows updated.  
  
SQL>  
Table truncated.  
  
SQL> SQL> SQL> not spooling currently  
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release  
12.1.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real Application  
Testing options  
bash-4.1$
```

44.2.4 Restoring Keystores and Custom Applications

Manually restore keystores and custom applications

The OAM Snapshot tool restores the OAM domain only. If you backed up any of the files on the source server referenced in [Backing up Keystores and Custom Applications](#) then these must be restored to the same location on the target server prior to starting the OAM domain.



Note:

Before starting the OAM domain on the target infrastructure, the AdminServer and Managed Servers in the OAM domain on the source infrastructure must be shut down.

44.3 Migrating OAM Domain and Database Schemas

Migrate the OAM Domain and Database Schemas using the OAM Snapshot Tool.

Topics



Note:

Before following the steps below make sure you have followed the instructions in [Backing Up an OAM Deployment](#) to backup the domain and database schemas.

- [Installing and Creating a New Database](#)
- [Installing the OAM Software in the New Infrastructure](#)
- [Creating a restore.properties file](#)
- [Configuring OAM Snapshot Tool when the DB is on a Different Infrastructure](#)
- [Running the applySnapshot.sh script](#)

- [Restoring Keystores and Custom Applications](#)

44.3.1 Installing and Creating a New Database

Install database software on a new target server and create a database.

On the target infrastructure you must manually install the database software and create a new database. The following requirements must be met:

- The Oracle database software must be a version supported by Oracle Access Management.
- The Oracle database software can be an earlier version, the same version, or later version than the source server.
- If you install an earlier version of the supported Oracle database software, the `DB_VERSION` parameter in `backup.properties` must match the version of the target database, prior to backing up.
- The installation `PATH` can be the same as the source, or installed to a different `PATH`.
- When creating the new database instance, it should be the same specification, or a higher specification than the source database.
- The new database instance `service_name` can be the same as the source, or given a different name.

If the target database `service_name` is different than the source, run the following script after running `applySnapshot.sh`.

```
perl $MW_HOME/idm/oam/server/upgrade/script/  
replaceHostnames.pl $DOMAIN_HOME <OLD SERVICE NAME> <NEW_SERVICE_NAME>  
"DB"
```

- The new database instance must have the same `DATA_PUMP_DIR` configured.
- The database `*.dmp` files must be copied from the source `DATA_PUMP_DIR` to the target `DATA_PUMP_DIR`.
- After the database is created, the database and listener must be started prior to running the restore script.

44.3.2 Installing the OAM Software in the New Infrastructure

Install the OAM software in the new target infrastructure.

Prior to installation refer to the `backup.properties` used to backup the OAM domain and database schemas and note the `PATH` for the `JDK_HOME` and `ORACLE_HOME`. For the restore to be successful on the target infrastructure, the following OAM software must be installed in the same paths as the source infrastructure:

- Oracle JDK
- Oracle Identity and Access Management 12cPS4 Infrastructure
- Oracle Identity and Access Management 12cPS4

 **Note:**

The version of the software installed on the target have to be the exact same version as the source, or a later version. It is not possible to restore to a version of the software earlier than used on the source.

To install the software refer to Roadmap for Installing and Configuring a Standard Installation Topology


44.3.3 Creating a restore.properties file

Create a `restore.properties` file that contains the OAM deployment settings for rebuilding the OAM domain and database schemas.

On the target server where the OAM domain will be restored, create a `restore.properties` file that contains the OAM deployment settings. The table below describes all the parameters that need to be added to the properties file for restoring the OAM domain and database schemas.

Parameter	Description	Sample Value
ORACLE_HOME	The path to the directory where OAM binaries are installed; same as MW_HOME.	/u01/oracle/product/OAM
DOMAIN_NAME	The name of the OAM WebLogic Server domain.	oam_domain
JAVA_HOME	The path to the JDK installed.	/u01/oracle/product/jdk8
CONNECT_STRING	The target Oracle database <hostname.domain>:<port>/<service_name> where the OAM schemas will reside.	target.example.com:1521 /oampdb.example.com
OAM_SCHEMA_USER	The OAM <schema_prefix>_OAM schema.	DEV_OAM
OAM_SCHEMA_PASSWORD	The OAM <schema_prefix>_OAM schema password.	<schema_password>


Parameter	Description	Sample Value
SCHEMA_PASSWORD	This must have the same value as OAM_SCHEMA_PASSWORD.	<schema_password>

 **N**
o
t
e
:
It is mandatory to have both OAM_SCHEMA_PASSWORD

Parameter	Description	Sample Value
		D a n d S C H E M A _ P A S S W O R D f o r O P E R A T I O N = M I G R A T E .
DB_SYS_USER	The target database SYS user.	SYS
DB_SYS_PASSWORD	The target database SYS user password.	<sys_password>
DATA_PUMP_DIR	The alias of the Data Pump directory configured in the target Oracle Database. The directory referenced by the alias should only contain the *.dmp files copied across from the source database DATA_PUMP_DIR.	DATA_PUMP_DIR

Parameter	Description	Sample Value
OPERATION	The operation to perform. In this case the operation is MIGRATE.	MIGRATE
OAM_SCHEMA_PREFIX	The OAM <schema_prefix>.	DEV

Parameter	Description	Sample Value
DB_HOME	The path to the directory where the Database binaries are located. The Database version must be same as the Database used in the CONNECT_STRING	/u01/oracle/product/database/12.2.1/db_1

 **N**
o
t
e
:
D
a
t
a
b
a
s
e
b
i
n
a
r
i
e
s
m
u
s
t
b
e
o
n
t
h
e
s
a
m
e
i
n
f
r
a
s
t
r

Parameter	Description	Sample Value
		u c t u r e f r o m w h e r e t h e c r e a t e s n a p s h o t a n d a p p l y s n a p s h o t s c r i p t

Parameter	Description	Sample Value
		s n e e d t o b e r u n . F o r m o r e i n f o r m a t i o n , s e e C o n f i g u r i n g O A M S n a p s h o t T o o

Parameter	Description	Sample Value
		I w h e n t h e D B i s o n a D i f f e r e n t I n f r a s t r u c t u r e
LOG_DIR	The path where the log output will be stored. Ensure that the directory exists and can be written to.	/tmp

An example `restore.properties` will look as follows:

```

ORACLE_HOME=/u01/oracle/product/OAM
DOMAIN_NAME=oam_domain
JAVA_HOME=/u01/oracle/product/jdk8
CONNECT_STRING=targetdb.example.com:1521/oampdb.example.com
OAM_SCHEMA_USER=DEV_OAM
SCHEMA_PASSWORD=<schema_password>
OAM_SCHEMA_PASSWORD=<schema_password>
DB_SYS_USER=sys
DB_SYS_PASSWORD=<sys_password>
DATA_PUMP_DIR=DATA_PUMP_DIR
OPERATION=MIGRATE
    
```

```
OAM_SCHEMA_PREFIX=DEV
DB_HOME=/u01/oracle/product/database/12.2.1/db_1
LOG_DIR=/tmp
```

44.3.4 Running the applySnapshot.sh script

Running the `applySnapshot.sh` script to restore the OAM domain and database schemas.

1. Launch a terminal window on the target server and navigate to the OAM `$MW_HOME/oam/server/upgrade/script` directory.
2. Run `$ sh applySnapshot.sh <path>/restore.properties`.

The output for an OAM domain and database schemas restore will look similar to the following:

```
bash-4.2$ sh applySnapshot.sh backup.properties
Invoking createtableusers command : sh createtableusers.sh /u01/oracle/
product/database/12.2.1/db_1 targetdb.example.com:1521/
oampdb.example.com sys DEV ***** /tmp *****
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Thu Apr 2 05:57:51 2020
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```

```
SQL> SQL> SP2-0606: Cannot create SPOOL file "sys/createtablespace.log"
```

```
SQL> show SGA
```

```
SQL>
```

```
Total System Global Area 4546625536 bytes
```

```
Fixed Size 2933352 bytes
```

```
Variable Size 889195928 bytes
```

```
Database Buffers 3640655872 bytes
```

```
Redo Buffers 13840384 bytes
```

```
SQL>
```

```
SQL>
```

```
sga_gb
```

```
-----
4.234375
```

```
.....
```

```
.....
```

```
.....
```

```
SQL>
```

```
1 row updated.
```

```
SQL>
```

```
3 rows updated.
```

```
SQL>  
1 row updated.
```

```
SQL>  
Table truncated.
```

```
SQL> SQL> SQL> SQL> Disconnected from Oracle Database 12c Enterprise Edition  
Release 12.1.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing  
options  
bash-4.2$
```

 **Note:**

The `applySnapshot.sh` restores all the OAM schemas to the target database. If each schema password is different in the source database, then you must change the passwords for those schemas in the target database manually prior to starting the OAM domain on the target infrastructure.

To change the passwords connect to the target database as the SYS user and run:

```
alter user <SCHEMA_PREFIX>_STB identified by <STB_PASSWORD>;  
alter user <SCHEMA_PREFIX>_IAU_APPEND identified by <IAU_APPEND_PASSWORD>;  
alter user <SCHEMA_PREFIX>_IAU_VIEWER identified by <IAU_VIEWERPASSWORD>;  
alter user <SCHEMA_PREFIX>_MDS identified by <MDS_PASSWORD>;  
alter user <SCHEMA_PREFIX>_WLS identified by <WLS_PASSWORD>;  
alter user <SCHEMA_PREFIX>_WLS_RUNTIME identified by <WLS_RUNTIME_PASSWORD>;  
alter user IAU identified by <IAU_PASSWORD>;  
alter user <SCHEMA_PREFIX>_OPSS identified by <OPSS_PASSWORD>;  
alter user <SCHEMA_PREFIX>_OAM identified by <OAM_PASSWORD>;
```

44.3.5 Restoring Keystores and Custom Applications

Manually restore keystores and custom applications.

The OAM Snapshot tool restores the OAM domain and database schemas only. If you backed up any of the files on the source server referenced in [Backing up Keystores and Custom Applications](#) then these must be restored to the same location on the target server, prior to starting the OAM domain.

Using the OAM Snapshot Tool

The OAM Snapshot Tool can be used to upgrade the existing OAM version to a newer version, rebuilding the OAM deployment on multi-node cluster, and so on.

Topics

- [Upgrading OAM Using Snapshot Tool](#)
- [Rebuilding OAM Deployment in a Multi-Node Cluster](#)

45.1 Upgrading OAM Using Snapshot Tool

OAM Snapshot tool supports upgrade from OAM 12c PS3 to OAM 12c PS4.

The following steps describe how to do an out-of-place upgrade from OAM 12c PS3:

1. On a separate infrastructure, install OAM 12c PS4. Ensure that the Middleware home path is same as the OAM 12c PS3 installation. For details, see [Installing and Configuring the Oracle Access Management Software](#).
2. Patch the OAM 12c PS3 installation with the latest Snapshot Tool binaries

To create a backup of the OAM Domain configuration, you must patch the OAM 12c PS3 binaries with the latest `dbconfig-utility.jar` and the Snapshot Tool script files. Copy the following files from the infrastructure containing OAM 12c PS4 installation to similar location on the infrastructure containing OAM 12c PS3 installation:

- a. Copy all the `*.sh`, `*.pl`, and `*.properties` files from the `$MW_HOME/idm/oam/server/upgrade/script` directory on the OAM 12c PS4 installation to the corresponding directory path `$MW_HOME/idm/oam/server/upgrade/script` on the OAM 12cPS3 installation.
 - b. Copy the `dbconfig-utility.jar` file from `$MW_HOME/idm/oam/server/tools/dbconfigutility` on the OAM 12c PS4 installation to the corresponding directory path `$MW_HOME/idm/oam/server/tools/dbconfigutility` on the OAM 12c PS3 installation.
3. Run the `createSnapshot.sh` with `OPERATION=BACKUP` from the OAM 12C PS3 installation location. For details, see [Backing Up an OAM Deployment](#).
 4. Run the `applySnapshot.sh` with `OPERATION=RESTORE` from the OAM 12C PS4 installation location. For details, see [Rebuilding an OAM Deployment - Domain Only](#)

Note:

To perform the migration of the DB Schemas along with OAM Upgrade to a new infrastructure, run the `createSnapshot.sh` and `applySnapshot.sh` with `OPERATION=MIGRATE`. For more information, see [Migrating OAM Domain and Database Schemas](#)

5. Restart the servers from the OAM 12c PS4 domain.

45.2 Rebuilding OAM Deployment in a Multi-Node Cluster

This section provides instructions about rebuilding OAM deployment in a multi-node cluster.

The OAM Snapshot Tool supports the following operations in a multi-node cluster:

- Scale-Out: Clone a Managed Server on a new infrastructure.
- Backup and Restore: Rebuild Admin Server and Managed Server cluster on new infrastructure.
- Migration: Rebuild Admin and the Managed Server cluster on new infrastructure and migrate the database schemas to a new infrastructure.

In this section, the following example is used to describe the steps to rebuild the OAM deployments on the target infrastructure:

Source Infrastructure has:

- `Infra1` that runs `adminserver`
- `Infra2` that runs `oam_server1`
- `Infra3` that runs `oam_server2`

Target Infrastructure has:

- `Infra1A` for rebuilding `adminserver`
- `Infra2A` for rebuilding `oam_server1`
- `Infra3A` for rebuilding `oam_server2`

Rebuilding Admin Server on a New Infrastructure

To rebuild the Admin Server on a new infrastructure, follow the procedures provided in [Performing Backup and Restore and Migration of OAM System Using Snapshot Tool](#).

Rebuilding Managed Servers on New Infrastructure

Prerequisite: In case of backup and restore, and migration scenarios, you must perform the following step:

After rebuilding the Admin Server on a new Infrastructure (for example, `Infra1A`), you must again create a backup of the Admin Server on `Infra1A`. This step is not required for the scale-out operation.

For each of the Managed Server in the cluster (for example, `Infra2` and `Infra3`), perform the following steps on the target infrastructure (`Infra2A` and `Infra3A`):

1. Copy all the binaries from `MW_HOME` directory in the source infrastructure to the corresponding path on the target infrastructure.

 **Note:**

`MW_HOME` is the path where OAM binaries are installed.

2. From the new infrastructure that hosts the Admin Server (for example, Infra1A), copy the files `oam.properties` and `OamFileArtifacts.properties` from `$MW_HOME/idm/oam/server/upgrade/script` to the corresponding path on the target infrastructure (Infra2A and Infra3A) where the Managed Servers needs to be rebuilt.

 **Note:**

The `oam.properties` and `OamFileArtifacts.properties` are generated by the `moveToDB.sh` and `extract_from_db.sh` scripts during the backup and restore, or migrate operations performed on the Admin Server.

3. On the target infrastructure, run the following command to create the domain:

```
$JAVA_HOME/bin/java
-cp $MW_HOME/idm/oam/server/tools/dbconfigutility/dbconfig-utility.jar
-Doracle.jdbc.timezoneAsRegion=false
oracle.security.am.artifacts.main.AccessDBCommand
./oam.properties $ORACLE_HOME extractFromDB
```

 **Note:**

`MW_HOME` is the directory path where the OAM binaries are installed and `ORACLE_HOME` is the bath path where OAM domain has been created.

4. Make the startup scripts executable using the following command:

```
chmod +x <PATH_TO_DOMAIN>/bin/*.sh
```

5. Restart the Managed servers by providing the Admin Server T3 URL. For example, `./startManagedWeblogic.sh oam_server1 t3://adminhost:adminport on Infra2A` and `./startManagedWeblogic.sh oam_server2 t3://adminhost:adminport on Infra3A`.

Troubleshooting

Troubleshoot failures running the `createSnapshot.sh` and `applySnapshot.sh` scripts.

Failures running `createSnapshot.sh`

If the OAM domain and database schemas backup fails when using `OPERATION=MIGRATE` then:

1. Remove any `*.dmp` files from the directory referenced by the `DATA_PUMP_DIR` alias.
2. Verify the `backup.properties` file and correct any errors.
3. Re-run `applySnapshot.sh <path>/backup.properties`.

Failures running `applySnapshot.sh`

If the OAM domain only restore fails when using `OPERATION=RESTORE` then:

1. Delete the directory `$ORACLE_HOME/user_projects`.
2. Verify the `restore.properties` file and correct any errors.
3. Re-run `applySnapshot.sh <path>/restore.properties`.

If the OAM domain and database schemas restore fails when using `OPERATION=MIGRATE` then:

1. In SQLPlus connect to the target database as the `SYS` user and run:

```
drop user <SCHEMA_PREFIX>_STB cascade ;
drop user <SCHEMA_PREFIX>_IAU_APPEND cascade ;
drop user <SCHEMA_PREFIX>_IAU_VIEWER cascade ;
drop user <SCHEMA_PREFIX>_MDS cascade ;
drop user <SCHEMA_PREFIX>_WLS cascade ;
drop user <SCHEMA_PREFIX>_WLS_RUNTIME cascade ;
drop user <SCHEMA_PREFIX>_IAU cascade ;
drop user <SCHEMA_PREFIX>_OPSS cascade ;
drop user <SCHEMA_PREFIX>_OAM cascade ;
drop tablespace <SCHEMA_PREFIX>_OAM INCLUDING CONTENTS AND DATAFILES;
drop tablespace <SCHEMA_PREFIX>_IAU INCLUDING CONTENTS AND DATAFILES;
drop tablespace <SCHEMA_PREFIX>_IAU_APPEND INCLUDING CONTENTS AND
DATAFILES;
drop tablespace <SCHEMA_PREFIX>_MDS INCLUDING CONTENTS AND DATAFILES;
drop tablespace <SCHEMA_PREFIX>_OPSS INCLUDING CONTENTS AND DATAFILES;
drop tablespace <SCHEMA_PREFIX>_IAS_OPSS INCLUDING CONTENTS AND DATAFILES;
drop tablespace <SCHEMA_PREFIX>_WLS INCLUDING CONTENTS AND DATAFILES;
drop tablespace <SCHEMA_PREFIX>_WLS_RUNTIME INCLUDING CONTENTS AND
DATAFILES;
drop tablespace <SCHEMA_PREFIX>_IAU_VIEWER INCLUDING CONTENTS AND
DATAFILES;
drop tablespace <SCHEMA_PREFIX>_STB INCLUDING CONTENTS AND DATAFILES;
```

2. Delete the directory `$ORACLE_HOME/user_projects`.
3. Verify the `restore.properties` file and correct any errors.
4. Re-run `createSnapshot.sh <path>/restore.properties`.

The `createSnapshot.sh` and `applySnapshot.sh` Scripts Hang

During the export or import of the database schemas while using `Operation=Migrate`, the scripts `createSnapshot.sh` and `applySnapshot.sh` appear to hang.

The export and import of database schemas take time to complete based on the amount of data. The hardcoded two minute delay specified in the scripts is to avoid overloading the database. It does not affect the time to export or import database schemas.

Database Migrate Using the Snapshot Tools Fails to Create `schema_version_registry` Table/Entries in Target Environment

Before running the `createSnapshot.sh` script, perform the following:

Connect to the source database as `sys` or `system` user and run the following SQL command:

```
create table <SCHEMA_PREFIX>_OAM.SCHEMA_VERSION_REGISTRY as select *
from SYSTEM.SCHEMA_VERSION_REGISTRY$ where MRC_NAME = '<SCHEMA_PREFIX>'
```

where `<SCHEMA_PREFIX>` is the `SCHEMA_PREFIX` used for creating the RCU.

After running the `applySnapshot.sh` script, perform the following:

Connect to the target database as `sys` or `system` user and run the following SQL commands:

```
insert into SYSTEM.SCHEMA_VERSION_REGISTRY$ select * from
REL2_OAM.SCHEMA_VERSION_REGISTRY;
drop table REL1_OAM.SCHEMA_VERSION_REGISTRY;
```

Part XII

Integrating Access Manager with Other Products

Administrators integrate Access Manager with products from other vendors.

This section contains the following chapters:

- [Integrating RSA SecurID Authentication with Access Manager](#)
- [Configuring Access Manager for Windows Native Authentication](#)
- [Integrating Access Manager with Outlook Web Application](#)
- [Integrating Access Manager with SAP NetWeaver Enterprise Portal](#)

Integrating RSA SecurID Authentication with Access Manager

Oracle provides components that interface with RSA Security products to provide native RSA SecurID® authentication for Access Manager protected resources.

This chapter introduces SecurID authentication and the components, requirements, and processes needed to successfully integrate SecurID authentication with Access Manager 12c. The following topics are included:

- [Introduction to Access Manager and RSA SecurID Authentication](#)
- [Components Required for SecurID Authentication](#)
- [SecurID Authentication Modes](#)
- [Configuring Access Manager for RSA SecurID Authentication](#)
- [Running a Custom RSA Plug-in](#)

47.1 Introduction to Access Manager and RSA SecurID Authentication

Access Manager 12c integrates with RSA components to provide SecurID authentication.

RSA SecurID authentication is based on two factors: something the user knows and something the user has:

- **Something the User Knows:** This is a secret personal identification number (PIN), similar in concept to a personal bank code PIN. In this case, the PIN may be system generated or personally chosen and registered with the RSA Authentication Manager.
- **Something the User Has:** This is the current code generated by a hand held device known as a token. Oracle Access Manager supports all RSA SecurID token form factors, both hardware and software-based.

These tokens algorithmically, based on an internal clock or event, generate tokencodes with unpredictable values. Together, the user's PIN and the SecurID tokencode become the user's Passcode.

Access Manager uses and supports RSA two-factor SecurID authentication security features and enables integration with SecurID authentication by providing:

- The HTML forms required for SecurID authentication operations
- The RSA SecurID Plugin you can use with the User Identification Plugin to create and orchestrate authentication

47.2 RSA Features Supported by Access Manager

Access Manager integrates with RSA Authentication Manager and provides the integration features described in [Table 47-1](#).

Table 47-1 Access Manager Support for RSA Features

RSA Feature	Access Manager Support
Authentication method	Native SecurID authentication
New PIN Mode (user-generated PINs)	<p>Asks for new PIN with confirmation.</p> <p>The token may be in New PIN mode the first time the user logs in or the Authentication Manager Administrator can enable New PIN mode. New PIN mode requires the user to complete a sequence of forms to define, or have the system generate, a new PIN number.</p> <p>Oracle-Provided New PIN Forms and Functions:</p> <ul style="list-style-type: none"> • System Generated PIN (not supported) • User Defined (4-8 Alpha/numeric characters) • User Defined (5-7 Numeric) • Deny 4 and 8 Digit PIN • Deny Alphanumeric PIN • Deny Numeric PIN • PIN Reuse <p>See Also: "SecurID New PIN Authentication".</p>
Next Tokencode	<p>During authentication, the Authentication Manager may direct the user to provide the next tokencode that appears on their SecurID token to prove that they have the assigned token. This operation is known as Next Tokencode mode, which can be triggered by one of the following situations:</p> <p>See Also: "SecurID Next Tokencode Authentication".</p>
Passcode	<ul style="list-style-type: none"> • 16 Digit Passcode • 4 Digit Fixed Passcode
Load Balancing	RSA Authentication Manager Replicas.
Secondary server support	Yes
SecurID user specification	Designated users
SecurID protection of Administrators	Yes
Access Manager features and functions	All

Access Manager does not support the RSA features in [Table 47-2](#).

Table 47-2 RSA Features Not Supported

RSA Feature	Not supported by Access Manager
RSA Authentication Manager 7.1 SP2	Is not supported in an Active Directory Forest multi-domain environment

Table 47-2 (Cont.) RSA Features Not Supported

RSA Feature	Not supported by Access Manager
Multiple ACE Realms	The RSA Authentication API uses an automatic response time load balancing algorithm to determine where to send an authentication request. Such requests go to either a primary RSA Authentication Manager or a replica. The automatic algorithm can be overridden by creating a manual load balancing configuration file, <code>sdopts.rec</code> . However manually weighting an RSA Authentication Manager as a server of last resort does not preclude the Agent from communicating with it. As such, a true failover setup cannot be achieved with this method. For more information, see your RSA Authentication Manager documentation
System Generated PINs	Not supported by Access Manager.
Failover	Not supported for OAM SecurID Servers because only one OAM SecurID Server can perform SecurID authentication.

47.3 Components Required for SecurID Authentication

The following components are needed for the integration:

- [Supported Versions and Platforms](#)
- [Required RSA Components](#)
- [Installation and Configuration Requirements](#)

47.3.1 Supported Versions and Platforms

RSA Authentication Manager v8.3+ and the SecurID Authentication API are supported with OAM 12.2.1.4.0.

47.3.2 Required RSA Components

The following RSA components are required for integrating Access Manager and SecurID Authentication.

- [RSA Authentication Manager](#)
- [RSA SecurID Tokens](#)

47.3.2.1 RSA Authentication Manager

Residing somewhere in your network are records of users, agents, tokens, and user's PINs. Portions of these records might reside in the Authentication Manager or in LDAP directories.

During authentication, Authentication Manager compares these records to the information it receives when a user attempts to access the network. If the records and tokencode or passcode match, the user is granted access.

47.3.2.2 RSA SecurID Tokens

An RSA SecurID token is either a hardware device or software-based security token that generates and displays a random number that enables users to securely access protected resources.

The random number is called a tokencode. Before a user can authenticate with a token, the token must be recognized by Authentication Manager. RSA, or your vendor, ships a token seed file that you must import into the data store. Seeds listed in this file are assigned to tokens for generating the tokencode when an authentication request is received from an Authentication Manager agent.

During the SecurID authentication process, users must submit their username and passcode using an HTML form. The RSA Authentication Manager authenticates the identity of each user through a server that is registered with the Authentication Manager as a client (RSA Authentication Agent). One Access Server (known as the Oracle SecurID Access Server to distinguish it from other Access Servers) must be registered and set up as a client/Agent.

The RSA Authentication Manager compares the tokencode it has generated with the tokencode the user has entered. Tokencodes change at a specified interval, typically 60 seconds. Time synchronization ensures that the tokencode displayed on a user's token is the same code the Authentication Manager software has generated for that moment. Authentication is successful when the tokencodes match. Two-factor authentication provides stronger legal evidence of who performed the task. When properly configured, the Authentication Manager tracks all login requests and operations to reliably identify the user who is responsible for each logged action.

47.3.3 Installation and Configuration Requirements

SecurID requires affinity between the OAM Server and the RSA Authentication Manager for a user interaction. Therefore, the authentication dialog between the user and OAM Server must be sticky (this constraint is a security feature of SecurID authentication). In a cluster environment, if a load balancer is used to route requests to multiple managed server, ensure that stickiness is set between the load balancer and OAM Server. The SecurID Authentication API is bundled with Access Manager and installed on all OAM Servers. The SecurID Authentication API provides the connection functionality that eliminates the need for an Authentication Agent to be installed on the OAM Server. In other words, the API is the agent.

Every OAM Server must be registered as an RSA Authentication Agent host on the Authentication Manager along with the guidelines as follows:

- Only one designated OAM SecurID Server can complete SecurID authentication. However, every OAM Server must be registered as an RSA Authentication Agent Host on the Authentication Manager.
- Enable the OAM SecurID Server to be recognized as an Authentication Manager client.
- Port 5500 (UDP) should be available for the Authentication Manager to communicate with authentication agents (OAM SecurID Server). This service receives authentication requests from Oracle SecurID Server and sends replies. For more details refer to your RSA Authentication Manager documentation.
- Manage authentication requests from the client to the Authentication Manager.
- Enforce two-factor authentication and block unauthorized access.

- Provide automatic load balancing by detecting replica Authentication Manager response times and routing authentication requests accordingly.
- Ensure that the system time on the client is correct to prevent the server and client from being out of sync.
- Failover is not supported for Access Manager.
- The SecurID Authentication Manager must be installed on a supported platform.
- The system time must be correct to prevent the server and client from being out of sync.
- The SecurID tokens or key fobs must be provisioned with the Authentication Manager by providing it with the token seed records.
- Each user name must be mappable through an LDAP filter to a Distinguished Name in the directory.
- An Authentication Manager slave and/or replicated Authentication Manager can provide failover if the primary Authentication Manager is down.
- This integration requires a custom HTML login form and a properties file. Sample Oracle-provided custom html and custom html properties files can be found in:

```
$ORACLE_HOME/oam/server/tools/customLoginHtml
```

See Also:

- [About Custom Login Pages in the *Developing Applications with Oracle Access Management*](#)
- ["Configuring Access Manager for RSA SecurID Authentication"](#)

47.4 SecurID Authentication Modes

The following scenarios illustrate the three modes of operation:

- [Standard SecurID Authentication](#)
- [SecurID Next Tokencode Authentication](#)
- [SecurID New PIN Authentication](#)

47.4.1 Standard SecurID Authentication

Here is an overview of the process that occurs when a user attempts to access a resource protected by the SecurID authentication scheme.

For information on Credential Collectors, see [Understanding Credential Collection and Login](#) .

Process overview: When the user requests a resource

1. The WebGate intercepts the resource request and queries the Access Server to determine if and how the resource is protected, and if the user is authenticated.
2. The OAM SecurID Server queries the directory for the authentication scheme, and receives authentication information from the directory.
3. The WebGate redirects to the Credential Collector, which presents a form challenging the user for a two-part SecurID Passcode.
4. The user submits credentials to the Credential Collector
5. The Credential Collector hands off the credentials to the OAM SecurID Server

6. The SecurID Authentication API on the OAM SecurID Server performs the authentication dialog and sends an LDAP bind to the Authentication Manager.
7. The Authentication Manager database matches the SecurID passcode to the user ID and returns a success response to the Authentication Manager, which matches the user's PIN.
8. The Authentication Manager returns the response to its Agent, the OAM SecurID Server.
9. When the user's credentials are valid, SecurID authentication is successful. The OAM SecurID Server creates a session for the user and redirects the user to the Webgate, which then queries the OAM SecurID Server for resource authorization:
 - Under certain conditions a New Tokencode mode is initiated, as described in "[Standard SecurID Authentication](#)".
 - Under certain conditions a New Pin mode is initiated, as described in "[SecurID Next Tokencode Authentication](#)".
10. The OAM SecurID Server evaluates the authorization request, which allows or denies access based upon the authorization rule.
11. When access is granted, the OAM SecurID Server passes authorization to the WebGate, which presents the resource to the user.

47.4.2 SecurID Next Tokencode Authentication

When Next Tokencode mode is On, the user must supply the next tokencode on their SecurID token.

This mode can be triggered when:

- An incorrect Passcode was provided repeatedly during login. When a user attempts authentication with incorrect passcodes four consecutive times, the Authentication Manager turns on Next Tokencode mode, as noted in the Authentication Manager's Activity Report. The next time the user successfully authenticates with their correct Passcode, they are challenged for the next tokencode that appears on their SecurID token.
- The Authentication Manager requires confirmation of, or synchronization with the token. Even with a correct Passcode, the Authentication Manager Administrator might set the Next Tokencode mode On to force the user to confirm that they have the SecurID token or to synchronize the token with the Authentication Manager. When Next Tokencode mode is On, the Next Tokencode challenge form is presented to the user immediately following a successful login.

Process overview: When Next Tokencode is On

1. The Credential Collector presents a form to challenge the user for the next tokencode on the token following a successful login.
2. The user enters a username, waits 60 seconds, then enters the next tokencode on the SecurID token.
3. When the tokencode is correct, the Passcode the user originally entered is accepted and the user is authenticated.

47.4.3 SecurID New PIN Authentication

When the user is required to have a new PIN, the Credential Collector prompts the user with specific forms.

Process overview: When New PIN is required

1. The Credential Collector presents a form that allows the user to enter the PIN they want.
2. The user enters the new PIN and then re-enters the new PIN to complete the form.
3. The OAM SecurID Server forwards the information to the Authentication Manager.
4. The Authentication Manager registers the new PIN, which becomes part of the Pincode the user must supply during subsequent logins.
5. The Login Form appears again where the user enters the username and Passcode for a forced re-authentication.

47.5 Configuring Access Manager for RSA SecurID Authentication

Users with valid Oracle Access Management Administrator credentials can enable RSA SecurID authentication.

Prerequisites

See [Installation and Configuration Requirements](#) for installation and configuration that is outside the scope of this manual) and which must be completed before you begin SecurID integration with Access Manager.



See Also:

- [Developing Custom Pages in *Developing Applications with Oracle Access Management*](#)

To set up SecurID Authentication with Access Manager

1. In your `oam-config.xml`, set the OAM SecurID Sever `serverRequestCacheType` parameter to BASIC, as follows:
 - a. Stop all WebLogic servers (OAM Servers and AdminServer).
 - b. Change the `serverRequestCacheType` from `COOKIE` (default) to `BASIC`, as follows:


```
<Setting Name="serverRequestCacheType" Type="xsd:string">BASIC</Setting>
```

See [Updating OAM Configuration](#) for information on how to update OAM configuration.
 - c. Start all WebLogic Servers (OAM Servers and AdminServer).
2. Register a Web agent from the RSA Console that will be used by Access Manager, then copy the agent configuration file (`sdconf.rec`) as follows:

```
$DOMAIN_HOME/config/fmwconfig/servers/$SERVER_NAME/oam/sdconf.rec
```

3. Using the Oracle Access Management Console, create a custom authentication module for RSA, as follows:

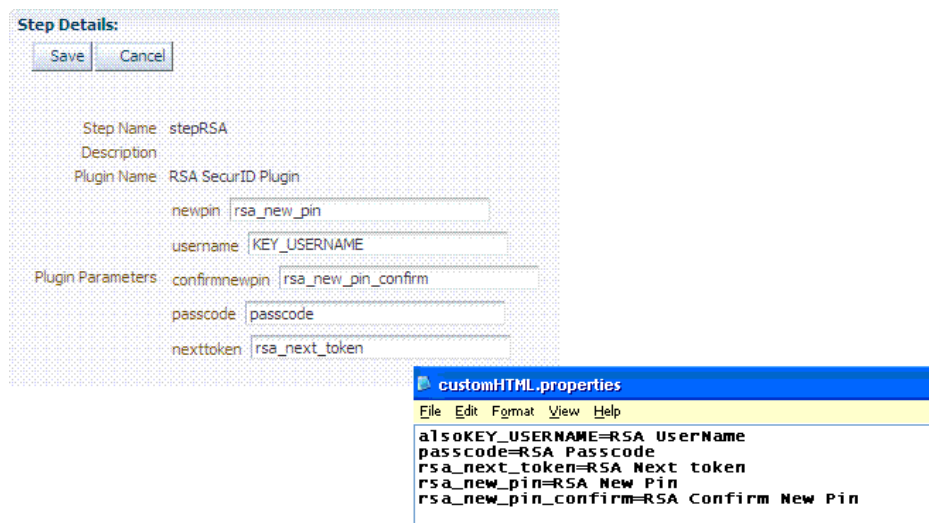
 **See Also:**

"Orchestrating Multi-Step Authentication with Plug-in Based Modules "

- a. Click **Application Security** at the top of the window.
- b. Select **Create Custom Authentication Module** from the **Create (+)** drop-down menu in the **Plug-ins** section.
- c. Select the **General** tab and enter the following:


```
Name: RSA_AUTH
```
- d. Select the **Steps** tab and enter a name for the Step, then choose the RSA SecurID Plugin


```
Step Name: stepRSA
Plugin Name: RSA SecurID Plugin
OK
```
- e. In the **stepRSA, Step Details** tab, enter and **Save** the Step Details shown in the next screen, which should also appear in your customhtml.properties file:



Step Details:

Save Cancel

Step Name: stepRSA
Description:
Plugin Name: RSA SecurID Plugin

newpin: rsa_new_pin
username: KEY_USERNAME

Plugin Parameters: confirmnewpin: rsa_new_pin_confirm
passcode: passcode
nexttoken: rsa_next_token

customHTML.properties

File Edit Format View Help

```
alsoKEY_USERNAME=RSA UserName
passcode=RSA Passcode
rsa_next_token=RSA Next token
rsa_new_pin=RSA New Pin
rsa_new_pin_confirm=RSA Confirm New Pin
```

- f. **Steps** tab: Add the User Identification Plugin: Enter a name for the Step, then choose the RSA SecurID Plugin:


```
Step Name: rsa_useridentification
Plugin Name: UserIdentificationPlugin
OK
```
- g. **rsa_useridentification, Step Details:** Enter and Save the following details for your environment:


```
KEY_LDAP_FILTER: (uid={KEY_USERNAME})
KEY_IDENTITY_STORE_REF: The registered Default Store.
```

KEY_SEARCH_BASE_URL: *dc=us,dc=example,dc=com*

- Orchestrate the steps as follows: stepRSA should be first (to authenticate the user with the RSA Server); designate your User Identification Plugin for the success step.

Initial Step: stepRSA

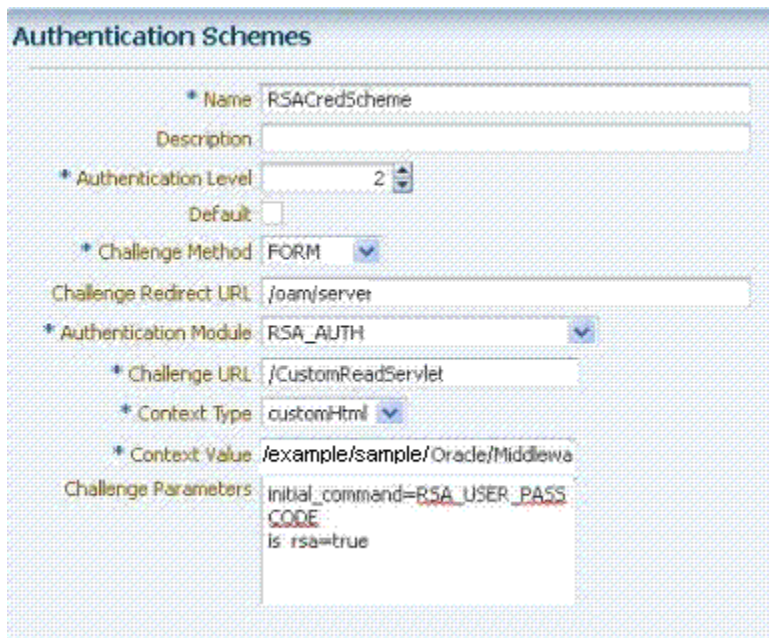
Name: StepRSA
 On Success: *rsa_useridentification*
 On Failure: failure
 On Error: failure
 Apply

Name: *rsa_useridentification*
 On Success: Success
 On Failure: failure
 On Error: failure
 Apply

 **Note:**

The On Failure and On Error fields must both be set to failure.

- Create a new authentication scheme (*RSACredScheme*, for example) that uses the custom authentication module that you just created for RSA with a custom HTML login form. Sample values are shown in the following screen:



Authentication Schemes

* Name: RSACredScheme
 Description:
 * Authentication Level: 2
 Default:
 * Challenge Method: FORM
 Challenge Redirect URL: /oam/server
 * Authentication Module: RSA_AUTH
 * Challenge URL: /CustomReadServlet
 * Context Type: customHtml
 * Context Value: /example/sample/Oracle/Middlewa
 Challenge Parameters: initial_command=RSA_USER_PASS
 CODE
 is_rsa=true

 **Note:**

The authentication scheme's Context Value specifies the path to your custom HTML login form. Your custom HTML properties file must share the same name as the form (with a `.properties` extension) in the same directory path. This example uses `customhtml.html` and `customhtml.properties`.

Challenge parameters specify the initial RSA command for authentication (`RSA_USER_PASSCODE`). The `is_rsa=true` parameter and value must be specified for RSA.

6. Use this scheme in the Application Domain protecting resources requiring SecurID authentication.
7. Ensure that your custom HTML file is present in:

```
$DOMAIN_HOME/config/fmwconfig/customhtml.html
```

The Custom HTML for RSA Login Form requires form action set to `/oam/server/auth_cred_submit`, as follows:

```
<form id="loginData" action="/oam/server/auth_cred_submit" method="post"
name="loginData">

<div id="oam_credentials" class="input-row">
<span class="ctrl"></span>
</div>
div class="button-row">
  <span class="ctrl">
<input id="login_button" type="submit" value="Login" class="formButton"
  onclick="this.disabled=true;document.body.style.cursor = 'wait';
  this.className='formButton-disabled';form.submit();return false;"/>
  </span>
</div>
<div id="oam_error_messages"></div>
</form>
```

8. Ensure that your `customHTML.properties` file is:
 - Named as your custom HTML file with a `.properties` extension
 - Stored in the same path as your custom HTML file
 - Confirmed; settings match the RSA SecurID plugin configuration parameters. For example:

```
username=Username
password=Password
passcode=Mother's maiden name
rsa_new_pin=RSA New Pin
rsa_new_pin_confirm=RSA Confirm New Pin
Pin=RSA Pin
rsa_sysgen_pin=RSA Create New Pin
rsa_sysgen_pin_confirm=RSA System Generated Pin
error1=Username not specified
```

9. Restart OAM Servers.

10. Test your configuration by accessing the appropriate protected resource and validating the various modes.
11. See "[RSA SecurID Issues and Logs](#)" for details if you experience problems.

47.6 Running a Custom RSA Plug-in

You can run a custom RSA plug-in located in `<ORACLE_HOME>/oam/custom_plugins/rsa/RSAPugin.jar`.

To run a custom RSA plug-in:

1. Download the RSA dependent libraries named `authapi.jar` and `cryptoj.jar`.
2. Add the `authapi.jar` and `cryptoj.jar` libraries to `<DOMAIN_HOME>/config/fmwconfig/oam/plugin-lib`.
3. Get the custom `RSAPugin.jar` file from its directory and import the plugin to add it to the list of custom plugins.
4. Once successfully imported, distribute and activate the plug-in.

Activation will fail the first time. When it does, restart the server and activate again. After activation, use the plugin to specify the necessary orchestration steps.

Configuring Access Manager for Windows Native Authentication

Access Manager enables browser users to automatically authenticate to their Web-based single sign-on applications using their desktop credentials. This is known as Windows Native Authentication (WNA).

This chapter contains the following sections to describe how to prepare your environment and perform this integration using Active Directory:

- [Introducing Access Manager with Windows Native Authentication](#)
- [About Preparing Your Active Directory and Kerberos Topology](#)
- [Confirming Access Manager Operations](#)
- [Enabling the Browser to Return Kerberos Tokens](#)
- [Integrating KerberosPlugin with Oracle Virtual Directory](#)
- [Integrating the KerberosPlugin with Search Failover](#)
- [Configuring Access Manager for Windows Native Authentication](#)
- [Validating WNA with Access Manager Protected Resources](#)
- [Configuring WNA For Use With DCC](#)
- [Troubleshooting WNA Configuration](#)

48.1 Introducing Access Manager with Windows Native Authentication

Access Manager supports Active Directory Multi-Domain and Multi-Forest topology integration with Windows Native Authentication (WNA).

The Active Directory directory service uses a data store that is also known as the **directory** for information about objects, such as users, groups, computers, domains, organizational units, and security policies.



See Also:

The System Requirements and Supported Platforms for Oracle Identity and Access Management at <https://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

For the integration, an application must be protected by an Access Manager authentication policy that uses the Kerberos authentication scheme (KerberosScheme) with WNA as the Challenge Method with the KerberosPlugin Authentication Module. In this case, credentials

must be stored in a Windows Active Directory instance that is registered as a user-identity store with Access Manager.

Each protected resource is defined in an Application Domain. The Authentication Policy includes the Authentication Scheme (KerberosScheme) that uses an Authentication Module (Kerberos) that is tied to the default User Identity Store. The store uses the value of "User Name Attribute" for authentication. This value is tied to the user in Active Directory and its values for `userprincipalname = username@domain` or `SamAccountName = username`, depending on the specific Access Manager release.

When Access Manager single sign-on is combined with WNA, a Kerberos session ticket is generated that contains the user's login credentials (among other things). This Kerberos session ticket is not visible to the user.

Access Manager interoperates with WNA, which uses Kerberos credentials obtained when the user logs in to a Windows Domain. This cross-platform authentication is achieved by emulating the negotiate behavior of native Windows-to-Windows authentication services that use the Kerberos protocol. For this cross-platform authentication to work, OAM Servers must parse SPNEGO tokens to extract the Kerberos tokens that are then used for authentication.

- **SPNEGO** is a Generic Security Services Application Programming Interface (GSSAPI) "pseudo mechanism" used to negotiate one of a number of possible real mechanisms. SPNEGO is largely employed in the Microsoft "HTTP Negotiate" authentication extension which uses it to allow initiators and acceptors to negotiate either Kerberos or NTLMSSP mechanisms. GSSAPI implementation is included with most major Kerberos distributions.

See <http://tools.ietf.org/html/rfc4559> for more information on SPNEGO.

- **Kerberos** is a network authentication protocol that provides strong authentication for client/server applications and services using a secret-key cryptography. A free implementation of Kerberos protocol is available from the Massachusetts Institute of Technology and is also commercially available.

See:

- [Understanding Access Manager WNA Login and Fall Back Authentication](#)
- [Supported Kerberos Authentication Modules](#)

48.1.1 Understanding Access Manager WNA Login and Fall Back Authentication

With WNA implemented, a user can open a Web application without another challenge for credentials because the Kerberos session ticket is passed through the browser to the OAM Server.

The OAM Server decrypts the received token (using keytab) and derives the authenticated user name from it. If authentication succeeds the user is granted access to the Web application automatically.



See Also:

Supported browsers in the System Requirements and Supported Platforms for Oracle Identity and Access Management at <https://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

The following sections describe an overview of the process of two WNA login scenarios.

- [Successful Access Manager WNA Authentication](#)
- [Access Manager WNA Fallback Authentication](#)

48.1.1.1 Successful Access Manager WNA Authentication

1. The Browser is configured for Integrated Windows Authentication (IWA).

This is a browser security configuration. If the browser being used is not configured to use IWA, no TGT is supplied when a resource protected by the Kerberos authentication module is requested. A browser basic authentication window is displayed where you can enter a valid username/password combination defined in the Default Identity Store for User login attribute.

2. A resource protected by Access Manager and WNA is called.

The protected resource should be configured as an intranet resource. This is done by adding the site in the "Local Intranet" zone of the Browser configuration.

3. A valid Kerberos ticket is present - Http headers... Authorization: Negotiate YIIJ/...
4. The user is not challenged for authentication. The requested resource is displayed, proving that WNA works.

In other words, when the browser is configured to use Integrated Windows Authentication, and a resource is protected by the Kerberos authentication module, then:

- If a Kerberos ticket is received by Access Manager (regardless of the domain), authentication is attempted:
 - Successful: Access is granted.
 - Failure: An incorrect user name or password error occurs if information from the Kerberos ticket is either not present or does not match the value of the User Name Attribute defined in the Default User Identity Store. Access is denied. The browser automatically submits the ticket, and the interaction with Access Manager is repeated until the user has been locked out. The browser cannot be made to pause before the start of each exchange.
- If the user is not logged on to a Windows Domain by way of Kerberos authentication, the browser sends OAM an NTLM token for authentication instead of a Kerberos token. Depending on how Access Manager is configured, it either uses WNA Fallback Authentication upon receiving an NTLM token or authentication fails.

 **Note:**

You need to configure Access Manager to provide fallback authentication when the browser sends an NTLM token. Without configuration, authentication fails. For configuration steps, see [Configuring WNA for NTLM Fallback](#).

NTLMSSP is a security support provider that is available on all versions of the Distributed Component Object Model (DCOM). It uses the NTLM protocol for authentication, which does not actually transmit the user's password to the server during authentication.

 **Note:**

If a Kerberos ticket cannot be identified by Access Manager (regardless of browser, Operating System, domain-login, and so on), the fallback mechanism is invoked.

48.1.1.2 Access Manager WNA Fallback Authentication

Fallback uses the authentication scheme "BasicScheme" with a challenge method of "Basic" and authentication module "LDAP".

This LDAP Authentication Module uses the LDAP plug-in. In this plug-in, the User Identity Store can be defined as any currently registered User Identity Store in which you define the attribute to be used for "User Name Attribute." The authentication module can be changed using the console.

1. The Browser is configured for Integrated Windows Authentication (IWA).

This is a browser security configuration. Access Manager handles two types of WNA fallback authentication.

- **Within Domain where IWA is enabled:** OAM supports WNA for the SPNEGO token. But sometimes due to configuration or other issues, OAM receives NTLM tokens from the client rather than SPNEGO. During the DEFAULT flow, OAM will try to authenticate using the NTLM token and fail because OAM doesn't have the capability to authenticate NTLM tokens. Thus, with introduction of "HandleNTLMResponse" configuration, OAM server will challenge client with Basic prompt for authentication. i.e. The fallback here is to prompt for basic mode of authentication if client is sending NTLM tokens to OAM Server. See [Configuring WNA for NTLM Fallback](#) for details.
- **Outside Domain where IWA is disabled:** Here no extra configuration is needed. By default the OOTB user will see a BASIC prompt during authentication.

2. A resource protected by Access Manager and WNA is called.

The protected resource should be configured as an intranet resource. This is done by adding the site in the "Local Intranet" zone of the Browser configuration.

3. No ticket is present (NTLM/Kerberos) - Http headers... Authorization: Basic

4. A basic authentication window pops up.
5. The user enters a valid username/password.
6. The requested resource is displayed (WNA Fallback works).

48.1.2 Supported Kerberos Authentication Modules

Use the Kerberos Authentication Module or KerberosPlugin Authentication Module when configuring Access Manager for Windows Native Authentication. The Kerberos Authentication Module identifies the key tab file and krb5 configuration file names and Principal.

The KerberosPlugin Authentication Module relies on bundled plug-ins (or those that are developed using the Access Manager Authentication Extensibility Java API). This module uses more than one plug-in that you can orchestrate to ensure that each one performs a specific authentication function. The KerberosPlugin Authentication Module is more robust and richer in functionality than the Kerberos Authentication Module. The KerberosPlugin Authentication Module (along with a plain WNA configuration) supports the following approaches:

- **Kerberos Plugin with Oracle Virtual Directory:** Using Access Manager with orchestrated authentication plug-ins integrated with Oracle Virtual Directory virtualize multiple Active Directory Global Catalogs.
- **Kerberos Plugin with Search Failover Across Multiple ADGCs:** Using Access Manager with orchestrated authentication plug-ins that exercise a failover pattern across multiple Active Directory Global Catalogs.

See Also:

- [About Preparing Your Active Directory and Kerberos Topology](#)
- [Confirming Access Manager Operations](#)
- [Integrating KerberosPlugin with Oracle Virtual Directory](#)

48.2 About Preparing Your Active Directory and Kerberos Topology

You need a fully-configured Microsoft Active Directory authentication service set up. Your Active Directory and Kerberos client will operate together.

The tasks in this section are required regardless of the approach you choose. However, none of this is Oracle specific.

Note:

The following sample scenario represents a typical Active Directory topology, and is not a requirement dictated by or for Access Manager. The naming used here is an example only. Your environment will be different.

As a sample scenario, consider two Active Directory forests operating within a company.

Forest	Domain Name
ORACLE	lm.example.com
SPRITE	lmsib.sprite.com

Consider that a child domain exists within the ORACLE forest: `child.lm.example.com`.

Trust is required as follows:

- Between forests: Two-way, non-transitive trust.
- Between the child domain and its parent: Two-way, transitive trust.

The suffixes and inheritance are:

- SPRITE users have UPN suffixes such as `sun.com` or `java.com`. The SPRITE forest contains `testuser.java.com`.
- ORACLE users have suffixes such as `myoracleco.com` and `oracleco.com`. The ORACLE forest contains `testuser.oracleco.com`.
- ORACLE child domain inherits the UPN suffixes of the parent domain.

**Note:**

Pre-Windows user names formed as `DOMAIN\USERNAME`, are not supported.

For integration with WNA, the `User Name Attribute` defined for the Default Identity Store can be any attribute whose value matches the Active Directory user's `samAccountName`.

You also need to know which encryption type your environment will use. In some cases a user might be created with "Use DES encryption types for this account" enabled. However, Active Directory is not using DES encryption.

**Note:**

The keytab file created in the following procedure uses RC4-HMAC encryption.

Access Manager supports what JGSS/JDK6 supports. The limitation on the TGT encryption that can be used would be determined by the piece that is the least common or lowest encryption supported: KDC, Keytab, Operating System, Kerberos client. Access Manager does not support any specific Kerberos encryption type. It is dependent on the Generic Security Services (GSS)/Kerberos jdk encryption types with which it is certified. Access Manager is not dependent on any encryption type and does not use TGT encryption. As part of SPNEGO token Access Manager only looks

into the Service Ticket which is encrypted with a key that the service (in this case Access Manager) has registered when executing the `ktpass/keytab` commands.



Note:

The keytab file created in the following procedure uses RC4-HMAC encryption.

Encryptions are used for communication among the different OS (Windows/Linux acting as Kerberos Server/Client). OAM Server just needs the SPNEGO token, from which it extracts the user credential. The encryption used in this three way negation process between the Windows Client (Browser), the Windows KDC, and the Generic Security Services (GSS) classes used by Access Manager, depend on the versions used (which must match).



See Also:

My Oracle Support for details about the Kerberos Encryption types Access Manager Supports [Doc 1212906.1] at: <https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1212906.1>

In the trusted domain (for example, the root domain `lm.example.com`), it is required that you follow the steps provided to:

- Create an account for the OAM Server.
- Extract the keytab file that was configured with the Active Directory Multi-Domain or Multi-Forest topology and trust relationships.
- Specify the Service Principal Name (SPN) using the fully-qualified hostname of the OAM Server (or the load balancer that represents the OAM Cluster), followed by the Realm name.

For this example the names in [Table 48-1](#) are used.

Table 48-1 Sample Naming

Name	Description
<code>kdc.lm.example.com</code>	KDC <i>hostname</i> KDC is a trusted network service that supplies session tickets and temporary session keys to users and computers within an Active Directory domain. The KDC runs on each domain controller as part of Active Directory Domain Services and is implemented as a domain service. The KDC uses Active Directory as its account database. In implementations of the Kerberos protocol, the KDC is a single process that provides two services: Authentication Service (AS) and Ticket Granting Service (TGS).
<code>kdc.lm.example.com</code>	AdminServer <i>hostname</i> This is the same as the KDC <i>hostname</i> .
<code>oam12c.example.com</code>	OAM Server <i>hostname</i>

Table 48-1 (Cont.) Sample Naming

Name	Description
LM.EXAMPLE.COM	Default Active Directory Realm
LMSIB.SPRITE.COM	Second Active Directory Realm The realm name identifies the location of the user account. A realm name can be either a prefix or a suffix. When an access client sends user credentials, a user name is often included. Within the user name are two elements: a user account name and user account location.
HTTP/ <i>fully_qualified_OAMServerhostname@REALM_NAME</i> (in CAPITAL letters)	Service Principal Names (SPNs) are needed for user accounts (the name by which a client uniquely identifies an instance of a service). Note: If you install multiple instances of a service on computers throughout a forest, each instance must have its own Service Principal Name.

48.2.1 Preparing Active Directory and Kerberos

You can prepare Active Directory and Kerberos.

Commands are for a Unix Operating System. Command syntax will vary depending on the specific Operating System in your environment.

1. Check the Oracle certification matrix to ensure you are installing a supported version of Active Directory for this integration:

<https://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

2. Install and configure Active Directory as follows:
 - Multi-Forest topology with requisite trust relationships configured and functional, including:
 - a. User accounts to map Kerberos services
 - b. Service Principal Names (SPNs) for these user accounts (the name by which a client uniquely identifies an instance of a service).
 - c. Key tab files
 - Active Directory Global Catalog (ADGC) enabled and functional within each forest
 - **Multi-Forest Deployment:** In this case, ensure there exists a naming attribute (available in global catalog) that uniquely identifies the users originating from various forests. Generally, `userprincipalname` is unique for the forest and `samAccountName` is unique for the domain
 - One domain that is directly or indirectly trusted by every other domain, regardless of forest affiliation.
3. Create a user for Access Manager use during WNA authentication and record this user name for generating the keytab file (no DES encryption).
4. Record the OAM Server *hostname*. For example:

`oam12c.example.com`

- Record the KDC *hostname* and the Active Directory domain/realm:

```
KDC = kdc.lm.example.com
Default AD Realm = lm.example.com
```

- Create the Service Principal Name (SPN) of the Active Directory user that the OAM Server client is using, and record the results (including encryption type).

The user name should be in the format `user_name@example.com` where `example.com` is the domain name of the Active Directory. For example:

```
ktpass -princ <protocol/oamserver_host> -pass <mypassword> -mapuser <user from
step 3> -out <path_to_filename>
```

Note:

Ensure that the case of the user name is consistent when entering it with the `ktpass`, `kinit` and `klist` commands. If you enter the user name in lower case when running one command, it must be entered in lower case when running the other commands.

For example, the case used in the commands to create the keytabs and the configuration in `/etc/krb5.conf` file need to match. When `ktpass` is run to create the keytab (as below), the host name of the KDC server is `lm.example.com`. Since this is all lower case, the configuration in the `/etc/krb5.conf` file must also be lower case. Case sensitivity is not the issue as long as the case matches.

```
ktpass -princ HTTP/oam12c.example.com@lm.example.com -mapuser oam -pass
examplepw -out c:\temp\oam.keytab
```

```
C:\Users\Administrator>ktpass -princ HTTP/oam12c.example.com@LM.EXAMPLE.COM
-mapuser oam -pass welcome1 -out c:\temp\oam.keytab
Targeting domain controller: kdc.lm.example.com
Using legacy password setting method
Successfully mapped HTTP/oam12c.example.com to oam.
WARNING: pType and account type do not match. This might cause problems.
Key created.
Output keytab to c:\temp\oam.keytab:
Keytab version: 0x502
keysize 80 HTTP/oam12c.example.com@lm.example.com ptype 0 (KRB5_NT_
UNKNOWN) vno 3 etype 0x17 (RC4-HMAC) keylength 16 (0xa3a685f89364d4a
5182b028f8e79ac38)
```


 **Note:**

If the user is not part of the Administrators group, follow this procedure to explicitly allow a remote desktop connection for the user.

- a. From the Oracle Access Management Console, navigate to the Remote tab through Control Panel -> Remote Settings -> System Properties.
- b. Select the "allow connections from computers running any version of Remote Desktop" option.
- c. Click Select Users.
- d. Add the user.
- e. Click Apply.

7. Copy the newly created keytab file to the proper location on the OAM Server and ensure permissions are correct so that the user who created Access Manager can access this file for running `ktpass` command.
8. Create a simple OAM Server Kerberos `krb5.conf` or `krb5.ini` configuration. For example:

```
[libdefaults]
default_realm = lm.example.com
ticket_lifetime = 600
clock_skew = 600

[realms]
lm.example.com = { --
kdc = kdc.lm.example.com
admin_server = kdc.lm.example.com
default_domain = lm.example.com
}
[domain_realm]
lm.example.com =LM.EXAMPLE.COM
.lm.example.com = LM.EXAMPLE.COM
```

 **Note:**

The OAM account is created in one domain that is trusted by all, `lm.example.com`. This is not required for `lmsib.sprite.com`.

9. Verify the `klist` and `kinit`s work using the keytab file and SPN of the Active Directory and Access Manager user created, then record the results.
 - a. `kdestroy`
 - b. `klist [-k] [-t <keytab_filename>]`. For example:

```
bash-3.2$ klist -k -t -K -e FILE:/refresh/home/oam.keytab
Keytab name: FILE:/refresh/home/oam.keytab
KVNO Timestamp Principal
-----
-----
3 12/31/69 19:00:00 HTTP/oam12c.example.com@lm.example.com (ArcFour
```

```
with HMAC/md5) (0xa3a685f89364d4a5182b028f8e79ac38)
bash-3.2$
```

c. `kdestroy`

d. `kinit [-k] [-t <keytab_filename>] [<principal>]`. For example:

```
klist -k -t -K -e FILE:/refresh/home/oam.keytab
```

```
bash-3.2$ kinit -V -k -t /refresh/home/oam.keytab
HTTP/oam12c.example.com@lm.example.com
Authenticated to Kerberos v5
```

e. `klist -e`

```
bash-3.2$ klist -e
Ticket cache: FILE:/tmp/krb5cc_8000
Default principal: HTTP/oam12c.example.com@lm.example.com
```

```
Valid starting Expires Service principal
02/25/12 18:46:55 02/25/12 18:56:55 krbtgt/LM.EXAMPLE.COM@LM.EXAMPLE.COM
Etype (skey, tkt): ArcFour with HMAC/md5, AES-256 CTS mode with 96-bit
SHA-1 HMAC
```

```
Kerberos 4 ticket cache: /tmp/tkt8000
klist: You have no tickets cached
bash-3.2$
```

10. Proceed as follows:

Successful: Continue with "[Confirming Access Manager Operations](#)".

Not Successful: Stop and resolve the issue which is not related to this integration. Any failure at this point indicates Access Manager WNA cannot work.

48.3 Confirming Access Manager Operations

You need a fully-functioning Access Manager deployment.

The tasks in this section are required regardless of the approach you choose. In this procedure you will install and register a WebGate, which configures an Application Domain to protect resources. Then you verify that the environment is working with an authentication scheme other than Kerberos.

 **See Also:**

Creating a High Availability Environment in the *High Availability Guide* for details about high availability environments with two or more Managed Servers configured to operate as a cluster

1. Log in to the Oracle Access Management Console using Administrator credentials.
2. Verify the Default Identity Store connection.
3. Register and install WebGate as an OAM Agent and accept automatic policy generation.
4. Add resources to the Application Domain and customize the authentication policy protecting resources to use any Authentication Scheme other than Kerberos.

5. Test the configuration to ensure that resource protection and access are working as expected.
6. Proceed to [Enabling the Browser to Return Kerberos Tokens](#)

48.4 Enabling the Browser to Return Kerberos Tokens

You can configure Internet Explorer, Mozilla Firefox, Edge or Chrome to return Kerberos tokens.

Perform the appropriate procedure on all Active Directory servers. Use either of the following procedures to configure the browsers.

- [Enabling Kerberos Tokens in Internet Explorer](#)
- [Enabling Kerberos Tokens in Mozilla Firefox](#)
- [Enabling Kerberos Tokens in Edge or Chrome](#)



Note:

With Internet Explorer browsers, Integrated Windows Authentication is enabled by default and you might not need any changes to the default configuration for WNA to work.

48.4.1 Enabling Kerberos Tokens in Internet Explorer

You can enable Kerberos token in Internet Explorer.

To enable Kerberos token:

1. On a Windows host in the Active Directory domain, sign in as a domain user.
2. Open the Internet Explorer browser.
3. From the Tools menu, click Internet Options, click Security, click Local Intranet, click Advanced.
4. On the Advanced tab, Security section, check the box beside Enable Integrated Windows Authentication, and click OK.
5. Add *Oracle Access Manager CC host or domain name* to Local Intranet zone (use the format `http://node.host:port` (the *port* is not required)). For example:
`http://oam12c.example.com`
6. Restart the Internet Explorer browser to enable the change.

48.4.2 Enabling Kerberos Tokens in Mozilla Firefox

You can enable Kerberos tokens in Mozilla Firefox.

To enable Kerberos tokens:

1. In the browser Address bar, enter `about:config`.

2. Add *Oracle Access Manager CC host or domain name* under `network.negotiate-auth.trusted-uris` as: `network.negotiate-auth.trusted-uris=http://oam12c.example.com`

Multiple URIs are separated with a comma.

48.4.3 Enabling Kerberos Tokens in Edge or Chrome

You can enable Kerberos tokens in Edge or Chrome browser.

To enable Kerberos tokens:

1. On a Windows host in the Active Directory domain, sign in as a domain user.
2. Open the Control Panel.
3. Select Network & Internet option, click Internet Options, click Security, click Local Intranet, click Advanced.
4. On the Advanced tab, Security section, check the box beside Enable Integrated Windows Authentication, and click OK.
5. Add *Oracle Access Manager CC host or domain name* to Local Intranet zone (use the format `http://node.host:port` (the port is not required)). For example:
`http://oam12c.example.com`
6. Restart Edge or Chrome browser to enable the change.

48.5 Integrating KerberosPlugin with Oracle Virtual Directory

Oracle Virtual Directory provides the ability to integrate LDAP-aware applications into diverse directory environments while minimizing or eliminating the need to change either the infrastructure or the applications.

This section provides the tasks you must perform to configure Access Manager KerberosPlugin authentication for WNA with Oracle Virtual Directory.

1. Perform tasks in this section:
 - [Preparing Oracle Virtual Directory for Integration](#)
 - [Registering Oracle Virtual Directory as the Default Store for WNA](#)
 - [Setting Up Authentication with Access Manager KerberosPlugin and OVD](#)
2. [Configuring Access Manager for Windows Native Authentication](#)
3. [Enabling the Browser to Return Kerberos Tokens](#)
4. [Validating WNA with Access Manager Protected Resources](#)

48.5.1 Preparing Oracle Virtual Directory for Integration

Oracle Virtual Directory communicates with other directories through adapters. Before you can start using Oracle Virtual Directory as an identity store, you must create adapters to each of the directories you want to use.

The procedure differs slightly, depending on the directory to which you are connecting. If you choose to use Oracle Internet Directory, Active Directory, Oracle Directory Server Enterprise Edition (ODSEE), or Oracle Unified Directory, the required adapters are created and

configured while installing and configuring the Oracle Identity Management Server. For more information on managing the adapters, see "Managing Identity Virtualization Library (libOVD) Adapters" in the Integration Guide for Oracle Identity Management Suite.

In the following procedure you create an account for the OAM Server in the trusted domain. Additionally, you create two Active Directory Adapters (one for each forest) using the fully-qualified domain names as namespaces. By default Active Directory uses `dc` to construct the root context distinguished name. If this is different in your deployment, adjust your adapter namespaces accordingly.

1. Perform tasks described in "[Confirming Access Manager Operations](#)".
2. Install Oracle Virtual Directory, as described in *Installing and Configuring Oracle Identity and Access Management*.
3. In **Oracle Virtual Directory Console**, create two Active Directory Adapters (one for each forest) using the fully-qualified domain names as namespaces as follows:
 - a. Adapter 1, EXAMPLE Adapter namespace (domain DNS `lm.example.com`):
`dc=lm,dc=example,dc=com`
 - b. Adapter 2, SPRITE Adapter namespace (domain DNS `lmsib.sprite.com`):
`dc=lmsib,dc=sprite,dc=com`
4. Shut down the OAM Cluster.
5. Restart the AdminServer and all OAM Servers.
6. Proceed with "[Registering Oracle Virtual Directory as the Default Store for WNA](#)".

48.5.2 Registering Oracle Virtual Directory as the Default Store for WNA

Users with valid Oracle Access Management Administrator credentials can register Oracle Virtual Directory as the user store for Access Manager interoperating with Windows Native Authentication.

For Windows Native Authentication, the user credentials must reside in Microsoft Active Directory. Access Directory can be managed by Oracle Virtual Directory instance. For single sign-on with Access Manager, each User Identity Store must be registered to operate with Access Manager.

Typically, `userprincipalname` reflects the Windows login name. For WNA with Access Manager, either leave the User Search Base and Group Search Base blank or provide the distinguished name path that is common to both the adapters configured while performing prerequisite tasks. Before you begin, be sure to complete the sections [About Preparing Your Active Directory and Kerberos Topology](#) and [Confirming Access Manager Operations](#).

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. Click **User Identity Stores**.
3. In the **OAM ID Stores** section, click **Create**.
4. Enter required values for your Oracle Virtual Directory instance. For example:

Name: *OVD*
 LDAP Url: *ldap://ovd_host.domain.com:389*
 Principal: *cn=Administrator,cn=users,dc=lm,dc=example,dc=com*
 Credential: *******
 User Search Base: *dc=com*
 User Name Attribute: *userprincipalname*
 Group Name: *cn*
 Group Search Base: *dc=com*
 LDAP Provider: Oracle Virtual Directory

5. **Default Store:** Click the **Default Store** button to make this the user Identity Store for Access Manager.
6. Click **Apply** to submit the registration, then dismiss the Confirmation window.
7. Restart the AdminServer and OAM Servers.
8. Proceed to "[Setting Up Authentication with Access Manager KerberosPlugin and OVD](#)".

48.5.3 Setting Up Authentication with Access Manager KerberosPlugin and OVD

When a native authentication module does not offer enough flexibility for your needs, you can create a custom authentication module using plug-ins designed to meet specific needs.

The `KerberosPlugin` is a credential mapping module that matches the credentials (encrypted username in the Kerberos ticket (SPNEGO token)) of the user who requests the resource. By default, `KerberosPlugin` maps the domain DNS name to the corresponding distinguished name using the `dc` component. However, if the mapping is different, you can specify the correct mapping as a semi-colon (;) separated list of name:value tokens. For example:

```
LM.EXAMPLE.COM:dc=lm,dc=example,dc=com;LMSIB.SPRITE.COM:dc=lmsib,dc=sprite,dc=com
```

Users with valid Oracle Access Management Administrator credentials can perform the following task to replace default `KerberosPlugin` steps with steps that enable integration for Windows Native Authentication using the Oracle Access Management Console.

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. Click **Authentication Modules** in the **Plug-ins** section.
3. Click **Search**, locate the **KerberosPlugin** plug-in and open it for editing.
4. On the `KerberosPlugin` page, click the **Steps** tab.

Steps Tab: Replace `stepKTA`, as described here, then click **Save**.

- a. Click **stepKTA** then click the **Delete (x)** button to remove this step.
- b. Click the **Add (+)** button and add the following step to the plug-in:

Element	Description
Name	stepKTA
Class	KerberosTokenAuthenticator

Step Details:

Edit this new **stepKTA** to change the Step Orchestration value from NULL (defined during the step deletion) to its default value of:

On Success: StepUIF Failure Failure

Also, confirm that this new **stepKTA** includes the parameter **KEY_DOMAIN_DNS2DN_MAP** (created earlier), enter the appropriate values for your deployment and click Save.

Element	Description
KEY_DOMAIN_DNS2DN_MAP	Active Directory Forests in your deployment. For example: LM.EXAMPLE.COM:dc=lm,dc=example,dc=com;LMSIB.SPRITE.COM:dc=lmsib,dc=sprite,dc=com Note: By default, a DN domain name a.b.c is mapped into dc=a,dc=b,dc=c. Only if the mapping is different, one has to specify the parameter. Otherwise it is best not to use it and let the default behavior take its course.
Service Principal	HTTP/oam12c.example.com@LM.EXAMPLE.COM
keytab.conf	keytab.conf location for stepKTA
krb5.conf	krb5.conf location for stepKTA

5. stepUIF Details: Configure as follows and click **Save**:

Element	Description
KEY_LDAP_FILTER	(samAccountName={KEY_USERNAME})
KEY_IDENTITY_STORE_REF	OVD
KEY_SEARCHBASE_URL	Leave this empty

6. stepUI and stepUA: Configure as follows and **Save**:

Element	Description
KEY_IDENTITY_STORE_REF	OVD

- Save the changes.
- Restart the OAM Cluster.
- Proceed with ["Configuring Access Manager for Windows Native Authentication"](#).

48.6 Integrating the KerberosPlugin with Search Failover

In cases where an Oracle Virtual Directory deployment is not viable, and it is acceptable to perform search failover based on some order or hierarchy when finding the user, you can configure Access Manager.

- Complete tasks in the following earlier sections:
 - ["About Preparing Your Active Directory and Kerberos Topology"](#)

- ["Confirming Access Manager Operations"](#) (except "Preparing Oracle Virtual Directory for This Integration", which is not needed for Search Failover)
 - ["Enabling the Browser to Return Kerberos Tokens "](#)
2. Perform tasks in this section:
 - ["Registering Microsoft Active Directory Instances with Access Manager"](#)
 - ["Setting Up the KerberosPlugin for ADGCs"](#)
 3. ["Configuring Access Manager for Windows Native Authentication"](#)
 4. ["Validating WNA with Access Manager Protected Resources "](#)

48.6.1 Registering Microsoft Active Directory Instances with Access Manager

Users with valid Oracle Access Management Administrator credentials can register each Active Directory Global Catalog (ADGC), with relevant search bases and naming attributes, as an individual User Identity Store for Oracle Access Management.

A fully-configured Microsoft Active Directory authentication service should be set up with User accounts for mapping Kerberos services, Service Principal Names (SPNs) for those accounts, and Key tab files. For more information, see *Oracle Fusion Middleware Securing Oracle WebLogic Server 11g Release 1 (10.3.3)*.

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. Click **User Identity Stores**.
3. In the **OAM ID Stores** section, click **Create**.
4. Enter required values for your first ADGC. For example:

```
Name: ADGC1-EXAMPLE
LDAP Url: ldap://ADGC1_host.domain.com:389
Principal: cn=Administrator,cn=users,dc=lm,dc=example,dc=com
Credential: *****
User Search Base: dc=lm,dc=example,dc=com
User Name Attribute: userprincipalname
Group Search Base: dc=lm,dc=example,dc=com
LDAP Provider: AD
```

5. **Default Store**: Click the **Default Store** button.
6. Click **Apply** to submit the changes and dismiss the confirmation window.
7. Repeat these steps to add the second ADGC (ADGC2-SPRITE) with appropriate search bases and naming attributes.

```
Name: ADGC2-SPRITE
LDAP Url: ldap://ADGC2_host.domain.com:389
Principal: cn=Administrator,cn=users,dc=lm,dc=example,dc=com
Credential: *****
User Search Base: dc=lmsib,dc=example,dc=com
User Name Attribute: userprincipalname
```


Group Search Base: `dc=lmsib,dc=example,dc=com`
LDAP Provider: AD

8. Restart the AdminServer and OAM Servers.
9. Proceed to "[Setting Up the KerberosPlugin for ADGCs](#)".

48.6.2 Setting Up the KerberosPlugin for ADGCs

When a native authentication module does not offer enough flexibility for your needs, you can create a custom authentication module using plug-ins designed to meet specific needs. The KerberosPlugin is a credential mapping module that matches the credentials (username and password) of the user who requests a resource to the encrypted "Kerberos ticket". By default, KerberosPlugin maps the domain DNS name to the corresponding distinguished name using the `dc` component.

However, if the mapping is different, you can specify the correct mapping as a semi-colon (;) separated list of name:value tokens. For example:

```
LM.EXAMPLE.COM:dc=lm,dc=example,dc=com;LMSIB.SPRITE.COM:dc=lmsib,dc=sprite,dc=com
```

Users with valid Oracle Access Management Administrator credentials can perform the following task to replace or update KerberosPlugin steps with steps that point to the ADGCs you have created. These will operate in tandem with their counterparts (if the initial step and ADGC fail, the secondary ADGC is used). Before you begin, be sure to complete the sections [About Preparing Your Active Directory and Kerberos Topology](#) and [Confirming Access Manager Operations](#).

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. Click **Authentication Modules** in the **Plug-ins** section.
3. Click **Search**, locate the **KerberosPlugin** plug-in and open it for editing.
4. On the KerberosPlugin page, click the **Steps** tab.

Steps Tab: Replace **stepKTA**, as described here, then click **Save**.

- a. Click **stepKTA** then click the **Delete (x)** button to remove this step.
- b. Click the **Add (+)** button and add the following step to the plug-in:

Element	Description
Name	stepKTA
Class	KerberosTokenAuthenticator

New stepKTA Details:

Confirm that this new stepKTA includes the parameter `KEY_DOMAIN_DNS2DN_MAP` (created earlier) and enter values for your deployment:

Element	Description
<code>KEY_DOMAIN_DNS2DN_MAP</code>	<code>LM.EXAMPLE.COM:dc=lm,dc=example,dc=com;LMSIB.SPRITE.COM:dc=lmsib,dc=sprite,dc=com</code>
Service Principal	<code>HTTP/oam12c.example.com@LM.EXAMPLE.COM</code>

Element	Description
keytab.conf	keytab.conf location for stepKTA. For example: /refresh/home/oam.keytab
krb5.conf	krb5.conf location for stepKTA. /etc/krb5.conf

5. **stepUIF: Step Details** (configure as follows and save):

Element	Description
KEY_IDENTITY_STORE_REF	ADGC1-ORACLE
KEY_SEARCHBASE_URL	{KEY_USERDOMAIN}
KEY_LDAP_FILTER	(samAccountName={KEY_USERNAME}) NOTE: For untrusted, multi-domain Active Directory environments, use the userPrincipalName user attribute.

6. **stepUI and stepUA: Step Details** (configure these steps and save):

Element	Description
KEY_IDENTITY_STORE_REF	ADGC1-ORACLE

7. Save the changes.

8. **Add stepUIF2:** This will operate in tandem and execute if stepUIF fails:

Element	Description
KEY_IDENTITY_STORE_REF	ADGC2-SPRITE
KEY_SEARCHBASE_URL	{KEY_USERDOMAIN}
KEY_LDAP_FILTER	(samAccountName= {KEY_USERNAME}) NOTE: For untrusted, multi-domain Active Directory environments, use the userPrincipalName user attribute.

9. **Add stepUI2:** This will operate in tandem and execute if stepUI fails:

Element	Description
KEY_IDENTITY_STORE_REF	ADGC2-SPRITE

10. **Add stepUA2:** This executes when stepUI2 succeeds:

Element	Description
KEY_IDENTITY_STORE_REF	ADGC1-EXAMPLE and ADGC2-SPRITE, respectively

11. **Add Step Details:** Common Configuration, Plugins, KerberosTokenAutheticator.

Enter values for your deployment:

Element	Description
keytab.conf	keytab.conf location for stepKTA. For example: /refresh/home/oam.keytab
krb5.conf	krb5.conf location for stepKTA. For example: /etc/krb5.conf

12. Restart the OAM Cluster.
13. Proceed with "[Configuring Access Manager for Windows Native Authentication](#)".

48.7 Configuring Access Manager for Windows Native Authentication

Whether you are using Oracle Virtual Directory or Active Directory with Global Catalogs, this section provides the following topics with steps you can follow:

- [Creating the Authentication Scheme for Windows Native Authentication](#)
- [Configuring Policies for Windows Native Authentication](#)
- [Configuring WNA for NTLM Fallback](#)
- [Configuring WNA Fallback to FORM-based Authentication Scheme](#)
- [Verifying the Access Manager Configuration File](#)

48.7.1 Creating the Authentication Scheme for Windows Native Authentication

Users with valid Oracle Access Management Administrator credentials can define an authentication scheme to use in policies protecting applications for Windows Native authentication.

Before you begin, be sure to complete one of the following sections: [Integrating KerberosPlugin with Oracle Virtual Directory](#) or [Integrating the KerberosPlugin with Search Failover](#).

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. Click **Authentication Schemes** in the **Access Manager** section.
3. Under **Search**, type *KerberosScheme* in the **Name** box and click Search.
4. Click **KerberosScheme** in the search results to open it.

Set (or confirm) the following attributes:

Challenge Method: WNA

Authentication Module: KerberosPlugin

5. Finish configuring **KerberosScheme** for your deployment.
6. Click **Apply** and close the confirmation window.
7. Proceed to "[Configuring Policies for Windows Native Authentication](#)".

48.7.2 Configuring Policies for Windows Native Authentication

You edit (or create) an Application Domain and policies to protect resources for Windows Native Authentication.

Before you begin, complete [Creating the Authentication Scheme for Windows Native Authentication](#).

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. Click **Application Domains** in the Access Manager section.
3. Open (or Create) the desired Application Domain, as described in "[Managing Application Domains Using the Console](#)".
4. **Resource Definitions:** Add Resource Definitions to the domain as described in "[Adding and Managing Policy Resource Definitions](#)".
5. **Authentication Policies:**
 - a. Open the Authentication Policies node, and open (or Create) the desired Authentication Policy with the following attributes:

Authentication Scheme: **KerbScheme** as the and ensure that it includes the updated **KerberosPlugin**.

Choose **KerbScheme** as the Authentication Scheme and ensure that it includes the updated **KerberosPlugin**.
 - b. Click **Apply**, close the Confirmation window.
 - c. **Resources for Authentication Policy:** Add Resources to the Authentication Policy as described in the *Administering Oracle Access Management*.
 - d. Complete the Authentication Policy with any desired Responses.
6. **Authorization Policies:** Complete the Authentication Policy with any desired Responses or Conditions as described in "[Defining Authorization Policies for Specific Resources](#)".
7. Proceed to "[Verifying the Access Manager Configuration File](#)".

48.7.3 Configuring WNA for NTLM Fallback

You can configure Access Manager to use WNA Fallback Authentication upon receiving an NTLM token.

For more information, see [Understanding Access Manager WNA Login and Fall Back Authentication](#).

To configure:

1. Stop the OAM managed server.
2. Export the `oam-config.xml` file from the database. See [Updating OAM Configuration](#) for details.
3. Edit `oam-config.xml` as follows:
 - a. Find the following line:

```
<Setting Name="CredentialCollector" Type="htf:map">
```

- b. After the line, add the following elements (if they are not already present):

```
-----
-
    <Setting Name="WNAOptions" Type="htf:map">
    <Setting Name="HandleNTLMResponse" Type="xsd:string">BASIC</
Setting>
    </Setting>
-----
-
```

If the following parameter already exists:

```
<Setting Name="HandleNTLMResponse" Type="xsd:string">DEFAULT</Setting>
```

change the `HandleNTLMResponse` value from `DEFAULT` to `BASIC`. For example:

```
<Setting Name="HandleNTLMResponse" Type="xsd:string">BASIC</Setting>
```

4. Import the modified `oam-config.xml` file into the database. See [Updating OAM Configuration](#) for details.
5. Restart the OAM server processes.

 **Note:**

See [Two BASIC Authentication Prompts Are Displayed](#) for troubleshooting information.

48.7.4 Configuring WNA Fallback to FORM-based Authentication Scheme

The `OAM_WNA_OPT_OUT` is a host-scoped persisting encrypted cookie set by the OAM Server. This cookie indicates the OAM server to challenge the user with FORM-based authentication when the browser presenting the cookie is not supporting WNA authentication. When set to `TRUE`, the `OAM_WNA_OPT_OUT` cookie makes the OAM server to change the authentication scheme from `DEFAULT` or `BASIC` to FORM-based before displaying the protected resource.

Upon receiving the NTLM tokens, OAM servers fall back to other authentication mechanisms. When `HandleNTLMResponse` is set to `BASIC`, the OAM server falls back to `BASIC` authentication scheme.

The WNA fallback to FORM-based authentication scheme relies on setting the pre-authentication rule. Create a pre-authentication rule that checks for `OAM_WNA_OPT_OUT` cookie which supports WNA FORM fallback mechanism. If the value of the `OAM_WNA_OPT_OUT` cookie is set `TRUE`, the authentication scheme is switched to FORM-based authentication.

1. Export the `oam-config.xml` file from the database. See [Updating OAM Configuration](#) for details.

2. Edit the file as follows:

```
<Setting Name="WNAOptions" Type="htf:map">
<Setting Name="HandleNTLMResponse" Type="xsd:string">FORM</Setting>
</Setting>
```

3. When NTLM and Kerberos authentications do not work with a browser (such as a non-domain attached browser), the OAM Server responds with an authorization error (403) and HTML content in the body of the response. By default, OAM displays an authorization error page with a **Login** button. The user needs to click the **Login** button in the customized page to invoke WNA fallback to FORM-based authentication. You can optionally configure *CustomOptOutPage* or *IsOptOutPersistent* parameters in the `oam-config.xml` and customize the error page.

- a. Configure the Custom Opt Out Page as follows to emit all the HTML contents from the `oam-config.xml` file. The JavaScript function `optOut()` is invoked when a button in the customized page is clicked. Then OAM emits the JavaScript function `optOut()`.

```
<Setting Name="CustomOptOutPage" Type="xsd:string">/home/custom.html</
Setting>
```

- b. The `OAM_WNA_OPT_OUT` cookie is set as persistent cookie, by default. Configure it as a session cookie as follows:

```
<Setting Name="IsOptOutPersistent" Type="xsd:boolean">>false</Setting>
```

4. Import the `oam-config.xml`. See [Updating OAM Configuration](#) for details.

5. Verify if the value of the `OAM_WNA_OPT_OUT` cookie is set to `TRUE` and the pre-authentication condition is set as follows:

```
str(request.requestMap['Cookie']).lower().find('oam_wna_opt_out=true') >=
0
```

6. If `enableWhiteListValidation` is set to `true` in the `oam-config.xml`, then the URLs for WNA protected resources must be defined in the Whitelist URL section. Whitelisting all the WNA-protected resources within the Whitelist URL section is an enormous task. You can use the following patterns and wild cards to reduce the number of needed entries:

```
http://oamwna1.vm.oracle.com/printenv_ecc_wna.pl
```

or

```
http://oamwna1.vm.oracle.com
```

or

```
http://*.vm.oracle.co
```

A sample `oam-config.xml` file after adding one of the above options as the protected resource to the Whitelist URL section looks like:

```
<Setting Name="WhiteListURLs" Type="htf:map">
  <Setting Name="Wild-Domain" Type="xsd:string">http://
```

```
*.vm.oracle.com</Setting>
</Setting>
```

 **Note:**

- In the absence of this step, WNA and WNA Fallback to FORM authentication fail with the following error: System error. Please re-try your action. If you continue to get this error, please contact the Administrator.
- For more details on this error, see Oracle Access Manager (OAM) Windows Native Authentication (WNA) Fails Error "System error, ..." (Doc ID 2815777.1) at <https://support.oracle.com>.

7. Restart the OAM server processes.

The OAM server falls back on FORM-based authentication scheme.

48.7.5 Verifying the Access Manager Configuration File

You can verify the Access Manager Configuration file, `oam-config.xml`.

Verify that the following are specified in the `oam-config.xml` file as in the following example:

- path to the `krb5.conf` file
- path to the `keytab` file
- a principal to connect with KDC

`oam-config.xml`

```
<Setting Name="KerberosModules" Type="htf:map">
  <Setting Name="6DBSE52C" Type="htf:map">
    <Setting Name="principal" Type="xsd:string">HTTP/
oam12c.example.com@LM.EXAMPLE.COM
  </Setting>
  <Setting Name="name" Type="xsd:string">XYZKerberosModule</Setting>
  <Setting Name="keytabfile" Type="xsd:string">/refresh/home/
oam.keytab
  </Setting>
  <Setting Name="krbconfigfile" Type="xsd:string">/etc/krb5.conf</Setting>
</Setting>
</Setting>
```

48.8 Validating WNA with Access Manager Protected Resources

Integrated Windows Authentication (IWA) is associated with Microsoft products that use SPNEGO, Kerberos, and NTLMSSP authentication protocols included with certain Windows operating systems.

The term Integrated Windows Authentication (IWA) is used for the automatic authentication process that happens between Microsoft Internet Information Services, browser, and Microsoft's Active Directory.

Note:

IWA is also known by other names such as HTTP Negotiate authentication, NT Authentication, NTLM Authentication, Domain authentication, Windows Integrated Authentication, Windows NT Challenge/Response authentication and Windows Authentication.

WNA authentication occurs internally. When integrated with Access Manager:

- The user is redirected to the Access Manager for authentication.
- The OAM Server requests authentication with a `www-negotiate` header when the resource is protected by Access Manager with a challenge method of WNA.
- The browser configured for Integrated Windows Authentication (IWA) sends the Kerberos SPNEGO token to the OAM Server for decryption.
- The OAM Server decrypts the received user SPNEGO token (using keytab) and redirects the user back to the Agent with the cookie and gets access to the resource.

Use this procedure to validate WNA with Access Manager protected resources.

1. Log in to a Windows system in the Active Directory domain as a domain user.
2. Sign in to the Windows OS client using the Windows domain credentials stored in a hosted Active Directory that is registered with Access Manager.
3. Open your browser window, and enter the URL for the OAM-protected application in your environment.
4. Confirm that you are logged in to the application with your Windows domain credentials with no additional login.

48.9 Configuring WNA For Use With DCC

The Kerberos authentication protocol provides a mechanism for mutual authentication between entities before a secure network connection is established.

This section provides information on how to configure Windows Native Authentication and Kerberos to use the DCC with Access Manager. It contains the following topics.

- [Initializing the Kerberos Protocol](#)
- [Configuring Access Manager](#)

 **Note:**

See [Understanding Credential Collection and Login](#) for details on DCC.

48.9.1 Initializing the Kerberos Protocol

You can initialize Access Manager for the Kerberos protocol.

To initialize:

1. Run the `ktpass` command on the Windows data store, substituting the appropriate values for service, realm, user and user password.

```
ktpass -princ <SPN>@<REALM> -pass <Password> -mapuser <UserName>
      -out <Keytab file name>
```

For example:

```
ktpass -princ HTTP/adc.example1.com@EXAMPLE.COM -pass Welcome1 -mapuser
      anil@example.com -out foobar2.keytab
```

This command creates an SPN and associates it with the local service account created in the previous step.

 **Note:**

Only RC4-HMAC encryption is supported; do not use DES encryption.

2. Copy the keytab output generated by the `ktpass` command and leave it at an appropriate location on the DCC host machine.
3. Modify the `/etc/krb5.conf` file on the DCC host machine accordingly.

For example:

```
[loggings]
default = FILE:/scratch/anikukum/krb/krb5libs.log
kdc = FILE:/scratch/anikukum/krb/krb5kdc.log
admin_server = FILE:/scratch/anikukum/krb/krbadmin.log
```

```
[libdefaults]
default_realm = EXAMPLE.COM
ticket_lifetime = 24h
forwardable = yes
dns_lookup_realm = false
dns_lookup_kdc = false
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
permitted_enctypes = rc4-hmac
clockskew = 3600
```

```
[realms]
EXAMPLE.COM = {
  kdc = adc.example1.com
  admin_server = adc.example1.com
  default_domain = EXAMPLE.COM
```

```

}

[domain_realm]
example.com = EXAMPLE.COM.example.com = EXAMPLE.COM

```

 **Note:**

For multiple domain Active Directory environments, add entries for each domain as documented below.

```

[realms]
EXAMPLE.COM = {
  kdc = adc.example1.com
  admin_server = adc.example1.com
  default_domain = EXAMPLE.COM
}

SPRITE.COM = {
  kdc = lmsib.sprite.com
  admin_server = lmsib.sprite.com
  default_domain = SPRITE.COM
}

[domain_realm]
example.com = EXAMPLE.COM
.example.com = EXAMPLE.COM
sprite.com = SPRITE.COM
.sprite.com = SPRITE.COM

```

4. Run the `kinit` command on the DCC host machine to obtain a Kerberos ticket.

```
kinit -k -t <keytab file> <SPN>@<Realm>
```

For example:

```
kinit -k -t foobar1.keytab HTTP/adc.example1.com@EXAMPLE.COM
```

5. Validate the Kerberos ticket on the DCC host machine using the `klist` command.

```
klist
```

48.9.2 Configuring Access Manager

You can configure Access Manager to use the Kerberos Authentication Module.

To configure:

1. Modify the Challenge Method of the Kerberos authentication scheme to WNA, if applicable.
 - a. In the Oracle Access Management Console, click **Application Security** at the top of the window.
 - b. In the Launch Pad tab, click **Authentication Schemes** in the **Access Manager** section.
 - c. Search for **KerberosScheme** and click **Edit**.
 - d. Change the Challenge Redirect URL to DCC WebGate URL.

For example, `http://<DCC-WebGate-Hostname>:<Port>/`

- e. Click **Apply** and close the page.
2. Configure the User Identity Store for LDAP Authentication Module to the configured Windows data store.
 - a. In the Oracle Access Management Console, click **Application Security** at the top of the window.
 - b. In the Launch Pad tab, click **Authentication Modules** in the **Access Manager** section.
 - c. Search for **LDAP** and click **Edit**.
 - d. Change the User Identity Store to, for example, Active Directory.
 - e. Click **Apply** and close the page.
3. Configure the Application Domain protecting the resource to use the Kerberos authentication scheme.

Before accessing the protected resource ensure that its URL is added to the local intranet Site of Security. Additionally, check the Enable Integrated Windows Authentication option under Security in the Advance tab.

48.10 Troubleshooting WNA Configuration

This section provides information about the following errors:

- [Kinit Fails](#)
- ["An Incorrect Username or Password was Specified" Is Displayed](#)
- [User Identity Store is Not Registered Correctly](#)
- [Two BASIC Authentication Prompts Are Displayed](#)

See Also:

Access Manager WNA Quick Start Guide on My Oracle Support, Knowledge Base note 1416903.1 at: <https://support.oracle.com/>

48.10.1 Kinit Fails

While retrieving initial credentials, the client may not be found in the Kerberos database.

This is the Kerberos version of "User not found" and might be related to one of the following:

- Misspelling or typo of the principal name
- The principal was not added to the Kerberos database, the principal doesn't exist.
- The user name does not exist in Active Directory or has not been registered as a Kerberos user.
- The SPN is not unique.

- On the Active Directory side one or more duplicate entries were found.

The solution would be to have the Active Directory Administrator search the LDAP tree for duplicate entries of the SPN, and remove them.

48.10.2 "An Incorrect Username or Password was Specified" Is Displayed

If unable to access a resource protected by Access Manager using the WNA authentication scheme, the error message is displayed.

When the error message, "An incorrect Username or Password was specified" is displayed, check the following.

- An incorrect username or password was specified.
- There is a mismatch in the encryption types being used.
- The key version number (kvno) of the SPN mentioned in the keytab does not match the kvno of the mapped user in the identity store.

48.10.3 User Identity Store is Not Registered Correctly

By default, the OAM identity store is Embedded LDAP. If you are using a different identity store (for example, Active Directory or Oracle Unified Directory) be sure to register the identity store.

[Managing Data Sources](#) has complete details on identity stores and how to register them.

- To set the identity store being used as the Default Store, see [About using the System Store for User Identities](#).
- To register the User Identity Store being used, see [Registering a New User Identity Store](#) with details in [User Identity Store Settings](#).

48.10.4 Two BASIC Authentication Prompts Are Displayed

If OAM is configured for WNA and the client browser is not configured for IWA, two BASIC authentication prompts might be displayed when accessing a WNA protected resource.

One prompt comes from the WebLogic Server and the second from OAM. To avoid this, the WebLogic Server must be configured to ignore HTTP Basic authentication requests.

1. Stop all WebLogic managed server and the admin server.
2. Create a copy of the config.xml file.
\$WLS_DOMAIN/config/config.xml
3. Add the following parameter at the end of the "<security-configuration>" section in the config.xml file.

```
<enforce-valid-basic-auth-credentials>>false</enforce-valid-basic-auth-credentials>
```

Be sure to add this parameter BEFORE the <cross-domain-security-enabled>>false</cross-domain-security-enabled> parameter.

4. Restart the WebLogic environment.

Integrating Microsoft SharePoint Server with Access Manager

This chapter explains how to integrate Access Manager with a 12c WebGate and Microsoft SharePoint Server. It covers the following topics:

- [What is Supported in This Release?](#)
- [Introduction to Integrating With the SharePoint Server](#)
- [Integration Requirements](#)
- [Preparing for Integration With SharePoint Server](#)
- [Integrating With Microsoft SharePoint Server](#)
- [Setting Up Microsoft Windows Impersonation](#)
- [Completing the SharePoint Server Integration](#)
- [Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider](#)
- [Configuring Single Sign-On for Office Documents](#)
- [Configuring Single Sign-off for Microsoft SharePoint Server](#)
- [Setting Up Access Manager and Windows Native Authentication](#)
- [Synchronizing User Profiles Between Directories](#)
- [Testing Your Integration](#)
- [Troubleshooting](#)

 **Note:**

Access Manager with a 12c WebGate supports Microsoft SharePoint Server 2010, 2013, and 2019.

Unless explicitly stated, all details in this chapter apply equally to Access Manager integration with Microsoft SharePoint Server using the OAM impersonation plug-in, and Microsoft SharePoint Server configured with the LDAP Membership Provider.

49.1 What is Supported in This Release?

Support for integration between Access Manager and SharePoint enables the following functionality:

- When a user accesses SharePoint before SSO login with Access Manager, the user is prompted for Access Manager SSO login credentials.

- When a user with a valid Access Manager login session wants to access SharePoint documents, he must be established with SharePoint (logged in and authenticated with SharePoint). Once the Access Manager session is established, it is also respected by SharePoint for integration with Access Manager and SharePoint using LDAP Membership Provider, OAM WNA, and impersonation. Based on authentication status, SharePoint either allows or denies access to documents stored in SharePoint.
- When a user opens an Office document from SharePoint using a browser, the SSO session should persist into the MS Office program so that access to the document through the MS Office program is maintained. See "[Configuring Single Sign-On for Office Documents](#)".

49.2 Introduction to Integrating With the SharePoint Server

SharePoint Server is a Microsoft-proprietary secure and scalable enterprise portal server that builds on Windows Server Microsoft Internet Information Services (IIS) and Windows SharePoint Services (WSS).

SharePoint Server is typically associated with Web content and document management systems. SharePoint Server works with Microsoft IIS web server to produce sites intended for collaboration, file sharing, web databases, social networking and web publishing. In addition to WSS functionality, SharePoint Server incorporates additional features such as News and Topics as well as personal and public views for My Site, and so on.

Microsoft SharePoint Server enhances control over content, business processes, and information sharing. Microsoft SharePoint Server provides centralized access and control over documents, files, Web content, and e-mail, and enables users to submit files to portals for collaborative work.

SharePoint server farms can host web sites, portals, intranets, extranets, Internet sites, web content management systems, search engine, wikis, blogs, social networking, business intelligence, workflow as well as providing a framework for web application development.

When integrated with Microsoft SharePoint Server, Access Manager handles user authentication through an ISAPI filter and an ISAPI Module. This enables single sign-on between Access Manager and SharePoint Server.

SharePoint Server supports the following authentication methods:

- Form Based Authentication
- Impersonation Based Authentication
- Windows Authentication: Used only for the configuration where the information about the users is stored in Active Directory server

The integrations in this chapter provide single sign-on to Microsoft SharePoint Server resources and all other Access Manager protected resources. For more information, see:

- [About Windows Impersonation](#)
- [Form Based Authentication With This Integration](#)
- [Authentication With Windows Impersonation and SharePoint Server Integration](#)
- [Access Manager Support for Windows Native Authentication](#)

49.2.1 About Windows Impersonation

Unless explicitly stated, the integrations described in this chapter rely on Windows impersonation. Windows impersonation enables a trusted user in the Windows server domain to assume the identity of any user requesting a target resource in Microsoft SharePoint Server.

This trusted impersonator maintains the identity context of the user while accessing the resource on behalf of the user.

Impersonation is transparent to the user. Access appears to take place as if the SharePoint resource were a resource within the Access System domain.



Note:

Windows impersonation is not used when integrating Microsoft SharePoint Server configured with the LDAP Membership Provider.

49.2.2 Form Based Authentication With This Integration

You can integrate Access Manager with SharePoint Server using any of the three authentication methods. Given common use of LDAP servers (Sun Directory Server and Active Directory for instance), your integration can include any LDAP server. Form-based authentication in SharePoint Server is claims-aware. When a user enters credentials on the Forms login page of SharePoint Relying Party (RP), these are passed to the SharePoint Security Token Service (STS). SharePoint STS authenticates the users against its membership provider and generates the SAML token, which is passed to SharePoint RP. SharePoint RP validates the SAML token and generates the FedAuth cookie. The user is then allowed to access the SharePoint RP site.

With form-based authentication, the WebGate is configured as an ISAPI filter. The form login page of SharePoint RP is customized such that the user is not challenged to enter the credentials by the SharePoint RP. Also, the membership provider is customized such that it just validates the OAMAuthnCookie set by the WebGate to authenticate the user.

The following overview outlines the authentication flow for this integration using form-based authentication.

Process overview: Request processing with form-based authentication

1. The user requests access to an SharePoint Server RP site.
2. The WebGate protecting the site intercepts the request, determines if the resource is protected, and challenges the user.
3. The user enters their OAM credentials. Next the OAM WebGate server verifies the credentials from LDAP and authenticates the user.

The WebGate generates the OAM native SSO cookie (OAMAuthnCookie), which enables single sign-on and sets the User ID header variable (to the user name) in the HTTP request and redirects the user to the SharePoint RP site.

4. The SharePoint RP custom login page is invoked, which sets the user name to the user ID passed in the header variable, and sets the password to the OAMAuthnCookie value. The login page also automatically submits these credentials to the SharePoint RP site.

5. The SharePoint RP site passes the credentials to SharePoint STS, which invokes the custom membership provider to validate the user credentials.
6. The custom membership provider gets the OAMAuthnCookie value (passed as a password) and sends it as part of the HTTP request to a resource protected by the WebGate to validate the OAMAuthnCookie.
7. If the OAMAuthnCookie is valid, SharePoint STS generates the SAML token and passes it to SharePoint RP.
8. SharePoint RP validates the SAML token and generates the FedAuth cookie. The user is then allowed to access the SharePoint RP site.

49.2.3 Authentication With Windows Impersonation and SharePoint Server Integration

Windows impersonation enables a trusted user in the Windows server domain to assume the identity of any user requesting a target resource in SharePoint Portal Server. The trusted impersonator maintains the identity context of the user while accessing the resource on behalf of the user. Impersonation is transparent to the user. Access appears to take place as if the SharePoint resource were a resource within the OAM Server domain.

The next overview identifies the authentication processing flow with SharePoint Server and Windows impersonation enabled.

Process overview: Integration Authentication with Windows Impersonation

1. The user requests access to a SharePoint Portal Server resource.
2. The WebGate ISAPI filter protecting SharePoint Portal Server intercepts the request, determines whether the target resource is protected, and if it is, challenges the user for authentication credentials.
3. If the user supplies credentials and the OAM Server validates them, the WebGate sets an OAMAuthnCookie in the user's browser, which enables single sign-on. The WebGate also sets an HTTP header variable named "impersonate," whose value is set to one of the following:
 - the authenticated user's LDAP `uid`
 - `samaccountname`, if the user account exists in Active Directory
4. The Access Manager HTTP module `IISImpersonationModule.dll` checks for the Authorization Success Action header variable named `impersonate`.
5. When the header variable exists, the Oracle ISAPI module obtains a Kerberos ticket for the user.

This Service for User to Self (S4U2Self) impersonation token enables the designated trusted user to assume the identity of the requesting user and obtain access to the target resource through IIS and the SharePoint Portal Server.

49.2.4 Access Manager Support for Windows Native Authentication

Access Manager provides support for Windows Native Authentication (WNA).

Your environment may include:

- Windows 2008/R2 or 2012/R2 server

- Internet Information services (IIS) 7.x or 8.x
- Active Directory

If the user's directory server has, for example, an NT Logon ID, or if the user name is the same everywhere, then a user is able to authenticate into any directory server. The most common authentication mechanism on Windows Server 2008 is Kerberos.

The use of WNA by Access Manager is seamless. The user does not notice any difference between a typical authentication and WNA when they log on to their desktop, open an Internet Explorer (IE) browser, request a protected web resource, and complete single sign-on.

Process overview: Using WNA for authentication

1. The user logs in to the desktop computer, and local authentication is completed using the Windows Domain Administrator authentication scheme.
2. The user opens an Internet Explorer (IE) browser and requests an Access System-protected Web resource.
3. The browser notes the local authentication and sends a Kerberos token to the IIS Web server.

 **Note:**

Ensure that Internet Explorer's security settings for the Internet and (or) intranet security zones are adjusted properly to allow automatic logon.

4. The WebGate installed on the IIS Web server sends the Kerberos token to the OAM sever. The OAM Server negotiates the Kerberos token with the KDC (Key distribution center).
5. Access Manager sends authentication success information to the WebGate.
6. The WebGate creates an OAMAuthnCookie and sends it back to the browser.
7. Access Manager authorization and other processes proceed as usual.

The maximum session time-out period configured for the WebGate is applicable to the generated OAMAuthnCookie.

49.3 Integration Requirements

Unless explicitly stated, this section introduces components required for integrations described in this chapter. It includes the following topics:

- [Requirements Confirmation](#)
- [Required Access Manager Components](#)
- [Required Microsoft Components](#)

49.3.1 Requirements Confirmation

References to specific versions and platforms are for demonstration purposes. For the latest Access Manager certification information, see the certification matrix on Oracle Technology Network at:

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

49.3.2 Required Access Manager Components

Access Manager provides access and security functions, including Web-based single sign-on, policy management, reporting, and auditing.

When integrated with Microsoft SharePoint Server, Access Manager handles user authentication through an ISAPI filter and an ISAPI Module, which enables single sign-on between the two products. The components in [Table 49-1](#) are required to integrate with Microsoft SharePoint Server (or Microsoft SharePoint Server configured with LDAP Membership Provider.)

Table 49-1 Component Requirements

Component	Description
12c WebGate	<p>The ISAPI version 12c WebGate must reside on the same computer as the SharePoint Server.</p> <p>Within the context of this integration, this WebGate is an ISAPI filter that intercepts HTTP requests for Web resources and forwards them to the OAM Server to authenticate the user who made the request. If authentication is successful, the WebGate creates an OAMAuthnCookie and sends it to the user's browser, thus facilitating single sign-on. The WebGate also sets impersonate as a HeaderVar action for this user session.</p> <p>For LDAP Membership Provider Scenario: See "Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider".</p>
IISImpersonationModule.dll	<p>This IIS-native module is installed with the WebGate. The IISImpersonationModule.dll module determines whether the Authorization Success Action HeaderVar has been set to impersonate and, if it has, the DLL file creates a Kerberos S4U2Self ticket that enables the special trusted user in the SharePoint Server Active Directory to impersonate the user who originally made the request.</p> <p>After a WebGate installation, you must configure IISImpersonationModule.dll manually to enable impersonation and this integration.</p> <p>For LDAP Membership Provider Scenario: Do not configure IISImpersonationModule.dll.</p>
Directory Server	<p>Access Manager can be connected to any supported directory server including, but not limited to, LDAP and Active Directory. Access Manager can even connect to the same instance of Active Directory used by SharePoint Server.</p> <p>In any case, the directory is not required on the same machine as SharePoint Server and the protecting WebGate.</p>

Table 49-1 (Cont.) Component Requirements

Component	Description
OAM Server	<p>The integration also requires installation of the OAM Server with which the WebGate protecting your SharePoint Server installation is configured to inter-operate.</p> <p>Except for the WebGate protecting SharePoint Server, your components do not need to reside on the machine hosting SharePoint Server.</p> <p>See Also: "Preparing for Integration With SharePoint Server".</p>

49.3.3 Required Microsoft Components

Minimum requirements dictate a 64-bit, four cores processor.

However, references to specific versions and platforms are for demonstration purposes. For the latest Access Manager certification information, see the following Microsoft library location for Microsoft SharePoint Server:

<https://technet.microsoft.com/en-us/library/cc262485.aspx>

The SharePoint multi-purpose platform allows for managing and provisioning of intranet portals, extranets, and Web sites; document management and file management; collaboration spaces; social networking tools; enterprise search and intelligence tooling; process and information integration; and third-party developed solutions.

Note:

Minimum requirements dictate a 64-bit, four cores processor. However, references to specific versions and platforms are for demonstration purposes. For the latest Access Manager certification information, see Oracle Technology Network at:

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

Table 49-2 describes the other components required for this integration.

See Also:

The following library location for Microsoft SharePoint Server and access to applicable software:

<http://technet.microsoft.com/en-us/library/cc262485.aspx>

Table 49-2 Microsoft Requirements for this Integration

Component	Description
Custom Login Page for SharePoint site	When the user tries to access a SharePoint site configured to use Form Based Authentication, the user is redirected to a login page where the user enters his or her credentials (user name and password). The custom login page passes the credentials to the SharePoint site.
SharePoint site	<p>You create the SharePoint site using the SharePoint Central Administration application. The site is configured to use Form Based Authentication as the authentication method by following the steps mentioned in http://technet.microsoft.com/en-us/library/ee806890.aspx.</p> <p>The SharePoint site passes the user credentials to the SharePoint STS that generates SAML token upon successful OAMAuthnCookie validation by the custom membership provider. The SharePoint site also generates FedAuth cookie upon receiving the SAML token from SharePoint STS. The SharePoint site passes the FedAuth cookie to the user so that he/she can access the SharePoint site.</p>
SharePoint Security Token Service (STS)	The SharePoint site passes the user credentials (user name and password) to SharePoint STS, which invokes the custom membership provider and passes the credentials to it. Once the custom membership provider validates the OAMAuthnCookie passed to it, the SharePoint STS generates the SAML token for the user that is passed to the SharePoint Relying Party (RP).
Custom Membership Provider for SharePoint STS	<p>The SharePoint STS invokes the membership provider (configured with Form Based Authentication). STS passes the user credentials and the URL for the IIS resource (configured in <code>web.config</code> on the SharePoint site) to the custom membership provider for cookie validation.</p> <p>The membership provider is customized such that it returns success if the OAMAuthnCookie value passed to it is valid.</p> <p>The custom membership provider library (<code>OAMCustomMembershipProvider.dll</code>) is packaged and installed with the 12c WebGate for IIS Web server. You must deploy the library in the global assembly cache of the SharePoint Server host.</p> <p>The <code>CustomMembershipProvider</code> class is derived from <code>LdapMembershipProvider</code> class present in the <code>Microsoft.Office.Server.Security</code> namespace.</p>
IIS resource for Cookie validation	<p>Configure the URL for the IIS resource in the SharePoint site's <code>web.config</code> file.</p> <p>For the HTTP validation method, the WebGate intercepts the request sent by the custom membership provider, extracts the OAMAuthnCookie from the request, and validates it. If the cookie is valid, then the request is redirected to the IIS resource, which returns the response with a 200 (OK) status code to the custom membership provider. Otherwise, a 403 (Forbidden) error code is returned to the custom membership provider.</p>

49.4 Preparing for Integration With SharePoint Server

The IIS 12c WebGate must be installed on the same computer as the SharePoint Server. Other components in this integration can reside on the same host as the WebGate or any other computer in your deployment (Solaris, Linux, or Windows platforms).

Tasks in the following procedure are required for all integration scenarios described in this chapter.

After installing and testing Microsoft components, perform steps here to install Access Manager for your integration. This task applies to both integration scenarios in this chapter. To avoid repetition, information here is not repeated elsewhere.

A different host can be set up for Active Directory or some other directory service. If both Access Manager and SharePoint Server are set up for different instances of Active Directory, both instances must belong to the same Active Directory domain.

Prerequisites

Install and test Microsoft components described in "[Required Microsoft Components](#)".

To prepare for integration with SharePoint Server

1. Install Oracle Identity Management and Access Manager as described in the *Installing and Configuring Oracle Internet Directory*.
2. Register a 12c WebGate for IIS Web server with Access Manager:
 - a. Log in to the Oracle Access Management Console. For example: `http://host:port/oamconsole`.
 - b. Click **Application Security** at the top of the window.
 - c. In the **Launch Pad** tab, click **SSO Agent Registration** in the **Quick Start Wizards** section.
 - d. Select **WebGate** as the agent type and click **Next**.
 - e. Set the agent version to **12c** and enter required details (those with an *):
 - Name
 - SharePoint user name and password
 - Security mode (Agent host must match OAM Server)
 - Auto Create Policies (Checked)

Note:

Do not specify a Base URL.

- f. **Protected Resource List:** In this table, enter individual resource URLs to be protected by this OAM Agent.
- g. **Public Resource List:** In this table, enter individual resource URLs to be public (not protected).
- h. Click **Apply** to submit the registration, check the Confirmation window for the location of generated artifacts, then close the window.

3. Proceed as follows:
 - **Install a fresh WebGate:** Continue with steps 6, 7, and 8.
 - **Existing WebGate on SharePoint Host:** Skip to "Integrating With Microsoft SharePoint Server".

 **Note:**

Only 64-bit ISAPI WebGates are supported as described in "Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider".

4. For Oracle Fusion Middleware 12c, the WebGate software is installed as part of the Oracle HTTP Server 12c software installation. Locate and download the 64-bit IIS WebGate installer as follows:
 - a. Go to Oracle Fusion Middleware 12c Software Downloads at:
<https://www.oracle.com/security/identity-management/technologies/downloads/>
 - b. Click **Accept License Agreement**, at the top of the page.
 - c. From the **Access Manager Webgates** row, click the download link for the desired platform and follow on-screen instructions.
 - d. Store the WebGate installer in the same directory as any 12c Access System Language Packs you want to install.
5. Launch the WebGate installer for your platform, installation mode, and Web server. Follow these steps:
 - a. Follow on-screen prompts.
 - b. Provide Administrator credentials for the Web server.
 - c. **Language Pack**—Choose a Default Locale and any other Locales to install, then click Next.
 - d. WebGate installation begins (IISImpersonationModule.dll will be installed in *WebGate_install_dir\webgate\iis\lib*).
6. Before updating the Web server configuration, copy WebGate artifacts from the Admin Server to the computer hosting the WebGate.
 - a. On the computer hosting the Oracle Access Management Console (AdminServer), locate and copy ObAccessClient.xml (and any certificate artifacts):

```
$DOMAIN_HOME/output/$Agent_Name/  
ObAccessClient.xml  
password.xml (if needed)  
aaa_key.pem (your private key generated by openssl)  
aaa_cert.pem (signed certificates in PEM format)
```
 - b. On the OAM Agent host, add the artifacts to the WebGate path. For example:

```
WebGate_instance_dir/webgate/config/ObAccessClient.xml
```

`WebGate_instance_dir/webgate/config/`

- c. Restart the WebGate Web server.
 - d. (Optional.) Restart the OAM Server that is hosting this Agent. This step is recommended but not required.
7. Proceed as needed to complete this integration within your environment:
- [Integrating With Microsoft SharePoint Server](#)
 - [Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider](#)

49.5 Integrating With Microsoft SharePoint Server

You can integrate with Microsoft SharePoint Server by creating a new Web application or site application.

The following overview outlines the tasks that you must perform for this integration and the topics where you will find the steps and details.

The custom membership provider library (`OAMCustomMembershipProvider.dll`) is packaged and installed with the 10g WebGate for IIS Web Server. You must deploy the library in the global assembly cache of the computer hosting SharePoint Server as outlined next.

Task overview: Integrating with Microsoft SharePoint Server includes

1. Performing prerequisite tasks:
 - [Installing "Required Microsoft Components"](#).
 - ["Preparing for Integration With SharePoint Server"](#)
2. Creating a new Web application (or site application) in SharePoint Server is described in following topics:
 - ["Creating a New Web Application in Microsoft SharePoint Server"](#)
 - [Creating a New Site Collection for Microsoft SharePoint Server](#)
3. ["Setting Up Microsoft Windows Impersonation "](#) (not used with LDAP Membership Provider).
4. ["Completing the SharePoint Server Integration"](#).
5. ["Configuring Single Sign-off for Microsoft SharePoint Server"](#).
6. ["Synchronizing User Profiles Between Directories"](#).
7. ["Testing Your Integration"](#).

49.5.1 Creating a New Web Application in Microsoft SharePoint Server

You can create a New Web Application in Microsoft SharePoint Server with or without LDAP Membership Provider.

You perform this task when integrating with Microsoft SharePoint Server, with or without LDAP Membership Provider.

Prerequisites

Installing Microsoft components. See "[Required Microsoft Components](#)".

To create a new Web application in Microsoft SharePoint Server

1. On the host where SharePoint Server is installed, open the Central Administration home page: Start, All Programs, SharePoint Products, SharePoint, Central Administration.
2. From the Central Administration home page, click Application Management.
3. From the Application Management page, Web Applications section, click Manage Web Applications.
4. In the top-left corner, click the New button to create a new web application.
5. Configure the items in [Table 49-3](#) on the Create New Web Application page:

Table 49-3 Create Web Application Options for Microsoft SharePoint Server

Section	What You Configure in This Section
Authentication	In this section you select either Claim Based Authentication or Classic Mode Authentication, as appropriate.
IIS Web Site	<p>In this section you configure the following settings for your new Web application, as follows:</p> <ul style="list-style-type: none"> • To choose an existing Web site, click Use an Existing Web Site... • To create a new site, click Create. • In the Port field, enter the port number you want to use to access the Web application. For a new Web site, this field contains a default port number. For an exiting site, this field contains the currently configured port number. • In the optional Host Header field, enter the URL for accessing the Web application. • In the Path field, enter the path to the directory that contains the site on the server. For a new Web site, this field contains a default path. For an exiting site, this field contains the current path.
Security Configuration	<p>In this section you configure authentication and encryption for your Web application, as follows:</p> <ul style="list-style-type: none"> • In the Authentication Provider section, select Negotiate(Kerberos) or NTLM, as appropriate. • In the Allow Anonymous section, choose Yes or No. A value of Yes allows anonymous access to the Web site by using a computer-specific anonymous access account. The account name is <i>IUSR_computername</i>. • In the Secure Sockets Layer (SSL) section, choose Yes or No. If you choose to enable SSL for the Web site, you must configure SSL by requesting and installing a certificate.
Public URL	Enter the URL for the domain name for all sites that users will access in this Web application. This URL domain will be used in all links shown on pages in the Web application. By default, the box is populated with the current server name and port. The Zone field is automatically set to Default for a new Web application and cannot be changed from this page.

Table 49-3 (Cont.) Create Web Application Options for Microsoft SharePoint Server

Section	What You Configure in This Section
Application Pool	<p>In the Application Pool section, choose whether to use an existing application pool or create a new application pool for this Web application, as follows:</p> <ul style="list-style-type: none"> To use an existing application pool, select Use Existing Application Pool, then select the application pool you wish to use from the drop-down menu. To create a new application pool, select Create a New Application Pool, and in the Application Pool Name field, type the name of the new application pool, or keep the default name. <p>In the section Select a Security Account for This Application Pool, select Predefined to use an existing application pool security account, then select the security account from the drop-down menu. To use a security account that is not currently being used for an existing application pool, select Configurable, enter the user name of the account you want to use in the User Name field, and enter the password for the account in the Password field.</p>
Database Name and Authentication	<p>In this section, choose the database server, database name, and authentication method for your new Web application.</p> <p>In the Database Name field, enter the name of the database or use the default entry. In the Database Authentication field, choose whether to use Windows authentication (recommended) or SQL authentication, as follows:</p> <ul style="list-style-type: none"> If you want to use Windows authentication, leave this option selected. If you want to use SQL authentication, select SQL authentication. In the Account field, type the name of the account that you want the Web application to use to authenticate to the SQL Server database, then type the password in the Password field.
Failover Server	You can optionally choose to specify a fail-over database server to configure a Fail-over Server.
Service Application Connections	You can use the default value or choose custom value and optionally select the services you want your web application to connect to.

- Click OK to create the new Web application, or click Cancel to cancel the process and return to the Application Management page.
- Proceed with "[Creating a New Site Collection for Microsoft SharePoint Server](#)".

49.5.2 Creating a New Site Collection for Microsoft SharePoint Server

You can create a new site collection for Microsoft SharePoint Server with or without LDAP Membership Provider.

To create a new site collection for Microsoft SharePoint Server

- From the Application Management page, Site Collection section, click Create Site Collections.

2. On the Create Site Collection page, in the Web Application section, either select a Web application to host the site collection (from the Web Application drop-down list), or create a new Web application to host the site collection, as follows:

NOT_SUPPORTED

Section	What You Configure in This Section
Quota Template	You can decide to use predefined quota template to limit resources used for this site collection or use "No quota" as appropriate.
Title and Description	Enter a title and description for the site collection
Web Site Address	Select a URL type, and specify a URL for the site collection.
Template	Select a template from the tabbed template control.
Primary Site Collection Administrator	Enter the user account name for the user you want to be the primary Administrator for the site collection. You can also browse for the user account by clicking the book icon to the right of the text box. You can verify the user account by clicking the check names icon to the right of the text box.
Secondary Site Collection Administrator (optional)	Enter the user account for the user that you want to be the secondary Administrator for the site collection. You can also browse for the user account by clicking the book icon to the right of the text box. You can verify the user account by clicking the Check Names icon to the right of the text box.

3. Refer to the following topics as you finish this integration:
 - ["Setting Up Microsoft Windows Impersonation "](#)
 - ["Completing the SharePoint Server Integration"](#)
 - ["Configuring Single Sign-off for Microsoft SharePoint Server"](#)
 - ["Synchronizing User Profiles Between Directories"](#)
 - ["Testing Your Integration"](#)

 **See Also:**

"Task overview: Integrating with Microsoft SharePoint Server Configured with LDAP Membership Provider"

49.6 Setting Up Microsoft Windows Impersonation

If you want to use a directory server other than Active Directory, use LDAP Membership provider. The OAMCustomMembership provider leverages the functionality of LDAP Membership provider.

This section describes how to set up impersonation, whether for SharePoint Server integration or for use by some other application.



Note:

Skip this section if you are integrating Microsoft SharePoint Server configured with LDAP Membership Provider. Windows impersonation is not used with the LDAP Membership Provider.

Task overview: Setting up impersonation

1. Create a trusted user account for only impersonation in the Active Directory connected to SharePoint Server, as described in "[Creating Trusted User Accounts](#)".
2. Give the trusted user the special right to act as part of the operating system, as described in "[Assigning Rights to the Trusted User](#)".
3. Bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user, as described in "[Binding the Trusted User to Your WebGate](#)".
4. Add a header variable named *IMPERSONATE* to the Authorization Success Action in the Application Domain for impersonation, as described in "[Adding an Impersonation Response to an Authorization Policy](#)".
5. Configure IIS by adding the *IISImpersonationModule.dll* to your IIS configuration, as described in "[Adding an Impersonation DLL to IIS](#)".
6. Test impersonation, as described in "[Testing Impersonation](#)".

49.6.1 Creating Trusted User Accounts

You can create trusted user accounts. The special user should not be used for anything other than impersonation.

The example in the following procedure uses *Impersonator* as the New Object - User. Your environment will be different.

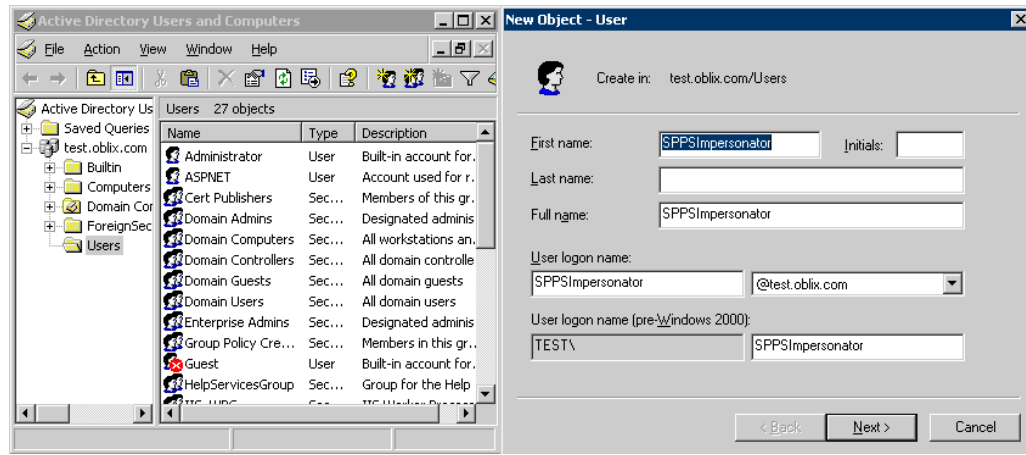
To create a trusted user account:

1. Perform the following steps on the computer hosting your SharePoint Server installation:
 - Windows: Select Start, Programs, Administrative tools, Active Directory Users and Computers.
2. In the Active Directory Users and Computers window, right-click Users on the tree in the left pane, then select New, User.
3. In the First name field of the pane entitled New Object - User, enter an easy-to-remember name such as *Impersonator*.
4. Copy this same string to the User logon name field, then click Next.
5. In succeeding panels, you will be asked to choose a password and then retype it to confirm.

 **Note:**

Oracle recommends that you chose a very complex password, because your trusted user is being given very powerful permissions. Also, be sure to check the box marked Password Never Expires. Since the impersonation module should be the only entity that ever sees the trusted user account, it would be very difficult for an outside agency to discover that the password has expired.

Figure 49-1 Setting up a Trusted User Account for Windows Impersonation

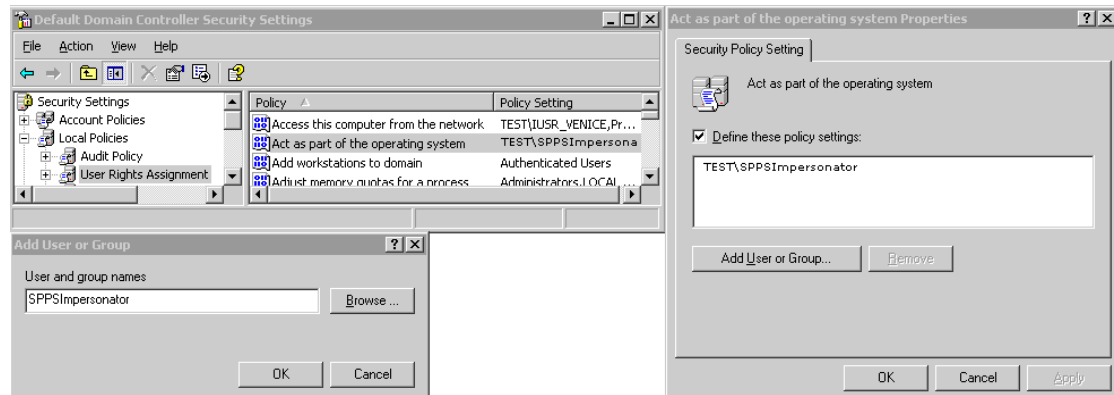


49.6.2 Assigning Rights to the Trusted User

You need to give the trusted user the right to act as part of the operating system.

To give appropriate rights to the trusted user:

1. Perform steps for your environment:
 - Windows: Select Start, Programs, Administrative tools, Local Security Policy.
2. On the tree in the left pane, click the plus icon (+) next to Local Policies.
3. Click **User Rights Assignment** on the tree in the left pane.
4. Double-click Act as part of the operating system in the right pane.
5. Click **Add User or Group**.
6. In the Add User or Group panel, type the User logon name of the trusted user (SPPSImpersonator in our example) in the User and group names text entry box, then click OK to register the change.

Figure 49-2 Configuring Rights for the Trusted User in Windows Impersonation

49.6.3 Binding the Trusted User to Your WebGate

You need to bind the trusted user to the 12c WebGate that communicates with Access Manager by supplying the authentication credentials for the trusted user.

The following procedure presumes that you have not yet registered a 12c WebGate with Access Manager. Values in the following procedure are provided as an example only. Your environment will be different.

To bind your trusted user to your WebGate

1. Go to the Oracle Access Management Console.

For example:

```
http://hostname:port/oamconsole
```

where *hostname* is the fully-qualified DNS name of the computer hosting the Oracle Access Management Console; *port* is the listening *port* configured for the OAM Server; *oamconsole* leads to the Oracle Access Management Console.

2. Click **Application Security** at the top of the window.
3. In the **Launch Pad** tab, click **SSO Agent Registration** in the **Quick Start Wizards** section.
4. Select **WebGate** as the agent type and click **Next**.
5. Set the version to **12c** and enter required details (those with an *) to register this WebGate.
6. **Protected Resource List:** In this table, enter individual resource URLs to be protected by this OAM Agent, as shown in Table 14–9.
7. **Public Resource List:** In this table, enter individual resource URLs to be public (not protected), as shown in Table 14–9.
8. **Auto Create Policies:** Check to create fresh policies (or clear and use the same host identifier as another WebGate to share policies (Table 14–9)).
9. Click **Apply** to submit the registration.
10. Check the Confirmation window for the location of generated artifacts, then close the window.

11. In the navigation tree, open the Agent page.
12. **SharePoint Requirements:** Enter trusted user credentials in the **Sharepoint Impersonator** fields and click **Apply**.
13. Copy the artifacts as follows (or install the WebGate and then copy these artifacts):
 - a. On the Oracle Access Management 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 computer hosting the agent, copy the artifacts. For example
12c WebGate/AccessClient: `$WebGate_instance_dir/webgate/config/ObAccessClient.xml`
`ObAccessClient.xml`
 - c. Proceed to "[Adding an Impersonation Response to an Authorization Policy](#)".

49.6.4 Adding an Impersonation Response to an Authorization Policy

An Application Domain and basic policies to protect your SharePoint resources was created when you registered the WebGate with Access Manager. You must add an Authorization Success Action (Response) with a return type of Header, set the name to `IMPERSONATE`, with the Response value of `$user.userid: "samaccountname"` for a single-domain Active Directory installation or "userPrincipalName" for a multi-domain Active Directory forest.

To add an impersonation response to your Authorization Policy:

1. Click **Application Security** at the top of the Console window.
2. In the **Launch Pad** tab, click **Application Domains** in the **Access Manager** section.
3. Search for the desired domain and open it for editing.
4. Click the **Authorization Policies** tab and open the desired policy for editing.
 "Desired domain" refers to the Application Domain created specifically for impersonation (*Impersonation* for example). "Desired policy" is your default policy created during agent registration. By default, no policy Responses exist until you create them.
5. On the Policy page, click the **Responses** tab, click the **Add (+)** button, and:
 - From the Type list, choose **Header**.
 - In the Name field, enter a unique name for this response: `IMPERSONATE`
 - In the Value field, enter a value for this Response. For example: `$user.userid`.
6. Click **Add** to save the Response, which is used for the second WebGate request (for authorization).

49.6.5 Adding an Impersonation DLL to IIS

You can add an impersonation DLL to IIS.

You can configure IIS Web server for the integration by registering and configuring the IISImpersonationModule.dll across all sites including central administration and web services.

Alternatively, if you have multiple Web sites, where some are integrated with Access Manager while others are not, you might want to enable impersonation only for those Web sites that are integrated with Access Manager. To do this, you must configure the Native Module only at those sites that require integration. See:

- [Configuring and Registering ImpersonationModule to IIS.](#)

49.6.5.1 Configuring and Registering ImpersonationModule to IIS.

You can configure and register ImpersonationModule to IIS

To configure and register ImpersonationModule to IIS

1. Select **Start, Administrative Tools, Internet Information Services (IIS) Manager.**
2. In the left pane of IIS, click the **hostname.**
3. In the middle pane, under the IIS header, double click **Modules.**
4. In the right pane, click **Configure Native Modules** and click **Register.**
5. In the window, provide a module Name (for example, *Oracle Impersonation Module*).
6. In the **Path** field, type the full path to IISImpersonationModule.dll.

By default, the path is:

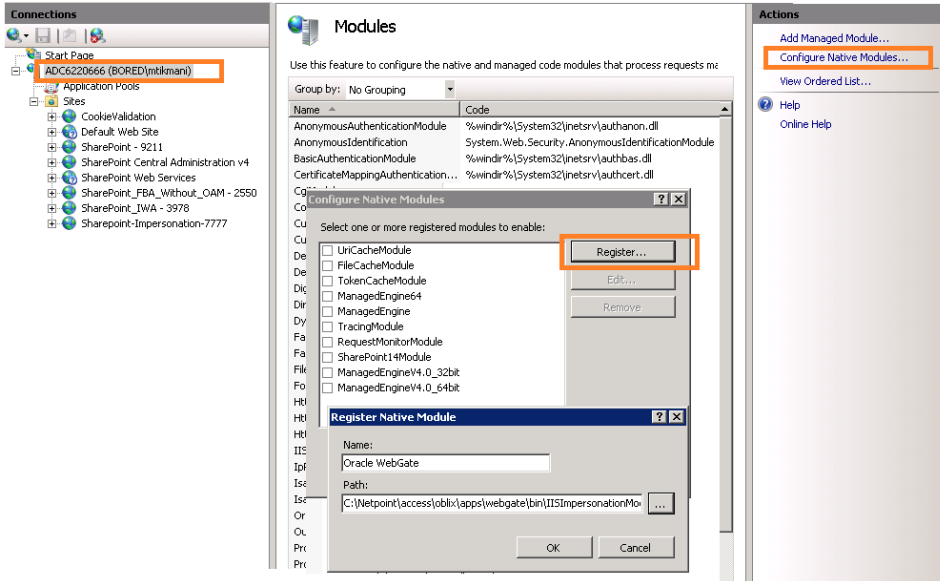
```
WebGate_Install_DIR\webgate\iis\lib\Impersonation.dll
```

Where *WebGate_install_dir* is the directory of your WebGate installation.

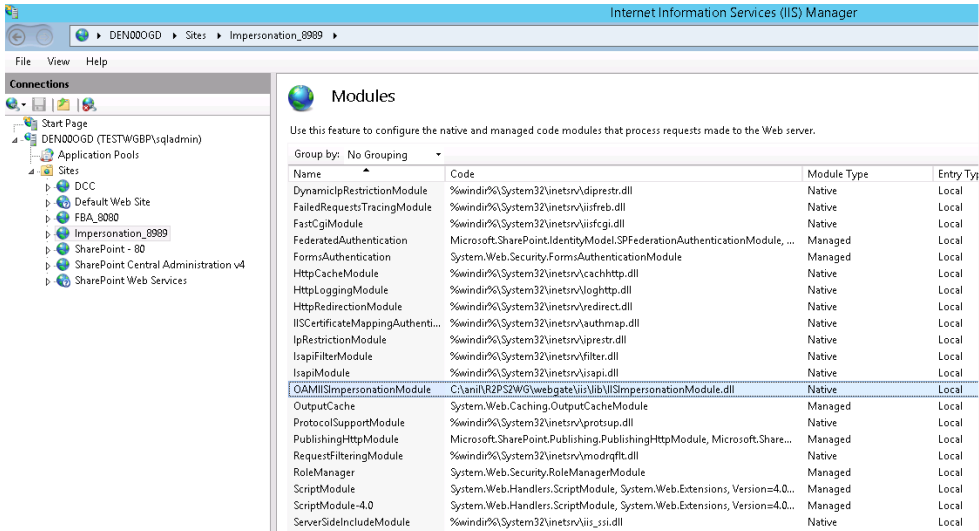
 **Note:**

If any spaces exist in the path (for example, C:\Program Files\Oracle\...) surround the entire string with double quotes (" ").

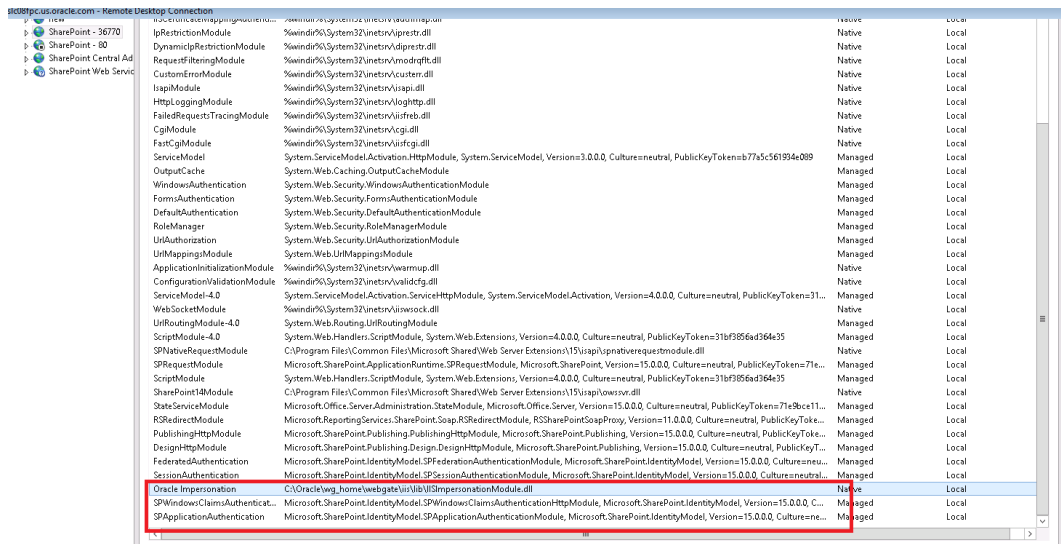
7. Click **OK** to register the module.
8. Check the name of the newly created module and click **OK** to apply the module across the Web sites.
9. Add IISImpersonationModule.dll as Native Module on the IIS.



10. Configure the IISImpersonationModule.dll as Native Module at site level only (not at global level).



11. IISImpersonationModule.dll must be above SPWindowsClaimAuthenticationHttpModule in Http module at site level for the integration. In order to move IISImpersonationModule.dll above SPWindowsClaimAuthenticationHttpModule, all the native module must be unlocked from the Http module at IIS Global level.



49.6.6 Testing Impersonation

You can test to ensure that impersonation is working properly before you complete the integration.

Following are the ways to test impersonation:

- Outside the SharePoint Server context or test single sign-on, as described in "[Creating an IIS Virtual Site Not Protected by SharePoint Server](#)"
- Using the Event Viewer, as described in "[Testing Impersonation Using the Event Viewer](#)"
- Using a Web page, as described in "[Testing Impersonation using a Web Page](#)"
- Using negative testing as described in "[Negative Testing for Impersonation](#)"



See Also:

"[Completing the SharePoint Server Integration](#)" after confirming impersonation configuration is working properly

49.6.6.1 Creating an IIS Virtual Site Not Protected by SharePoint Server

To test the impersonation feature outside the SharePoint Server context or to test single sign-on, you will need a target Web page on an IIS virtual Web site that is not protected by SharePoint Server.

You create such a virtual Web site by completing the following task.

To create an IIS virtual site not protected by SharePoint Server

1. Select Start, Administrative Tools, Internet Information Services (IIS) Manager.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Right-click Web Sites on the tree in the left pane, then navigate to New, Web Site on the menu.

4. Respond to the prompts by the Web site creation wizard.
5. After you create the virtual site, you must protect it with policies in an Application Domain.

49.6.6.2 Testing Impersonation Using the Event Viewer

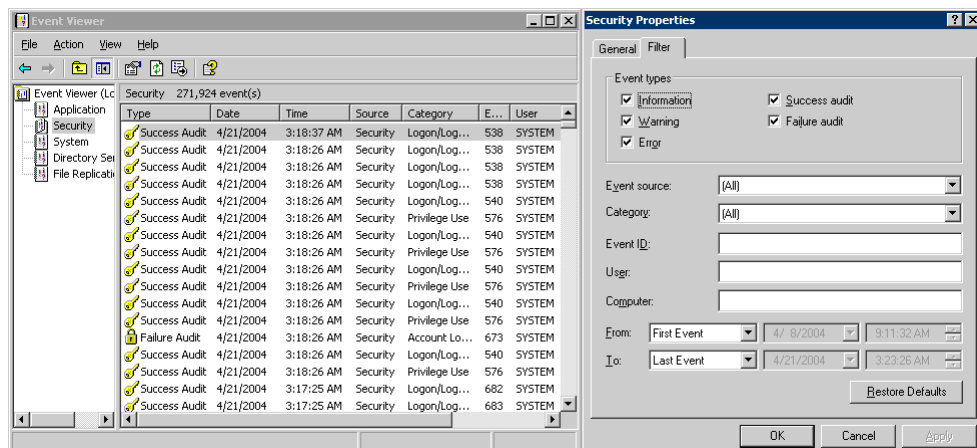
When you complete impersonation testing using the Windows 2003 Event Viewer, you must configure the event viewer before conducting the actual test.

To test impersonation through the Event Viewer:

1. Select **Start Menu, Event Viewer**.
2. In the left pane, right-click **Security**, then click **Properties**.
3. Click the **Filter** tab on the **Security** property sheet.
4. Verify that all **Event Types** are checked, and the **Event Source and Category** lists are set to **All**, then click **OK** to dismiss the property sheet.

Your Event Viewer is now configured to display information about the HeaderVar associated with a resource request.

Figure 49-3 Verifying Event Viewer Settings



5. Create a new IIS virtual server (virtual site).
6. Place a target Web page anywhere in the tree on the virtual site.
7. Point your browser at the Web page.

If impersonation is working correctly, the Event Viewer will report the success of the access attempt.

49.6.6.3 Testing Impersonation using a Web Page

You can also test impersonation using a dynamic test page, such as an .asp page or a Perl script, that can return and display information about the request.

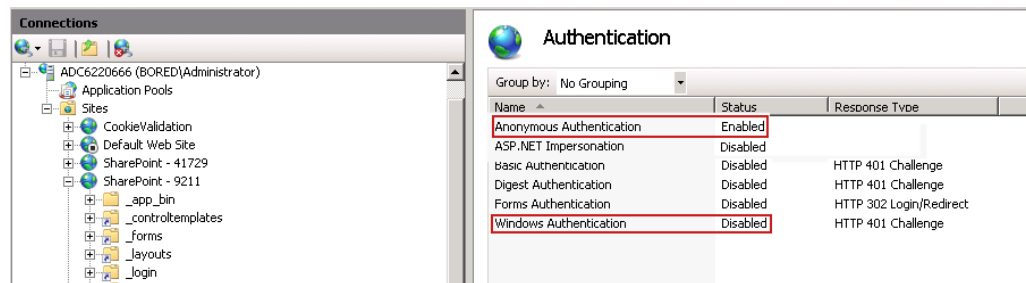
To test impersonation through a Web page that displays server variables

1. Create an .asp page or Perl script that will display the parameters AUTH_USER and IMPERSONATE, which can resemble the sample page presented in the following listing:

To configure IIS Security for the SharePoint Server integration

1. Select **Start, Administrative Tools, Internet Information Services (IIS) Manager**.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Click **Web Sites** on the tree in the left pane.
4. In the center pane, double-click on **Authentication** under **IIS**.
5. Ensure that **Anonymous Authentication** is enabled and **Windows Authentication** is disabled.

Figure 49-4 Impersonation Authentication



49.8 Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider

In this scenario, Access Manager gets integrated with SharePoint Server using SharePoint Security Token Service (STS). This includes the ISAPI WebGate installation on IIS, as well as Access Manager configuration and steps needed to achieve the HeaderVar integration.



Note:

Only 64-bit ISAPI WebGates are supported for this integration.

The following overview introduces the tasks that you must perform for this integration, including prerequisites, and where to find the information you need for each task.

Task overview: Integrating with Microsoft SharePoint Server Configured with LDAP Membership Provider

1. Preparing for this integration:
 - a. Install "[Required Microsoft Components](#)", as described.
 - b. Create a SharePoint Web site, as described in "[Creating a New Web Application in Microsoft SharePoint Server](#)".
 - c. Configure the SharePoint site collection, as described in "[Creating a New Site Collection for Microsoft SharePoint Server](#)".

- d. Configure the created Web site with LDAP directory using Claim-Based Authentication type (which uses the LDAP Membership Provider), as described in your SharePoint documentation.
 - e. Ensure that users who are present in the LDAP directory can log in to the SharePoint Web site and get proper roles.
 - f. Test the configuration to ensure that users who are present in the LDAP directory can log in to the SharePoint Web site and get proper roles, as described in your SharePoint documentation.
2. Perform all tasks described in "[Installing Access Manager for Microsoft SharePoint Server Configured With LDAP Membership Provider](#)".
This task includes installing a 12c WebGate for IIS and configuring a `WebGate.dll` for the individual SharePoint Web site.
 3. Add an authentication scheme for this integration, as described in "[Configuring an Authentication Scheme for Use With LDAP Membership Provider](#)".
 4. Synchronize directory servers, if needed, as described in "[Ensuring Directory Servers are Synchronized](#)".
 5. Configure single-sign-on for office documents as described in "[Configuring Single Sign-On for Office Documents](#)".
 6. Configure single sign-off, as described in "[Configuring Single Sign-off for Microsoft SharePoint Server](#)".
 7. Finish by testing your integration to ensure it operates without problem, as described in "[Testing the Integration](#)".

49.8.1 About Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider

The previous scenario, "[Integrating With Microsoft SharePoint Server](#)", describes how to use Windows authentication. In that scenario, authentication and authorization are performed for users residing in Active Directory. Access Manager used Windows impersonation for integration.

For the integration described in this section, support for the LDAP Membership Provider is achieved by using a HeaderVar-based integration. The ISAPI WebGate filter intercepts HTTP requests for Web resources and works with the OAM Server to authenticate the user who made the request. When authentication is successful, WebGate creates an `OAMAuthnCookie` and sends it to the user's browser to facilitate single sign-on (SSO). The WebGate also sets `OAM_REMOTE_USER` as a HeaderVar action for this user session. The Oracle Custom Membership provider in SharePoint validates the `ObSSOCookie` using the HTTP validation method, whereby the Access Manager Custom Membership Provider makes an HTTP/HTTPS request to a protected resource. Access Manager then validates and compares the user login returned on Authorization success with `OAM_REMOTE_USER`.



See Also:

"[Introduction to Integrating With the SharePoint Server](#)" for a look at processing differences between this integration and the other integrations described in this chapter.

Requirements: This integration requires that Microsoft SharePoint Server:

- Must be integrated with the LDAP Membership Provider. Must not use Windows authentication
- Must not have `IISImpersonationModule.dll` configured at the Web site using Claim Based Authentication



See Also:

["Integration Requirements"](#)

49.8.2 Installing Access Manager for Microsoft SharePoint Server Configured With LDAP Membership Provider

You can prepare your installation for integration with Microsoft SharePoint Server Configured with LDAP Membership Provider.

Prerequisites

Perform Step 1 of the previous ["About Integrating With Microsoft SharePoint Server Configured With LDAP Membership Provider"](#).

To prepare your deployment for integration that includes LDAP Membership Provider

1. Install Oracle Identity Management and Access Manager.
2. Provision and install an IIS WebGate.
3. Configure `Webgate.dll` at the SharePoint Web site that you want to protect. For example:
 - a. Start the Internet Information Services (IIS) Manager: Click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager
 - b. Under Web Sites, double click the name of the SharePoint Web site to protect.
 - c. In the Middle pane, double click IIS Filters and click Add in the right pane.
 - d. Enter the filter name as **Oracle WebGate**.
 - e. Enter the following path to the `Webgate.dll` file.

```
WebGate_install_dir\webgate\iis\lib
```
 - f. Save and apply these changes.
 - g. Double click **Authentication** in the middle pane.
 - h. Verify that the following Internet Information Services settings are correct: **Anonymous Authentication** and **Forms Authentication** is enabled, and **Windows Authentication** is disabled.

 **Note:**

For Claim-based Authentication to work with Access Manager, Windows Authentication for the SharePoint Site must be disabled.

- i. **Save** and **Apply** these changes.
4. Execute `ConfigureIISWebGate.bat` tool on the IIS website.

 **Note:**

Site name must be alphanumeric while executing `ConfigureIISwebgate.bat` tool.

Example: `configureIISwebgate -w C:\Oracle_WebGateInstance -oh C:\Oracle_WebGateHome -site "Default WebSite"`

5. Proceed to "[Configuring an Authentication Scheme for Use With LDAP Membership Provider](#)".

49.8.3 Configuring an Authentication Scheme for Use With LDAP Membership Provider

When your integration includes the LDAP Membership Provider, only three Access Manager authentication methods are supported.

To configure an authentication scheme for SharePoint with LDAP Membership Provider:

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the **Launch Pad** tab, select **Create Authentication Scheme** from the **Create (+)** drop-down menu the **Access Manager** section.
3. On the Authentication Scheme page, fill in the:
 - Name: Enter a unique name for this scheme. For example: *SharePoint w/LDAP-MP*
 - Description: Optional
4. Authentication Level: Choose a level of security for the scheme.
5. Choose a Challenge Method:
 - Basic Authentication for SharePoint Web site root (/)**
 - Form Authentication with Challenge Redirect for SharePoint Web site root (/)**
 - Client Certificate Authentication for SharePoint Web site root (/)**
6. Challenge Redirect: Enter your challenge redirect value, if required.
7. Choose an Authentication Module from those listed.
8. Challenge Parameters: Enter your challenge parameter values, if required.
9. Challenge URL: The URL the credential collector will redirect to for credential collection.
10. Click **Apply** to submit the new scheme, review details in the Confirmation window.

11. **Optional:** Click the **Set as Default** button to automatically use this with new Application Domains, then close the Confirmation window.
12. In the navigation tree, confirm the new scheme is listed, and then close the page.

49.8.4 Integrating SharePoint Server with OAM 12c using FBA

Access Manager is integrated with SharePoint 2013 OAM 12c using FBA (OAMCustomMembershipProvider). The following steps must be performed for this integration:

1. Update the WebGate profile (used for the integration). Enter user defined parameter `IISIntegrationMode=true`.
2. For Office Integration and "Open with explorer" feature support, update Authentication scheme "Challenge Parameter" with `ssoCookie=max-age=<time in seconds>`, to make `OAMAuthnCookie` as persistent cookie and to propagate it to WebDAV browser.
3. Add `OAMCustomMembershipProvider.dll` to GAC (Global Assembly Cache) using the following command:

```
gacutil -I OAMCustomMembershipProvider.dll
```

4. Update the `web.config` of the SharePoint Web Services `SecureTokenServiceApplication`, substitute the following line in `SecureTokenServiceApplication` Membership provider.

Note: Sample `web.config` can be found at `<physical path of the SecurityTokenServiceApplication>\web.config`. For example, `C:\Program Files\Common Files\microsoft shared\Web Server Extensions\16\WebServices\SecurityToken\web.config`.

```
type="Microsoft.Office.Server.Security.LdapMembershipProvider,
Microsoft.Office.Server, Version=15.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c"
```

```
type="Oracle.OAMCustomMembershipProvider,
OAMCustomMembershipProvider, Version=12.2.1.4, Culture=neutral,
PublicKeyToken=52e6b93f6f0427a1"
```

Note: Run the following `gacutil` command to get the version number to be updated in the `web.config` file:

```
.\gacutil.exe -l OAMCustomMembershipProvider
```

5. Update the SharePoint site `default.aspx` and add the following content after `</asp:login>` tag.

Note: Sample `default.aspx` can be found at `<physical path of the site>_forms\default.aspx`. For example, `C:\inetpub\wwwroot\wss\VirtualDirectories\3823_forms\default.aspx`.

```
<asp:HiddenField EnableViewState="false" ID="loginTracker"
runat="server" Value="autoLogin" />
```



```

<%
bool autoLogin = loginTracker.Value == "autoLogin";
%>
<script runat="server">
void Page_Load()
{
    signInControl.LoginError += new EventHandler(OnLoginError);
    NameValueCollection headers = Request.ServerVariables;
    NameValueCollection queryString = Request.QueryString;
    string loginasanotheruser = queryString.Get("loginasanotheruser");
    string username =
Request.ServerVariables.Get("HTTP_OAM_REMOTE_USER");
    bool isOAMCredsPresent = username != null && username.Length > 0;
    bool signInAsDifferentUser = loginasanotheruser != null &&
loginasanotheruser.Contains("true");
    if (isOAMCredsPresent)
    {
        //Handling For UTF-8 Encoding in HeaderName
        if (username.StartsWith("=?UTF-8?B?") && username.EndsWith("=?="))
        {
            username = username.Substring("=?UTF-8?B?".Length,
username.Length - 12);
            byte[] decodedBytes = Convert.FromBase64String(username);
            username = Encoding.UTF8.GetString(decodedBytes);
        }
    }
    if (isOAMCredsPresent && loginTracker.Value == "autoLogin" && !
signInAsDifferentUser)
    {
        bool
status=Microsoft.SharePoint.IdentityModel.SPClaimsUtility.AuthenticateForm
sUser(new Uri(SPContext.Current.Site.Url),username,"dummy");
        if(status)
        {
            if (Context.Request.QueryString.Keys.Count > 1)
            {
                Response.Redirect(Context.Request.QueryString["Source"].ToString());
            }
            else
                Response.Redirect(Context.Request.QueryString["ReturnUrl"].ToString());
        }
        else
        {
            loginTracker.Value = "";
        }
    }
}
void OnLoginError(object sender, EventArgs e)
{
    loginTracker.Value = "";
}
</script>
</asp:Content>

```

 **Note:**

After performing the above steps, if the following error is observed during login then revert the configuration changes done in steps 4 and 5 and try to access SharePoint site. The site should be accessible after OAM login and SharePoint login. If the SharePoint Site is still not accessible, then [Configure alternate access mappings for SharePoint Server](#) and try to access SharePoint site, now the site should be accessible after OAM login and SharePoint login. Reconfigure steps 4 and 5.

```

Line 73: {
Line 74:
Line 75: bool
status=Microsoft.SharePoint.IdentityModel.SPClaimsUtility.AuthenticateFormsUser(new
Uri(SPContext.Current.Site.Url),username,"dummy");
Line 76: if(status){
Line 77: if (Context.Request.QueryString.Keys.Count > 1)
Source File:
c:\inetpub\wwwroot\wss\VirtualDirectories\3261\_forms\Default.aspx Line:75
Stack Trace:
[NullReferenceException: Object reference not set to an instance of an object.]
ASP._forms_default.aspx.Page_Load() in
c:\inetpub\wwwroot\wss\VirtualDirectories\3261\_forms\Default.aspx:75
Microsoft.SharePoint.WebControls.UnsecuredLayoutsPageBase.OnLoad(EventArgs e) +299
Microsoft.SharePoint.IdentityModel.Pages.FormsSignInPage.OnLoad(EventArgs e) +18
System.Web.UI.Control.LoadRecursive() +94
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +2935

```

6. Remove OracleWebGateExtension from HandlerMapping of IIS Site.

49.8.5 Ensuring Directory Servers are Synchronized

Users in the directory server configured for Access Manager should be synchronized with the directory server used by SharePoint if these are different.

This is the same task that you perform for other integration scenarios in this chapter. When your SharePoint integration includes an LDAP Membership Provider, however, you can use a directory server that supports LDAP commands.

 **See Also:**

["Synchronizing User Profiles Between Directories"](#)

49.8.6 Testing the Integration

This is similar to the task you perform for other integration scenarios in this chapter. There are no differences when configured with LDAP Membership Provider.



See Also:

["Testing the SharePoint Server Integration"](#)

49.9 Configuring Single Sign-On for Office Documents

Single sign-on for Office documents can be achieved by setting a persistent cookie in the authentication scheme.

To do this using OAM 12c, you need to set `ssoCookie=max-age` in the authentication scheme. This creates a persistent cookie which lasts for more than one session.



Note:

For integration based on Windows Native Authentication, you do not need to set the persistent cookie parameter.

1. Log in to the Oracle Access Management Console.
2. Find the Authentication Scheme being used and open the page.
3. In the **Challenge Parameter**, add:

```
ssoCookie=max-age=1000000
```

Where, `time-in-seconds` represents the time interval when the cookie expires. For example, `ssoCookie=max-age=3600` sets the cookie to expire in 1 hour (3600 seconds).

4. Save the change.
5. Configure centralized logout for the OAM Webgate.

49.10 Configuring Single Sign-off for Microsoft SharePoint Server

Manual Logout occurs when the user clicks the Logout button from SharePoint Server. You can configure the SharePoint Server logout URL in Access Manager so that when a user clicks the Logout button from SharePoint Server site, Access Manager logout is also triggered.

Closing the browser window after sign-off is always recommended, for security. Cookie time-out occurs when the overall user session is controlled by `OAMAuthnCookie`. Consider the following use-case:

- FedAuth cookie time-out and OAMAuthnCookie is still valid: The user won't be challenged again because the OAMAuthnCookie is present. A new FedAuth cookie is generated (using the same flow described earlier).
- OAMAuthnCookie time-out and FedAuth Cookie is still valid: Since each request is intercepted by the WebGate, the user is challenged for credentials again.

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

This topic describes how to configure single sign-off for integration with SharePoint. Single sign-off kills the user session.

- [Configuring a Custom Logout URL in SharePoint Server](#)
- [Configuring Logout in SharePoint Server With Impersonation](#)

49.10.1 Configuring a Custom Logout URL in SharePoint Server

You can configure a custom logout URL in SharePoint Server.

To configure:

1. From the generated artifacts for WebGate, add `logout.html` to the SharePoint Server Site
2. Locate `C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\CONTROLTEMPLATES`.
3. In `\CONTROLTEMPLATES`, change the `welcome.ascx` by adding the following tag. For example:


```
<SharePoint:MenuItemTemplate runat="server" id="ID_OverrideLogout"
Text="Custom Logout"
    ClientOnClickNavigateUrl="/logout.html?end_url=_layouts/
SignOut.aspx"
    Description="My Custom Logout"
    MenuGroupId="200"
    Sequence="100"
    UseShortId="true" />
```
4. Click Save.
5. Protect the two URLs `/_layouts/SignOut.aspx` and `/_layouts/closeConnection.aspx` in an Application Domain using Anonymous authentication.
6. Proceed to [Configuring Logout in SharePoint Server With Impersonation](#).

49.10.2 Configuring Logout in SharePoint Server With Impersonation

You can configure logout in SharePoint server with Impersonation. If you do not have Impersonation configured, this procedure can be skipped.

To configure:

1. Copy `signout.aspx` from `C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\LAYOUTS`) to `MySignout.aspx` in the same path.

2. In `MySignout.aspx`, below (`<asp:content contentplaceholderid="PlaceHolderAdditionalPageHead" runat="server">`) add the following script details:

```
<script runat="server">
private void Page_Load(object sender, System.EventArgs e){
    Response.Status = "302 Moved Temporarily";
    Response.AddHeader("Location", "/logout.html?end_url=/_layouts/SignOut.aspx");
}
</script>
```

3. Save.
4. Use this URL `_layouts/MySignout.aspx` as custom logout URL for SharePoint Server in the case of Impersonation.
5. Proceed with "[Testing Your Integration](#)".

49.11 Setting Up Access Manager and Windows Native Authentication

This section provides the following topics:

- [Setting Up Access Manager WNA](#)
- [Setting Up WNA With SharePoint Server](#)
- [Installing Access Manager for WNA and SharePoint Server](#)
- [Testing Your WNA Implementation](#)

49.11.1 Setting Up Access Manager WNA

Configure Access Manager to use Windows Native Authentication.

49.11.2 Setting Up WNA With SharePoint Server

The following overview outlines the tasks that must be performed to set up WNA with Access Manager and the SharePoint Server.

Task overview: Setting up WNA with SharePoint Server

1. Complete the following prerequisite tasks:
 - Perform tasks in "[Required Microsoft Components](#)".
 - Create a SharePoint Web site, as described in "[Creating a New Web Application in Microsoft SharePoint Server](#)".
 - Configure the SharePoint site collection, as described in "[Creating a New Site Collection for Microsoft SharePoint Server](#)".
 - Test the configuration to ensure that users who are present in the directory server can log in to the SharePoint Web site and get proper roles, as described in your SharePoint documentation.
2. Install Access Manager as described in "[Installing Access Manager for WNA and SharePoint Server](#)".

This step includes installing the WebGate for IIS and configuring `webgate.dll` for the individual SharePoint Web site.

3. Configure the Active Directory authentication provider, as follows:
 - a. Login to the WebLogic Console.
 - b. Go to Security Realm and click the realm being used.
 - c. Go to the Provider tab provider, click New.
 - d. Enter the provider name, select the Type **ActiveDirectoryAuthenticator**, click OK.
 - e. Select the newly created Provider, change Control Flag to Sufficient, and Save.
 - f. Go to Provider Specific tab, enter details for your Active Directory, and save these.
4. Perform "[Testing Your WNA Implementation](#)".

49.11.3 Installing Access Manager for WNA and SharePoint Server

You perform this task after you perform all prerequisites described in step 1 of the "[Setting Up WNA With SharePoint Server](#)". Installing most Access Manager components for this integration scenario is the same as for any other situation.

Installing the IIS WebGate is similar to installing any other WebGate. The WebGate should be installed with the IIS v7 Web server; later it can be configured at the specific SharePoint Web site level to be protected. For IIS, the WebGate must be configured at the "web sites" level. For Microsoft SharePoint Server, you must configure the WebGate for the specific SharePoint Web site level to be protected.

To install Access Manager for WNA and SharePoint Server

1. Install Access Manager as described in the *Installing and Configuring Oracle Identity and Access Management*.
2. Install the ISAPI WebGate as follows:
 - Installing WebGates
 - Installing Web components for the IIS Web server

Next, you configure `Webgate.dll` at the SharePoint Web site that you want to protect. Configuring `Webgate.dll` at the "Website level" protects all Web sites on the IIS Web server. However, configuring `Webgate.dll` at the "SharePoint Website" protects only the expected Web site.
3. Configure `Webgate.dll` at the SharePoint Web site that you want to protect. For example:
 - 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 host name 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 the name of 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
```
4. 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. In the **Alias** field, specify access and the physical path to the WebGate \access folder (or click the ellipsis button (...), go to the \access folder, then click **OK**).


```
WebGate_install_dir\access\
```
 5. Set permissions on 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. Select **Read**, **Script**, and **Execute**, then click **OK**
 6. Configure Access Manager to use Windows Native Authentication.
 7. Configure Microsoft SharePoint Server Authentication to Classic Mode Authentication while creating a new Web Application in Microsoft SharePoint. In the Authentication Provider section, select Negotiate(Kerberos).
 8. Go to IIS newly created SharePoint site and:
 - a. Open **Authentication, Windows Authentication, Advance Settings**.
 - b. Select **Enable Kernel mode authentication**.
 - c. Select providers, delete NTLM provider.
 - d. Add **Negotiate:Kerberos** and move it to the top level.
 - e. Restart IIS.
 9. Proceed to "[Testing Your WNA Implementation](#)".

49.11.4 Testing Your WNA Implementation

Use the following steps to confirm your WNA implementation is working properly.

To test your WNA implementation

1. Log in to the machine as the Windows domain user (or AD user or AD user account).
The login account must also be a user of Access Manager.
2. Enter the URL of the protected resource.

49.12 Synchronizing User Profiles Between Directories

You need to synchronize user profiles between the SharePoint Server directory and the Access Manager directory.

Unless explicitly stated, this task should be performed for all integration scenarios in this chapter.



Note:

When your integration includes LDAP Membership Provider, you can use any directory server that supports LDAP commands.

To synchronize:

- **Uploading user data**—If your Access Manager installation is configured for any directory server other than SharePoint Active Directory, you must load the user profiles that reside on the other directory server to SharePoint Active Directory.

Proceed to "[Testing Your Integration](#)"

49.13 Testing Your Integration

After you complete the tasks to enable integration, you should test to verify that integration is working.

This section contains the following topics:

- [Testing the SharePoint Server Integration](#)
- [Testing Single Sign-On for the SharePoint Server Integration](#)

49.13.1 Testing the SharePoint Server Integration

You can verify that a user can access SharePoint Server resources through Access Manager authentication and SharePoint Server authorization.

To test your SharePoint Server integration:

1. Navigate to any SharePoint Server Web page using your browser.
You are challenged for your credentials.
2. Log in by supplying the necessary credentials.
3. Verify that the page you requested is visible.
4. **Optional:** Check the Event Viewer to confirm that the access request was successful.

49.13.2 Testing Single Sign-On for the SharePoint Server Integration

You can also test single sign-on by demonstrating that a user who has just supplied credentials and accessed an SharePoint Server resource can (before the

OAMAuthnCookie expires) access a non-SharePoint Server resource without having to supply credentials a second time.

For example, use a resource defined in the Policy Manager. When single sign-on is working, you should be granted access to the page without having to supply credentials a second time.

To test single sign-on for your SharePoint Server integration:

1. Create and protect a new virtual site with a Application Domain (or use one you have already created).
2. Place a Web page anywhere in the tree of this virtual site.
3. Using a browser, navigate to the page in the new virtual site.

If you have already passed authentication, you should be granted access to the page without having to supply credentials a second time.

49.14 Troubleshooting

- [Internet Explorer File Downloads Over SSL Might Not Work](#)

49.14.1 Internet Explorer File Downloads Over SSL Might Not Work

The issue may occur if the server sends a `Cache-control:no-store` header or sends a `Cache-control:no-cache` header. The WebGate provides configuration parameters to control setting these headers.

Following are the parameters and their default value:

```
CachePragmaHeader no-cache
```

```
CacheControlHeader no-cache
```

You can modify the WebGate configuration not to set these headers at all (the values for these parameters would be kept blank). By this, it would mean that Access Manager will not control the caching behavior.

Integrating Access Manager with Outlook Web Application

In a Windows environment, after a user authenticates, the authenticating application can impersonate that user's identity. The primary purpose of impersonation is to trigger access checks against a client's identity.

This chapter focuses on how to enable impersonation in Access Manager to override impersonation enabled with IIS. The following topics describes support for integration between Access Manager and Outlook Web Application (OWA) 2010, 2016, and 2019:

Note:

This chapter describes the configuration steps for Outlook Web Application (OWA), using OWA 2010 as an example. Similar configuration steps (not included in this documentation) may apply to OWA 2016 and 2019.

- [What is New in This Release?](#)
- [Introduction to Integration with Outlook Web Application](#)
- [Enabling Impersonation With a Header Variable](#)
- [Setting Up Impersonation for Outlook Web Application \(OWA\)](#)
- [Setting Up Access Manager WNA for Outlook Web Application](#)

50.1 What is New in This Release?

Support for integration between Access Manager and Outlook Web Application (OWA).

This chapter illustrates:

- [Enabling Impersonation With a Header Variable](#)
- [Setting Up Impersonation for Outlook Web Application \(OWA\)](#)
- [Setting Up Access Manager WNA for Outlook Web Application](#)

50.2 Introduction to Integration with Outlook Web Application

This section provides the following information to introduce the integration described in this chapter:

- [About Impersonation Provided by Microsoft Windows](#)
- [Access Manager 12c Support for Windows Impersonation](#)
- [Single Sign-On for Authenticated Access Manager Users into Exchange](#)
- [Confirming Requirements](#)

50.2.1 About Impersonation Provided by Microsoft Windows

Impersonation ensures that the server can or cannot do exactly what the client can or cannot do. When running in a client's security context, a service can to an extent become a client. After the user authenticates, the service can take on that user's identity through impersonation.

One of the service's threads uses an access token, known as an impersonation token, to obtain access to objects the client can access. The access token is a protected object that represents the client's credentials. 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. When the server is impersonating the client, any operations performed by the server are performed using the client's credentials.

Impersonation ensures that the server can or cannot do exactly what the client can or cannot do. Access to resources can be restricted or expanded, depending on what the client has permission to do. Impersonation requires the participation of both the client and the server. The client must indicate its willingness to let the server use its identity, and the server must explicitly assume the client's identity programmatically.

When impersonation concludes, the thread uses the primary token to operate using the service's own security context rather than the client's. The primary token describes the security context of the user account associated with the process (the person who started the application).

Services run under their own accounts and act as users in their own right. For example, system services that are installed with the operating system run under the Local System account. You can configure other services to run under the Local System account, or separate accounts on the local system or in Active Directory.

The IIS Web server provides impersonation capabilities. However, the OAM Server overrides IIS authentication, authorization, and impersonation functions. For more information, see "[Access Manager 12c Support for Windows Impersonation](#)" in the next section.

50.2.2 Access Manager 12c Support for Windows Impersonation

You can enable support for Windows impersonation to provide additional access control for protected applications.

You bind a trusted user to a Webgate and protect the application with a application domain that includes an impersonation action in the authorization rule. During the authorization process, the protected application creates an impersonation token.

For more information, see, [Enabling Impersonation With a Header Variable](#) It provides prerequisites and details about implementing impersonation using header variables.

50.2.3 Single Sign-On for Authenticated Access Manager Users into Exchange

Single Sign-On for Authenticated Access Manager Users into Exchange is also supported using the Windows Impersonation feature.

Outlook Web Access (OWA) provides Web access to Exchange mail services and may be configured on either of the following:

- An IIS Web server that does not reside on the same host as the Exchange server, which is also known as a front-end server
- An IIS Web server running on the same host as the Exchange server, which is also known as the back-end server

In a front-end server configuration, the front-end OWA server authenticates the user, determines the back-end Exchange server that hosts the user's mailbox, then proxies the request to the appropriate back-end Exchange server. No additional credential information is passed. No delegation is performed. Setting up Impersonation on the back-end Exchange server ensures that the Exchange server does not need to request credentials before granting access.

For more information, see [Setting Up Impersonation for Outlook Web Application \(OWA\)](#)

50.2.4 Confirming Requirements

Here is an example that illustrates setting up the impersonation feature for the OAM Server to Microsoft Exchange Server 2013 integration.

The principles are the same regardless of your application. Any references to specific versions and platforms in this chapter are for demonstration purposes. For the latest Access Manager certification information, see Oracle Technology Network at:

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

50.3 Enabling Impersonation With a Header Variable

Enabling impersonation with a header variable involves completing the procedures in the following sections.

1. [Reviewing all Requirements for Impersonation with a Header Variable](#)
2. [Creating an Impersonator as a Trusted User](#)
3. [Assigning Rights to the Trusted User](#)
4. [Binding the Trusted User to Your WebGate](#)
5. [Adding an Impersonation Response to An Application Domain](#)
6. [Adding an Impersonation DLL to IIS](#)
7. [Testing Impersonation](#)

**See Also:**

["Setting Up Impersonation for Outlook Web Application \(OWA\)".](#)

50.3.1 Requirements for Impersonation with a Header Variable

Prepare the environment and confirm that it is operating properly before implementing Windows impersonation with the OAM Server.

[Table 50-1](#) identifies the Access Manager platform requirements when you enable impersonation using a header variable.

Table 50-1 Requirements for Impersonation with a Header Variable

Item	Platform
OAM Webgate (and Impersonation dll)	Microsoft IIS 7.x and Windows Server 2008 and 2013
Impersonation dll	<i>Webgate_install_dir\webgate\iis\lib\IISImpersonationModule.dll</i> <ul style="list-style-type: none"> Must be installed as an IIS Module. May be installed at any level of the Web site tree.
Kerberos Key Distribution Center (KDC) and Active Directory	Windows Server 2008 and 2013
Client and Server machines	<ul style="list-style-type: none"> Both must be in the same Windows Server 2008 domain with a trust relationship. A bi-directional trust path is required because the service, acting on the client's behalf, must request tickets from the client's domain.
Security context	Must have <i>Act as operating system</i> privileges. Note: IWAM_Machine is not recommended
Mutual authentication is required	Mutual authentication is supported remotely.

50.3.2 Creating an Impersonator as a Trusted User

Whether you enable impersonation using a HeaderVar or user profile attribute, the return value must be a trusted user in Active Directory. This special user should not be used for anything other than impersonation.

The example in the following procedure uses *SPPSImpersonator* as the New Object - User. With *OWAImpersonator* as *SPPSImpersonator* denotes SharePoint impersonation specifically. Your environment will be different.

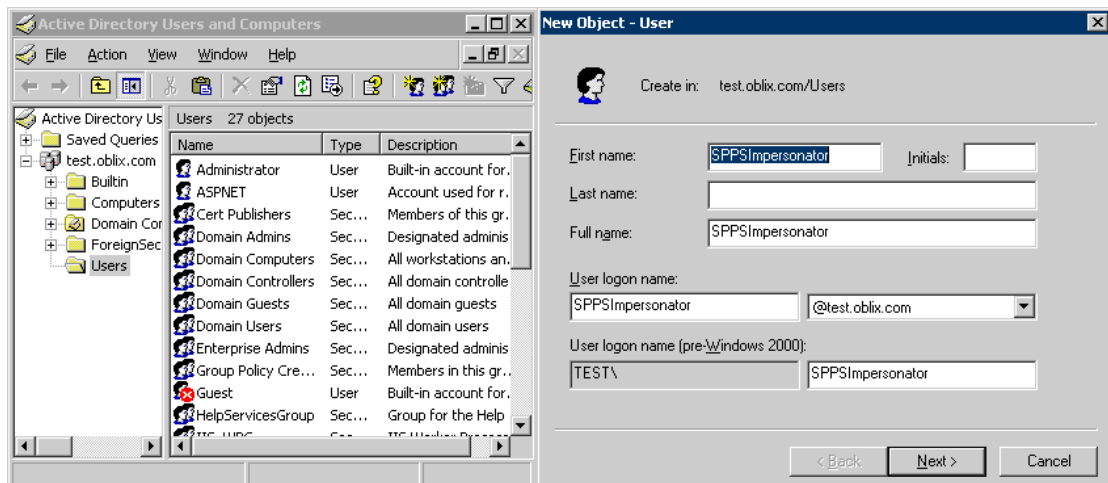
1. Perform the steps for your environment on the computer hosting your Microsoft Exchange Server 2013 installation:
 - Windows 2008 or 2012: Select Start, Programs, Administrative tools, Active Directory Users and Computers.
2. In the Active Directory Users and Computers window, right-click Users on the tree in the left pane, then select New; User.

3. In the First name field of the pane entitled New Object - User, enter an easy-to-remember name such as *SPPSImpersonator*.
4. Copy this same string to the User logon name field, then click Next.
5. In succeeding panels, you are asked to choose a password and then retype it to confirm.

 **Note:**

Oracle recommends that you choose a very complex password, because your trusted user is being given very powerful permissions. Also, be sure to check the box marked Password Never Expires. Since the impersonation module should be the only entity that ever sees the trusted user account, it would be very difficult for an outside agency to discover that the password has expired.

Figure 50-1 Setting up a Trusted User Account for Windows Impersonation



50.3.3 Assigning Rights to the Trusted User

You can give the trusted user the right to act as part of the operating system.

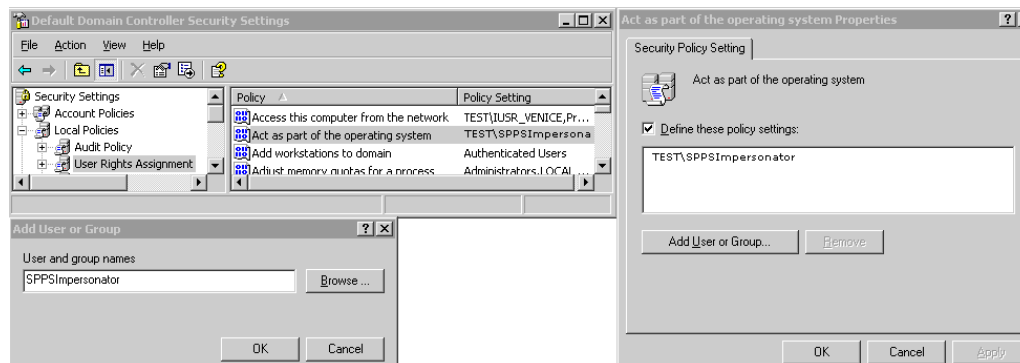
To assign rights to the trusted user:

1. Perform the appropriate step for your environment:
 - Windows 2008: Select Start > Programs > Administrative tools > Local Security Policy.

You must modify the group policy object that applies to the computer where the Webgate is installed.
2. On the tree in the left pane, click the plus icon (+) next to Local Policies.
3. Click User Rights Assignment on the tree in the left pane.
4. Double-click "Act as part of the operating system" in the right pane.
5. Click Add User or Group.

6. In the Add User or Group panel, type the User logon name of the trusted user (Microsoft Exchange Server 2010 Impersonator in our example) in the User and group names text entry box, then click OK to register the change.

Figure 50-2 Configuring Rights for the Trusted User in Windows Impersonation



50.3.4 Binding the Trusted User to Your WebGate

You need to bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user.

The procedure presumes that you have registered an OAM Webgate with Access Manager. The values in the procedure are provided as an example only. Your environment will be different.



See Also:

[Registering and Managing OAM Agents](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the **Launch Pad** tab, click **Agents**.
3. Find the desired OAM Webgate registration to modify for this integration:
 - **Find All Enabled:** Select **State All**, click the **Search** button, click the desired Webgate name in the results list.
4. On the WebGate registration page, enter the SharePoint username and password for the trusted user account, which you created earlier.
5. Click **Apply** to commit the changes.

A bind has been created for the WebGate and the trusted user. The WebGate is now ready to provide impersonation on demand. The demand is created by an Authorization Success Action in the application domain created for impersonation.

50.3.5 Adding an Impersonation Response to An Application Domain

You must create or configure an application domain to protect your OWA resources.

For this you must add Responses in Authorization Policies (Header type Responses), as described in this procedure.



See Also:

[Managing Policies to Protect Resources and Enable SSO](#)

The procedure presumes that you have an application domain created for the OAM Webgate you registered. The application domain in this example is *MyImpersonationDomain*. Your environment will be different.

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. Click **Application Domains** in the **Access Manager** section.
3. Search for and open the OWA Application Domain (the relevant application domain for impersonation).

Navigate as follows:

Authorization Policies
Protected Resource Policy
Responses

4. Click the **Add** button, then **Add Response**.

Complete the form as follows:

- From the **Type** list, choose **Header**.
- In the **Name** field, type a unique name for this response. For example, *IMPERSONATE*.
- In the **Value** field, type a value for this response. For example, *\$user.userid*.

5. Click **Add**, then click **Apply** to submit the changes.

This Response is used for the second WebGate request (for authorization).

50.3.6 Adding an Impersonation DLL to IIS

You are ready to configure IIS by adding the `IISImpersonationModule.dll` to your IIS configuration.

To add an Impersonation DLL to IIS:

1. Select **Start > Administrative Tools > Internet Information Services (IIS) Manager**.
2. In the left pane of IIS 7.x, click the hostname.
3. In the middle pane, under the "IIS" header, double click on "Modules".

4. In the right pane, click "Configure Native Modules" and click "Register".
5. In the window, provide a module **Name** (for example, *Oracle Impersonation Module*).
6. In the **Path** field, type the full path to `IISImpersonationModule.dll`.

By default, the path is:

```
Webgate_install_dir\webgate\iis\lib\IISImpersonationModule.dll
```

Where *Webgate_install_dir* is the directory of your WebGate installation.

 **Note:**

If any spaces exist in the path (for example, `C:\Program Files\Oracle\...`) surround the entire string with double quotes (" ").

7. Click **OK** to register the module.
8. Check the name of the newly created module and click **OK** to apply the module across the Web sites.
9. Remove the module from the **Default** site level (otherwise, it inherits when you add it on the machine level).
10. Ensure that the `IISImpersonationModule.dll` file added in these steps is applied only to "owa" and "ecp" applications and removed from the site level.

Go to OWA, double-click **modules, Configure Native Modules**, and check the desired module (for example, **Oracle Impersonation Module**).

Go to (ecp): Double-click **modules, Configure Native Modules**, and check the desired module (for example, **Oracle Impersonation Module**).

50.3.7 Testing Impersonation

You can test Impersonation using an Event Viewer or a web page.

Following are the two ways of testing Impersonation:

- [Testing Impersonation Using the Event Viewer](#)
- [Testing Impersonation using a Web Page](#)

50.3.7.1 Creating an IIS Virtual Site

To test the impersonation feature outside the Microsoft OWA 2010 context or to test single sign-on, you will need a target Web page on an IIS virtual Web site.

You create such a virtual Web site by performing the following task.

1. Click **Start > Administrative Tools > Internet Information Services (IIS) Manager**.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Right-click **Web Sites** on the tree in the left pane, then select **New**, then select **Web Site** on the menu.

4. Respond to the prompts by the Web site creation wizard.
5. After you create the virtual site, you must protect it with an application domain, as described elsewhere in this guide.

50.3.7.2 Testing Impersonation Using the Event Viewer

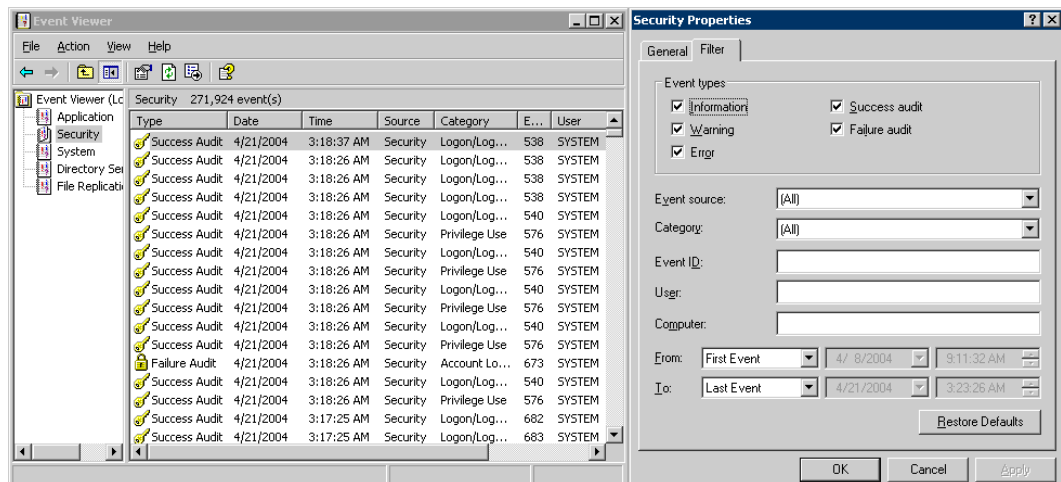
When you perform impersonation testing using the Windows 2008 Event Viewer, you must configure the event viewer before conducting the actual test.

To test impersonation:

1. Select **Start Menu > Event Viewer**.
2. In the left pane, right-click **Security**, then click **Properties**.
3. Click the **Filter** tab on the **Security** property sheet.
4. Verify that all Event Types are checked and the Event Source and Category lists are set to All, then click **OK** to dismiss the property sheet.

Your Event Viewer is now configured to display information about the headerVar associated with a resource request.

Figure 50-3 Verifying Event Viewer Settings



5. Create a new IIS virtual server (virtual site).
6. Place a target Web page anywhere in the tree on the virtual site.
7. Point your browser at the Web page

If impersonation is working correctly, the Event Viewer will report the success of the access attempt.

50.3.7.3 Testing Impersonation using a Web Page

You can test impersonation using a dynamic test page, such as a .asp page or a Perl script, that can return and display information about the request.

To test impersonation:

1. Create a .asp page or Perl script that will display the parameters AUTH_USER and IMPERSONATE. It can resemble the sample page presented in the following listing:

```

<TABLE border=1>
<TR>
<TD>Variable</TD>
<TD>&nbsp;&nbsp;&nbsp;</TD>
<TD>Value</TD></TR>
<%for each servervar in request.servervariables%>
<TR>
<TD><%=servervar%></TD>
<TD>&nbsp;&nbsp;&nbsp;</TD>
<TD><%=request.servervariables(servervar)%>&nbsp;&nbsp;&nbsp;</TD>
</TR>

```

2. Create an IIS virtual site, or use the one you created for the previous task.
3. Place a .asp page or Perl script (such as the sample in the preceding listing) anywhere in the tree of the new virtual site.
4. Point your browser at the page. The page should display, with both AUTH_USER and IMPERSONATE set to the name of the user making the request.

50.4 Setting Up Impersonation for Outlook Web Application (OWA)

In a distributed Exchange/OWA single sign-on environment, each server needs Access Manager to impersonate the current user. When you enable Impersonation, you need to include additional HTTP headers in the "Response" tab of the Authorization Policy of your impersonation application domain.

The following solution has been tested in both standalone and distributed OWA environments.

1. Install Access Manager 12c, as described in *Installing and Configuring Oracle Identity and Access Management*.
2. Install a OAM WebGate on all OWA client servers, as described in the *Administering Oracle Access Management*.
3. On the WebGate registration page, Disable IP Checking for Webgates on the back-end server using the AccessGate (because the request comes from the front-end server, not from the user's browser).
4. Ensure that OWA is not using Integrated Windows Authentication, as described in "[Prerequisites to Setting Impersonation for Outlook Web Application](#)".
5. Create a trusted user account for only impersonation in the Active Directory, as described in "[Creating a Trusted User Account for Outlook Web Application](#)".
6. Give the trusted user the special right to act as part of the operating system, as described in "[Assigning Rights to the Outlook Web Application Trusted User](#)".
7. Bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user, as described in "[Binding the Trusted Outlook Web Application User to Your WebGate](#)".
8. Add a header variable named *impersonate* to the Authorization Policy Response tab (in the impersonation application domain), as described in, as described in "[Adding an Impersonation Action to an Application Domain for Outlook Web Application](#)".

9. Configure IIS by adding `IISImpersonationModule.dll` to your IIS configuration, as described in "[Adding an Impersonation dll to IIS](#)".
10. Test Impersonation, as described in "[Testing Impersonation for Outlook Web Application](#)".

**See Also:**

"[Enabling Impersonation With a Header Variable](#)".

50.4.1 Prerequisites to Setting Impersonation for Outlook Web Application

Before you proceed with impersonation setup for Outlook Web Application, ensure that OWA is not using Integrated Windows (or any other) Authentication.

If it is not, you can use the following steps to set up OWA with Windows Authentication.

1. Open Exchange Management console.
2. Go to Server Configuration and click Client Access.
3. Select Outlook Web Access and click Properties.
4. In the Properties dialog box, click the Authentication tab.
5. Clear (unselect) all the authentication methods.
6. Click Apply, and click OK.
7. Restart the IIS server.
8. Proceed with "[Creating a Trusted User Account for Outlook Web Application](#)".

50.4.2 Creating a Trusted User Account for Outlook Web Application

The special user should not be used for anything other than impersonation. Oracle recommends that you chose a very complex password, because your trusted user is being given very powerful permissions.

Also, be sure to check the box marked Password Never Expires. Since the impersonation module should be the only entity that ever sees the trusted user account, it would be very difficult for an outside agency to discover that the password has expired.

To create a Trusted User Account for Outlook Web Application:

1. On the Windows 2008 machine, select Start; Programs; Administrative tools, Active Directory Users and Computers.
2. In the Active Directory Users and Computers window, right-click Users on the tree in the left pane, then select New; User.
3. In the First name field of the pane entitled New Object - User, enter an easy-to-remember name such as OWAImpersonator.
4. Copy this same string to the User logon name field, then click Next.
5. In succeeding panels, you will be asked to choose a password and then retype it to confirm.
6. Proceed to "[Assigning Rights to the Outlook Web Application Trusted User](#)".

50.4.3 Assigning Rights to the Outlook Web Application Trusted User

You need to give the trusted user the right to act as part of the operating system.

To assign rights to the Outlook Web Application trusted user:

1. Select Control Panel, Administrative Tools; and click either the Domain Controller Security Policy (if the computer is a domain controller) or Local Security Policy.
2. On the tree in the left pane, click the plus icon (+) next to Local Policies.
3. Click User Rights Assignment on the tree in the left pane.
4. Double-click "Act as part of the operating system" in the right pane.
5. Click Add User or Group.
6. In the Add User or Group panel, type the User logon name of the trusted user (OWAImpersonator in our example) in the User and group names text entry box, then click OK to register the change.
7. Proceed to "[Binding the Trusted Outlook Web Application User to Your WebGate.](#)"

50.4.4 Binding the Trusted Outlook Web Application User to Your WebGate

You need to bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user.

When the bind has been created for the WebGate and the trusted user, WebGate is ready to provide impersonation on demand. The demand is created by a Response set in the Authorization Policy of application domain created for impersonation.

The following procedure presumes that you have registered a 12c WebGate (*ImpersonateAgent*) with Access Manager. The values in the following procedure are provided as an example only. Your environment will be different.



See Also:

[Registering and Managing OAM Agents](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. in the **Launch Pad** tab, click **Agents**.
3. Find the desired OAM WebGate registration to modify for this integration. For example: ImpersonateAgent.
 - **Find All Enabled:** Select State All, click the Search button, click the desired Webgate name in the results list.
4. Open the Webgate registration page and enter the SharePoint username and password for the trusted user account, which you created earlier.
5. Click **Apply** to commit the changes.

A bind has been created for the Webgate and the trusted user. The Webgate is now ready to provide impersonation on demand. The demand is created by an Authorization Success Action in the application domain created for impersonation.

50.4.5 Adding an Impersonation Action to an Application Domain for Outlook Web Application

You must create or configure a application domain to protect your OWA resources (/owa and /ecp only).

Ensure that `IISImpersonation Module.dll` is applied only to "owa" and "ecp" applications in IIS7.x, and removed from the site level. The Authorization policy must set several HTTP Header variables (Header type Responses in the Authorization policy).

This procedure presumes that you have an existing application domain for the OAM WebGate (*ImpersonateAgent*) you registered with Access Manager.



See Also:

[Managing Policies to Protect Resources and Enable SSO](#)

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. Click **Application Domains** in the **Access Manager** section.
3. Search for and open the OWA2010 Application Domain (the relevant application domain for impersonation).

Navigate as follows:

Authorization Policies
Protected Resource Policy
Responses

4. Click the **Add** button, then **Add Response**.

Complete the form as follows:

- From the **Type** list, choose **Header**.
- In the **Name** field, type a unique name for this response. For example, `IMPERSONATE`.
- In the **Value** field, type a value for this response. For example, `$user.userid`.

5. Click **Add**, then click **Apply** to submit the changes.
6. Go to the next section, "[Adding an Impersonation DLL to IIS.](#)"

This Response is used for the second Webgate request (for authorization).

50.4.6 Adding an Impersonation dll to IIS

You are ready to configure IIS by adding the `IISImpersonationModule.dll` to your IIS configuration.

You also need to set Enable Anonymous Access because this is required for impersonation of a user.

1. Select **Start, Administrative Tools, Internet Information Services (IIS) Manager**.
2. In the left pane of IIS 7.x, click the hostname.
3. In the middle pane, under the **IIS** header, double click on **Modules**.
4. In the right pane, click **Configure Native Modules** and click **Register**.
5. In the window, provide a module **Name** (for example, *Oracle Impersonation Module*).
6. In the **Path** field, type the full path to `IISImpersonationModule.dll`.

By default, the path is:

```
Webgate_install_dir\webgate\iis\lib\IISImpersonationModule.dll
```

Where `Webgate_install_dir` is the directory of your WebGate installation.

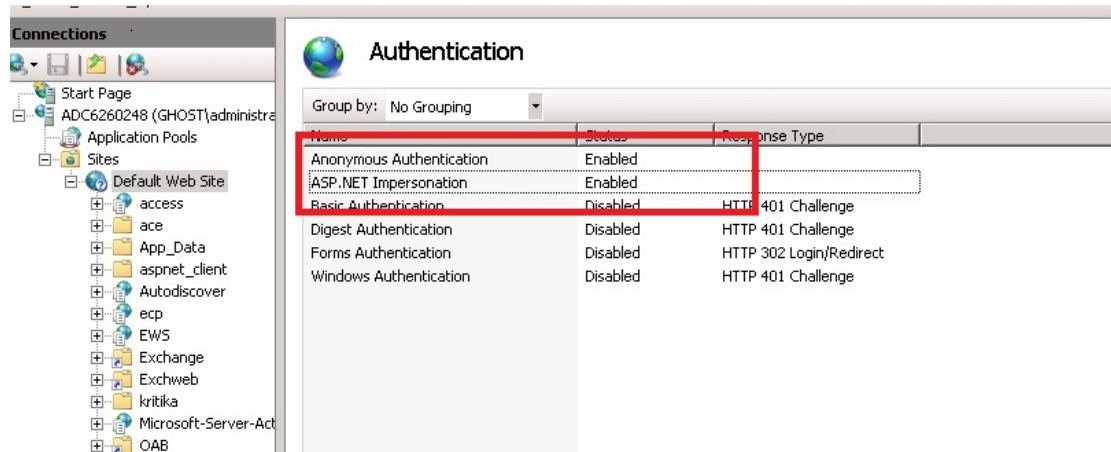
Note:

If any spaces exist in the path (for example, `C:\Program Files\Oracle\...`) surround the entire string with double quotes (" ").

7. Click **OK** to register the module.
8. Check the name of the newly created module and click **OK** to apply the module across the Web sites.

50.4.7 Configuring IIS Security

Be sure to configure IIS Security before you continue. [Figure 50-4](#) shows an example.

Figure 50-4 Impersonation Authentication

1. Select **Start, Administrative Tools, Internet Information Services (IIS) Manager**.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Click **Web Sites** on the tree in the left pane.
4. In the center pane, double-click **Authentication** under IIS.
5. Ensure that **Anonymous Authentication** is enabled and **Windows Authentication** is disabled.

50.4.8 Testing Impersonation for Outlook Web Application

The following options are provided to test the Impersonation configuration for OWA.

- [Testing Impersonation Using the Event Viewer](#)
- [Testing Impersonation using a Web Page](#)

50.4.8.1 Testing Impersonation Using the Event Viewer

You can test impersonation through the Event Viewer.

To test:

1. Select **Start Menu; Event Viewer**.
2. In the left pane, right-click **Security**, then click **Properties**.
3. Click the **Filter** tab on the **Security** property sheet.
4. Verify that all Event Types are checked, and the Event Source and Category lists are set to All, then click **OK** to dismiss the property sheet.
5. Your Event Viewer is now configured to display information about the headerVar associated with a resource request.
6. Create a new IIS virtual server (virtual site).
7. Place a target Web page anywhere in the tree on the virtual site.
8. From your browser, enter the URI to the Web page.

If impersonation is working correctly, the Event Viewer will report the success of the access attempt.

50.4.8.2 Testing Impersonation using a Web Page

You can test impersonation using a dynamic test page that can return and display information about the request.

To test:

1. Create a `.asp` page or Perl script that will display the parameters `AUTH_USER` and `IMPERSONATE`.

It can resemble this sample page:

```
<TABLE border=1>
<TR>
<TD>Variable</TD>
<TD>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</TD>
<TD>Value</TD></TR>
<%for each servervar in request.servervariables%>
<TR>
<TD><%=servervar%></TD>
<TD>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</TD>
<TD><%=request.servervariables(servervar)%>&nbsp;&nbsp;&nbsp;</TD>
</TR>
```

2. Create an IIS virtual site, or use the one you created for the previous task.
3. Place the `.asp` page or Perl script (such as the sample in the preceding listing) anywhere in the tree of the new virtual site.
4. Point your browser at the page, which should appear, with both `AUTH_USER` and `IMPERSONATE` set to the name of the user making the request.

50.4.8.3 Conducting Negative Testing for Impersonation

You can conduct negative testing for impersonation by unbinding the trusted user from the WebGate.

To conduct:

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the **Launch Pad** tab, click **Agents**.
3. Search for the desired WebGate and open it for editing.
4. In the WebGate registration page, remove the credentials for the trusted user.
5. Click **Apply** to save the change.
6. Restart the IIS server and in a browser window, go to a protected code page (previously accessible to the trusted user).
7. Confirm that you receive a message page. Values for `AUTH_USER` and `IMPERSONATE` are necessary for impersonation credentials to be bound to a Webgate.
8. Restore the trusted user to the WebGate registration page.

50.5 Setting Up Access Manager WNA for Outlook Web Application

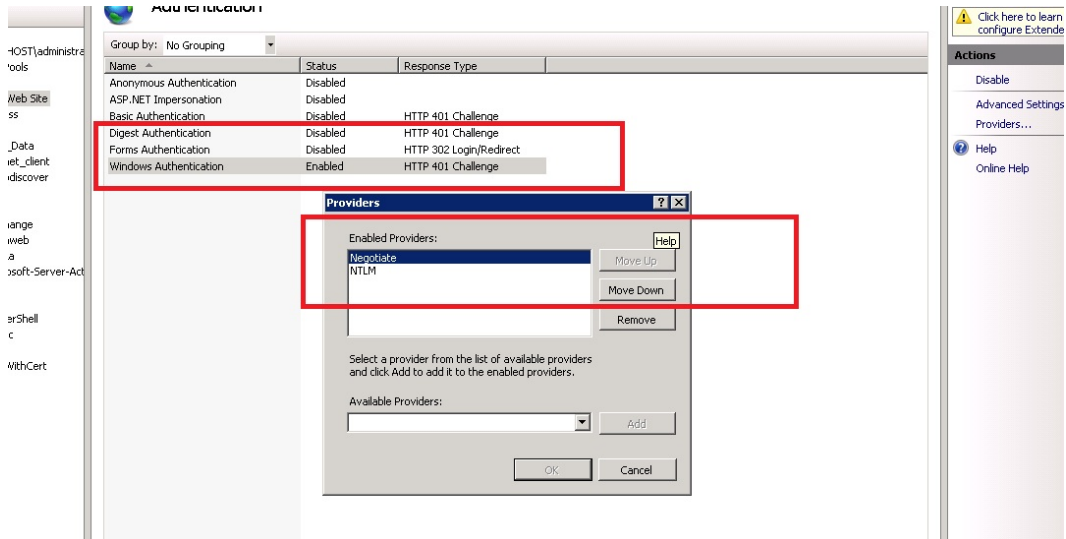
Access Manager 12c can operate with Windows Native Authentication (WNA).

This section describes setting up Access Manager with Windows Native Authentication (WNA) for Outlook Web Application (OWA).

Enabling WNA for the IIS Site front-ending OWA is described in the following procedure. It presumes a fully-configured Microsoft Active Directory authentication service is set up with user accounts to map Kerberos services, Service Principal Names (SPNs) for those accounts, and key tab files. For more information, see *Oracle Fusion Middleware Securing Oracle WebLogic Server 11g Release 1 (10.3.3) E13707-03*.

You need to configure Access Manager to use Windows Native Authentication (WNA), as described in [Configuring Access Manager for Windows Native Authentication](#).

1. Perform all prerequisite tasks.
2. Open IIS Authentication (OWA on the front-ending Site).
3. Enable **Windows authentication**.
4. Click on **Provider**.
5. Add **Negotiate** to **Provider** and move it to the top of the list.



6. Create a policy to protect OWA in IIS, as described in the *Administering Oracle Access Management*.

51

Integrating Access Manager with SAP NetWeaver Enterprise Portal

This chapter describes the integration of Access Manager 11.1.2 with SAP NetWeaver Enterprise Portal.

This chapter covers the following topics:

- [What is Supported in This Release?](#)
- [Supported Versions and Platforms](#)
- [Integration Architecture](#)
- [Configuring Oracle Access Management and NetWeaver Enterprise Portal 7.0.x](#)
- [Configuring Oracle Access Management and NetWeaver Enterprise Portal 7.4.x](#)
- [Testing the Integration](#)
- [Troubleshooting the Integration](#)

51.1 What is Supported in This Release?

Versions 7.0.x and 7.4.x of SAP NetWeaver Enterprise Portal are supported in this release.

Access Manager supports SAP NetWeaver Enterprise Portal v7.4.x with the following caveats:

- Apache 2.2.x and 2.0.x (from Apache.org) are supported Web servers with this release.
- MySAP is not certified.

Access Manager supports SAP NetWeaver Enterprise Portal v6.0 and v7.0.x with the following caveats:

- Apache 2.0 (from Apache.org) is supported as a Web server with this release.
- MySAP is not certified.

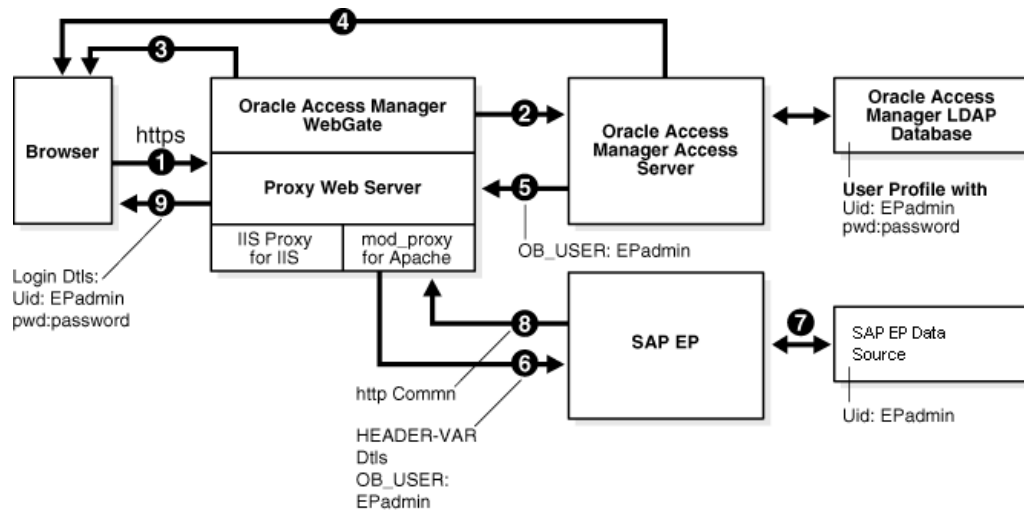
51.2 Supported Versions and Platforms

Access Manager supports the versions and platforms described on the following site:

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

51.3 Integration Architecture

The following diagram illustrates the integration between Access Manager and SAP NetWeaver Enterprise Portal.



51.3.1 Process Overview: Integration with SAP NetWeaver Enterprise Portal

Here is an overview of the integration process with SAP NetWeaver Enterprise Portal.

1. A user attempts to access content via the SAP NetWeaver Enterprise Portal.
For example, the user may enter the following URL to access an HR application through a proxy server:
`https://host:port/irj`
2. The WebGate intercepts the request and queries the Access Server for the security policy that determines if the resource is protected.
The security policy consists of an authentication scheme, authorization rules, and allowed operations. Based on the authentication and authorization success or failure, specified actions are performed.
The Access System security policy for the SAP /irj login URL is applicable to all resources accessed using the `https://host:port/irj` URL.
Note that the SAP NetWeaver Enterprise Portal has its own authorization system that can be configured to set user access to iViews.
3. If the resource is protected, the WebGate prompts the user for authentication credentials.
The credentials that the WebGate requests depend on the authentication scheme configured in the Access System, for example, Basic over LDAP or Form-based authentication.
4. If the credentials are validated, the Access System authenticates the user and sets an encrypted ObSSOCookie in the user's browser.
5. After authenticating, the authorization rules defined in the Access System are applied based on the security policy.
Specific actions are performed based on the authorization rules. If the user is authorized, access to the SAP Portal login (the requested content) is allowed. For

SAP Enterprise Portal header variable integration, the Access Server sets the authenticated user ID in a header variable.

If the user is not authenticated or authorized, he or she is denied access and redirected to another URL, as determined by the administrator. For example, the user may be redirected to an "invalid credentials" page.

6. For the integration with SAP NetWeaver Enterprise Portal, the proxy Web server redirects the request to the SAP NetWeaver Enterprise Portal internal Web server that contains the header variable details.
7. The SAP NetWeaver Enterprise Portal uses the header variable value to check the mapping of the user ID against the configured data source in the portal.

Both the Access Manager and SAP NetWeaver Enterprise Portal data source must contain the same user ID value.

Upon successful mapping, SAP NetWeaver Enterprise Portal allows the user to access the requested resource.

SAP NetWeaver Enterprise Portal sends a response to the proxy, and the proxy redirects to the client browser.

8. All interaction with the SAP Enterprise Portal takes place through the proxy server.

51.4 Configuring Oracle Access Management and NetWeaver Enterprise Portal 7.0.x

You can configure Access Manager and SAP NetWeaver Enterprise Portal 7.0.x to work together.

This section contains the following tasks:

- [Before You Begin Configuring OAM and NetWeaver Enterprise Portal 7.0.x](#)
- [Configuring the Apache HTTP Server as a Proxy](#)
- [Configuring SAP NetWeaver Enterprise Portal for External Authentication](#)
- [Adjusting the Login Module Stacks for using Header Variables](#)
- [Configuring Access Manager for SAP Enterprise Portal](#)

51.4.1 Before You Begin Configuring OAM and NetWeaver Enterprise Portal 7.0.x

- Install SAP NetWeaver Enterprise Portal before completing the steps in this section.
- Install the Apache HTTP Server by following the installation steps provided by apache.org.
- Install and configure a WebGate on each Apache HTTP Server instance that supports the proxy connection to the SAP Enterprise Portal instance. See *Installing Webgates for Oracle Access Manager* for details.
- Install Access Manager before completing the steps in [Configuring Access Manager for SAP Enterprise Portal](#) . See the *Installing and Configuring Oracle Identity and Access Management* for details.

- Synchronize the time on all servers where SAP NetWeaver Enterprise Portal and Access Manager components are installed.
- Ensure that the users exist in the Access Manager LDAP directory as well as on the SAP R3 system database.

The user ID in Access Manager and the SAP database must be the same or be mapped to each other. Any attribute in a user's profile can be configured as the SAP ID and passed directly to SAP. Alternatively, SAP can be configured to map the SAP ID to any user attribute that it receives from Access Manager.

- Verify that the Web browser is configured to allow cookies.

 **Note:**

Oracle suggests reviewing the following topics prior to integrating Access Manager with SAP NetWeaver Enterprise Portal.

- [Managing Data Sources](#) to understand how to add and configure data sources in Access Manager.
- [Managing Authentication and Shared Policy Components](#) to understand how to configure Form and Basic mode authentication in Access Manager.
- [Configuring Cert Mode Communication for Access Manager](#) to understand how to configure Cert mode for Access Manager.

51.4.2 Configuring the Apache HTTP Server as a Proxy

You can configure a proxy (Apache HTTP Server 2.0.x) to access SAP NetWeaver Enterprise Portal.

To configure Apache HTTP Server 2.0.x

1. Set up the Apache HTTP Server proxy in non-SSL mode or SSL mode, as described in the Apache documentation.

If HTTPS communication is used with the SAP NetWeaver Enterprise Portal, use SSL mode.

2. To enable the proxy to access the SAP NetWeaver Enterprise Portal, enter the following in the `httpd.conf` configuration file:

For SAP NetWeaver Enterprise Portal 6:

```
ProxyRequests Off
ProxyPass /irj http://sap_host:port/irj
ProxyPassReverse /irj http://sap_host:port/irj
ProxyPreserveHost On
```

For SAP NetWeaver Enterprise Portal 7:

```
ProxyRequests Off
ProxyPass /webdynpro http://sap_host:port/irj
ProxyPassReverse /webdynpro http://sap_host:port/irj
ProxyPreserveHost On
```

Where *sap_host* is the name of the machine hosting the SAP NetWeaver Enterprise Portal instance and *port* is the listen port for the SAP NetWeaver Enterprise Portal instance. This set of directives specifies that all of the requests to this Web server of the form `http://apache_host:port/irj` or `https://apache_host:port/irj` are redirected to `http://sap_host:port/irj` or `https://sap_host:port/irj`.

- Restart the proxy Web server.

- Access the following URL:

Non-SSL—`http://apachehost:port/irj`

SSL—`https://apachehost:port/irj`

This request should be redirected to the SAP NetWeaver Enterprise Portal login.

- Log in using the SAP NetWeaver Enterprise Portal administrator login ID.

The administrator should be able to perform the available administrative functions.

- Log in as a non-administrative user.

This user should be able to perform non-administrative functions.

51.4.3 Configuring SAP NetWeaver Enterprise Portal for External Authentication

You can enable external authentication in SAP Enterprise Portal using the `OB_USER` header variable.

For more information about configuring authentication schemes for SAP Enterprise Portal, see the SAP documentation.

To configure the header variable

- Stop the SAP J2EE dispatcher and server.

- Browse to the following directory:

`SAP_J2EE_engine_install_dir\ume`

- Back up the file `authschemes.xml.bak` to another directory.

- Rename `authschemes.xml.bak` to `authschemes.xml`.

- Open `authschemes.xml` in an editor and change the reference of the default authentication scheme to the authentication scheme header as follows:

```
<authscheme-refs>
  <authscheme-ref name="default">
    <authscheme>header</authscheme>
    <authscheme>uidpwdlogon</authscheme>
  </authscheme-ref>
</authscheme-refs>
```

- In the authentication scheme header of `authschemes.xml`, specify the name of the HTTP header variable where the Access System provides the user ID.

As described in "[Configuring Access Manager for SAP Enterprise Portal](#)", this is the `OB_USER` header variable. You configure this header variable as follows:

```
<authscheme name="header">
  <loginmodule>
    <loginModuleName>
```

```

        com.sap.security.core.logon.imp.HeaderVariableLoginModule
    </loginModuleName>
        <controlFlag>REQUISITE</controlFlag>
        <options>Header=OB_USER</options>
    </loginmodule>
    <priority>5</priority>
    <frontEndType>2</frontEndType>
    <frontEndTarget>com.sap.portal.runtime.logon.header</frontEndTarget>
</authscheme>

```

The control flag value `REQUISITE` means the login module must succeed. If login succeeds, authentication continues through the list of login modules. If it fails, control immediately returns to the application and authentication does not continue through the list of login modules.

- Restart the portal server and J2EE engine.

The modified `authschemes.xml` file will be loaded into the Portal Content Directory (PCD). SAP Enterprise Portal will rename it as `authschemes.xml.bak`.

To Configure Logout

- To enable logout from a single sign-on session in both SAP Enterprise Portal and Access Manager, configure a logout URL in SAP Enterprise Portal from the administration interface.

The URL for the administration interface is as follows:

```
http://SAP_host:port/irj/
```

Where `SAP_host` is the name of the machine hosting the SAP Enterprise Portal and `port` is the listen port for the portal.

- From the administration interface, click System Administration, then System Configuration, then UM Configuration, then Direct Editing.
- Add the following lines to the end of the configuration file:

```

ume.logoff.redirect.url=http(s)://proxy_host:port/logout.html
ume.logoff.redirect.silent=false

```

Where `http(s)` is either `http` or `https`, `proxy_host` is the name of the proxy Web server, and `port` is the listen port for the proxy.

- Save the changes and log out.

51.4.4 Adjusting the Login Module Stacks for using Header Variables

Add the `HeaderVariableLoginModule` to the appropriate login module stack or template and configure the options.

Table 51-1 Login Module Stacks for using Header Variables

Login Modules	Flag	Options
EvaluateTicketLoginModule	SUFFICIENT	{ume.configuration.active=true}
HeaderVariableLoginModule	OPTIONAL	{ume.configuration.active=true, Header=<header_name>}

Table 51-1 (Cont.) Login Module Stacks for using Header Variables

Login Modules	Flag	Options
CreateTicketLoginModule	SUFFICIENT	{ume.configuration.active=true}
BasicPasswordLoginModule	REQUISITE	{}
CreateTicketLoginModule	OPTIONAL	{ume.configuration.active=true}

To adjust the Login Module Stacks for using Header Variables

1. Run the Visual Administrator tool, in the following location:
SAPJ2EEEngine_install_dir\j2ee\admin\go.bat
2. In the Visual Administrator, choose **Security Provider**.
3. Switch to edit mode by choosing the pencil icon.
4. Choose **Policy Configurations**, then **Authentication**.
5. For each template or application that is to support header variable authentication, add the login module `HeaderVariableLoginModule` to the login module stack (see [Table 51-1](#)).

51.4.5 Configuring Access Manager for SAP Enterprise Portal

You can configure the security policy in Access Manager to protect log-ins to SAP NetWeaver Enterprise Portal.

To configure Access Manager for SAP NetWeaver Enterprise Portal

1. In to the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the **Launch Pad** tab, select **Create Application Domain** from the **Create (+)** drop-down menu in the **Access Manager** section.
The Create OAM Webgate page opens.
3. Complete the form to create a WebGate for this integration. For example:
Name—*SAP_AG*
Version - OAM
Host Identifier—Apache proxy host
Auto Create Policies—Enabled (checked)
Public Resource List—Add any public Resources to this list.
Apply—Click to create the WebGate.
4. Click the **Authorization Policies** tab, then click the **Create Authorization Policy** button to open a fresh page ([Managing Policies to Protect Resources and Enable SSO](#)).
5. **Summary Tab:** Add your information to the Summary tab.
6. Click the Resources tab, click Add (+), and define the resources for the policies in this application domain as follows:

Name: SAP EP Security Policy

Type: http

Host identifiers: Enter the proxy host URL prefix: /irj.

Description: SAP EP Login URL

7. **Add Resources:** The Resource must be defined in the Application Domain before you can add the resource to a specific policy.
 - Click the **Resources** tab on the Authorization Policy page.
 - Click the **Add** button on the Resources tab.
 - Click the **Search** button.
 - Click a URL in the Results table, then click **Add Selected**.
 - Repeat these steps to add more resources.
8. Click **Apply** to save changes and close the Confirmation window.
9. **Responses:** Add policy Responses, as described in "[Adding and Managing Policy Responses for SSO](#)".
10. **Conditions:** Add authorization conditions, as described in "[Defining Authorization Policy Conditions](#)".
11. **Rules:** Add authorization rules, as described in "[Defining Authorization Policy Rules](#)".
12. Close the page when you finish.

51.5 Configuring Oracle Access Management and NetWeaver Enterprise Portal 7.4.x

This section contains the following tasks.

- [Before You Begin Configuring OAM and NetWeaver Enterprise Portal 7.4.x](#)
- [Configuring Access Manager for SAP NetWeaver Enterprise Portal 7.4.x](#)
- [Configuring Apache Web Server 2.0.x or 2.2.x](#)
- [Configuring SAP Enterprise Portal 7.4 for External Authentication](#)
- [Adjusting the Login Module Stacks for Using Header Variables](#)

51.5.1 Before You Begin Configuring OAM and NetWeaver Enterprise Portal 7.4.x

- Install SAP NetWeaver Enterprise Portal version 7.4.x before completing the steps in this section.
- Install Access Manager as described in the *Installing and Configuring Oracle Identity and Access Management*.
- Install Apache HTTP Server 2.0.x or 2.2.x by following the installation steps provided by apache.org.

- Install and configure an OAM WebGate on each Apache HTTP Server instance that supports the proxy connection to the SAP Enterprise Portal 7.4 instance. See *Installing Webgates for Oracle Access Manager* for details.
- Synchronize the time on all servers where SAP NetWeaver Enterprise Portal and Access Manager components are installed.
- Ensure that the users exist in the Access Manager LDAP directory as well as on the SAP R3 system database.

The user ID in Access Manager and the SAP database must be the same or be mapped to each other. Any attribute in a user's profile can be configured as the SAP ID and passed directly to SAP. Alternatively, SAP can be configured to map the SAP ID to any user attribute that it receives from Access Manager.

- Verify that your Web browser is configured to allow cookies.

 **Note:**

Oracle suggests reviewing the following topics prior to integrating Access Manager with SAP NetWeaver Enterprise Portal.

- [Managing Data Sources](#) to understand how to add and configure data sources in Access Manager.
- [Managing Authentication and Shared Policy Components](#) to understand how to configure Form and Basic mode authentication in Access Manager.
- [Configuring Cert Mode Communication for Access Manager](#) to understand how to configure Cert mode for Access Manager.

51.5.2 Configuring Access Manager for SAP NetWeaver Enterprise Portal 7.4.x

You can configure the Access Manager security policy that protects SAP NetWeaver Enterprise Portal log-ins.

To configure:

1. In to the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the **Launch Pad** tab, select **Create Application Domain** from the **Create (+)** drop-down menu in the **Access Manager** section.
3. Complete the form to create a WebGate for this integration. For example:

Name—Type a meaningful name, for example, `SAP_AG`. Do not include spaces in the name.

Version - select **OAM** from the drop-down menu.

Access Client Password—Enter a password to be used during the installation of the WebGate.

Security—Choose the type of communication that should occur between the WebGate and the OAM server.

Click **Apply**.

A confirmation page opens.

4. At the bottom of the confirmation page, in the **Server Lists** section, associate the WebGate with a defined Access Server.

Click **Apply**.

5. On the Launch Pad page, go to the **Access Manager** section and click **Host Identifiers**.

Click **Search**, then click the WebGate in the search results.

Configure the host identifiers using the fully qualified proxy machine name and port for the Apache proxy.

6. Click **Application Domains** and search for the application domain name that you used to create the WebGate (for example, *SAP_WG*).

Click the application domain name in the search results to open it

- a. Click the **Resources** tab and search for the resource that the WebGates should protect. Select the resource in the search results then click the **Create** button.

Complete the form and click **Apply**.

Type - HTTP

Resource URL - /irj

Protection Level - Protected

Authentication Policy - Protected Resource Policy

Authorization Policy - Protected Resource Policy

- b. Click the **Authentication Policies** tab, then click **Protected Resource Policy**.

Choose the appropriate authentication scheme from the **Authentication Scheme** drop-down that you want to configure for this particular domain. For example, for a form-based authentication policy (FAAuthScheme), enter the following:

Name - Protected Resource Policy

Authentication Scheme - FAAuthScheme

 **Note:**

Select either basic-over-LDAP or form-based authentication.

Oracle recommends that you use a form-based authentication scheme. If you use the basic authentication scheme, also set the **Challenge Redirect** field to another WebGate to ensure that the `ObSSOCookie` is set.

Click **Apply** to save your changes.

- c. Click the **Authorization Policies** tab, then click **Protected Resource Policy**.

Click the **Responses** tab and add the following:

Type - Header

Name - OAM_REMOTE_USER

Value - Same account name

The other tabs in Authorization Policies include conditions and rules:

Condition - Creates a list of users and puts them in a group.

Rule - Allows or denies access to the group of users created in the conditions tab.

Click **Apply** to save your changes.

7. If you configured a form-based authentication scheme, ensure that a `login.html` page is configured in the proxy server document root.

Also, ensure that a `logout.html` page is present on the proxy Web server document root. You can create a custom logout page using HTML, a JSP file, or a CGI protocol.

The default logout page (`logout.html`) is located here:

```
WebGate_install_dir/webgate/apache/oamssso/logout.html
```

Where:

`WebGate_install_dir` is the directory where the WebGate is installed. Ensure that the name of the logout page contains the string `logout`.

8. Ensure that the user ID that is returned by the `OAM_REMOTE_USER` header variable exists in the user management data sources for SAP Enterprise 7.4.
9. On the Launch Pad page, go to the **Access Manager** section and click **Authentication Schemes**.

Choose the authentication scheme to use. This is the scheme that you selected inside the application domain of the WebGate.

51.5.3 Configuring Apache Web Server 2.0.x or 2.2.x

You can configure a proxy to access SAP Enterprise Portal 7.4.

To configure:

1. Set up the Apache proxy in non-SSL mode or in SSL mode. Refer to the Apache documentation for details.

If HTTPS communication is used with the SAP Enterprise Portal 7.4, use SSL mode.

2. To enable the proxy to the SAP Enterprise Portal 7.4, add the following to the `httpd.conf` file:

```
ProxyRequests Off
ProxyPass /http://sap_host:port/
ProxyPassReverse / http://sap_host:port//
ProxyPreserveHost On
```

Where:

`sap_host` - The name of the machine hosting the SAP Enterprise Portal 7.4 instance

`port` - The listening port for the SAP Enterprise Portal 7.4 instance.

This set of directives specifies that all requests to the Web server that take the form `http://apache_host:port/irj` or `https://apache_host:port/irj` are redirected to `http://sap_host:port/irj` or `https://sap_host:port/irj`.

3. Uncomment the following proxy related modules:
 - `LoadModule proxy_module modules/mod_proxy.so`
 - `LoadModule proxy_http_module modules/mod_proxy_http.so`
4. Restart the proxy Web server.
5. Open a browser and access the following URL:
 - Non-SSL: `http://apachehost:port/irj`
 - SSL: `https://apachehost:port/irj`

This request should be redirected to the SAP Enterprise Portal 7.4 login ID.
6. Log in using the SAP Enterprise Portal 7.4 administrator login ID.

Verify that you can perform the provided administrative functions when logged in as an administrator.
7. Log in as a non-administrative user.

Verify that you can perform the provided non-administrative functions when logged in.

51.5.4 Configuring SAP Enterprise Portal 7.4 for External Authentication

You can enable external authentication in SAP Enterprise Portal 7.4 using the `OAM_REMOTE_USER` header variable.

See the *SAP Enterprise Portal 7.4 Enterprise Postal Security Guide* for more information about configuring authentication schemes for SAP Enterprise Portal.

1. To enable logout from a single sign-on session in both SAP Enterprise Portal 7.4 and Access Manager, use the SAP NetWeaver Administrator interface to configure a logout URL.

Set the SAP NetWeaver Portal Logoff URL (`ume.logoff.redirect.url`) to the appropriate logout URL.
2. Open the config tool by running the `configtool.bat` file, which is located here:

```
SAP_J2EE_engine_install_dir\configtool
```

Prepare to edit the configuration by switching to configuration editor mode, and choosing edit mode.
3. Edit the properties for the following workernode service:

```
com.sap.security.core.ume.service
```

Update the `ume.logoff.redirect.url` property and the `ume.logoff.redirect.silent` property with the logoff URL configured in step 1.

```
ume.logoff.redirect.url=http(s)://proxy_host:port/logout.html
ume.logoff.redirect.silent=false
```

Save your changes and close the config tool.
4. Stop the SAP J2EE dispatcher and server.

51.5.5 Adjusting the Login Module Stacks for Using Header Variables

You can use the NetWeaver Admin console to add the `HeaderVariableLoginModule` to the appropriate login module stack or template and configure the options.

In the console, choose **Configuration > Authentication and Single Sign-On**. Click **Login Modules** under the **Authentication** tab. Create the `HeaderVariableLoginModule` login module, with the display name as `HeaderVariableLoginModule` and class name as `com.sap.security.core.server.jaas.HeaderVariableLoginModule`. Choose **Component > ticket** from the **Login Module Use** tab, and add the login module `HeaderVariableLoginModule` to the login module stack for each template or application that is to support header variable authentication.

Table 51-2 Login Module Stacks for using Header Variables

Login Modules	Flag	Options
EvaluateTicketLoginModule	SUFFICIENT	{ume.configuration.active=true}
HeaderVariableLoginModule	OPTIONAL	{ume.configuration.active=true, Header=<header_name>}
CreateTicketLoginModule	SUFFICIENT	{ume.configuration.active=true}
BasicPasswordLoginModule	REQUISITE	{}
CreateTicketLoginModule	OPTIONAL	{ume.configuration.active=true}

51.6 Testing the Integration

You can validate the front-end and back-end integrations. using web browser.

Front-End Integration Test Procedure

Follow these steps to test the integration using a Web browser.

1. Open a protected URL. For example: `https://host:port/irj`
Access Manager should prompt for authentication (either form based, or basic authentication over LDAP, or Cert Mode authentication).
2. Enter the correct user credentials.
If the credentials are correct, you will be logged into the SAP NetWeaver Enterprise Portal system.

Back-End Integration Test Procedure

To use these steps, download and install a plug-in for your Web browser that displays the HTTP requests and responses that happen when your browser requests a resource. Live HTTP Headers for Firefox, or ieHTTPHeaders for Internet Explorer are two such plug-ins.

1. Open the plug-in and type a URL in your browser to request a protected resource, for example: `https://host:port/irj`

The plug-in window will be populated with the HTTP requests and responses.

2. Analyze the requests and responses and make sure that each request returns a response without errors.

Once the user is authenticated you should see some sessions and cookies set in the HTTP Header logs. The cookies that are set include the following:

- ObSSOCookie
- JSESSIONID
- OAM_ID
- OAM_REQ

When the request reaches the SAP NetWeaver Enterprise Portal, you will receive responses from the Enterprise Portal system in the header logs.

51.7 Troubleshooting the Integration

You can troubleshoot issues with this integration.

Problem: The browser has problems displaying the SAP 7.0.x administration interface through the proxy server. You may receive an "object not found" error and related JavaScript errors.

Solution: See the following SAP document for a list of supported browsers, "[SAP NetWeaver 7.0.x Product Availability Matrix](#)."

Part XIII

Appendixes

Information that is outside the scope of day-to-day administration tasks with Oracle Access Management is discussed here.

This section contains the following appendixes:

- [Integrating Oracle ADF Applications with Access Manager SSO](#)
- [Securing Communication](#)
- [Troubleshooting](#)

A

Integrating Oracle ADF Applications with Access Manager SSO

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 Access Manager SSO security provider for user authentication, as described in the following topics:

- [Introducing Oracle Platform Security Services and Oracle Application Developer Framework](#)
- [Integrating Access Manager With Web Applications Using Oracle ADF Security and the OPSS SSO Framework](#)
- [Configuring Centralized Logout for Oracle ADF-Coded Applications](#)
- [Confirming Application-Driven Authentication During Runtime](#)

A.1 Introducing Oracle Platform Security Services and Oracle Application Developer Framework

Oracle Application Development Framework provides the ADF Security framework. ADF Security is built on top of the Oracle Platform Security Services (OPSS) architecture, which in turn incorporates the Java Authentication and Authorization Service (JAAS) and Java EE container-managed security.

You need to familiarize yourself with the following topics:

- [Oracle Platform Security Services Single Sign-on Framework](#)
- [Oracle Application Developer Framework](#)

A.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](#).

The 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; 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.

 **See Also:**

OPSS Architecture Overview in the *Securing Applications with Oracle Platform Security Services*.

A.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.

For information about upgrading the environments to Oracle Fusion Middleware 11g, refer to the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

A.2 Integrating Access Manager With Web Applications Using Oracle ADF Security and the OPSS SSO Framework

Integrate a Web application that uses Oracle ADF security and the OPSS SSO Framework with an Access Manager 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 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:

```
$MW_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)
```

Run the `addOAMSSOProvider` command as in the following example.

```
cd $MW_HOME/oracle_common/common/bin
```

```
./wlst.sh
```

```

.....after running ./wlst.sh.....
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands

addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
  logouturi="/oamssso/logout.html", autologinuri="/obrar.cgi")
addOAMSSOProvider(loginuri="/testapp/adfAuthentication",
  logouturi="/oamssso/logout.html", autologinuri="/obrar.cgi")
wls:/offline> addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
logouturi="/oamssso/logout.html", autologinuri="/obrar.cgi")

```

Table A-1 defines the expected value for each argument.

Table A-1 addOAMSSOProvider Command-line Arguments

Argument	Definition
loginuri	<p>Specifies the URI of the login page</p> <p>Note: For ADF security enabled applications, <code>"/<context-root>/adfAuthentication"</code> should be provided for the 'loginuri' parameter. Here is the flow:</p> <ol style="list-style-type: none"> 1. User accesses a resource that has been protected by authorization policies in OPSS, for example. 2. If the user is not yet authenticated, ADF redirects the user to the URI configured in 'loginuri'. 3. Access Manager should have a policy to protect the value in 'loginuri': for example, <code>"/<context-root>/adfAuthentication"</code>. 4. When ADF redirects to this URI, Access Manager displays a Login Page (depending on the authentication scheme configured in Access Manager for this URI).
logouturi	<p>Specifies the URI of the logout page</p> <p>Note: For ADF security enabled applications, logouturi should be configured based on logout guidelines in Configuring Centralized Logout for Sessions Involving OAM WebGates. For the:</p> <ul style="list-style-type: none"> • OAM WebGate the value of the logouturi should be sought from the OAM WebGate Administrator.
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.

See:

- [Sample SSO Configuration for Access Manager](#)
- [SSO Provider Configuration Details](#)

See Also:

- *Understanding the WebLogic Scripting Tool*
- *WLST Command Reference for WebLogic Server*

A.2.1 Sample SSO Configuration for Access Manager

The SSO service configuration entered with the procedure described in *Securing Applications with Oracle Platform Security Services* for all tasks involving Access Manager SSO providers and an OAM Configuration Example 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 Access Manager SSO provider. Some values are merely placeholders for actual content. Your configuration should contain values for your implementation.



See Also:

["SSO Provider Configuration Details"](#)

Sample SSO Configuration for Access Manager

```
<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="/${app.context}/adfAuthentication?
level=DIGEST" />
    <property name="autologin.url" value="/obrar.cgi" />
    <property name="logout.url" value="/${oamssso/logout.html" />
  </propertySet>

  <propertySet name = "props.auth.level">
    <property name = "level.anonymous" value = "0"/>
  </propertySet>
</propertySets>
```

```

    <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>

```

A.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 A–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 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 A–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 A–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>
```

A.3 Configuring Centralized Logout for Oracle ADF-Coded Applications

The 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 Access Manager, redirect to the `/oamssso/logout.html` resource.



See Also:

Securing Applications with Oracle Platform Security Services

 **Note:**

For ADF applications, only one extra configuration step is needed (to configure the OAMSSOProvider for OPSS).

Task overview: Protecting ADF-coded applications with Access Manager

1. Protect the ADF-coded application using OAM Webgate.
2. Perform the single extra configuration step for ADF-coded applications: configure the OAMSSOProvider.
[See Configuring Centralized Logout for ADF-Coded Applications with Access Manager.](#)
3. Perform logout configuration steps for your chosen Webgate version.

A.3.1 Configuring Centralized Logout for ADF-Coded Applications with Access Manager

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:**

Customization Commands in *WLST Command Reference for Oracle WebLogic Server*.

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="/oamssso/logout.html", autologinuri="/obrar.cgi")
```

Here, `loginuri=/${app.context}/adfAuthentication`; `logouturi` is the URI of the logout script `/logout.html`. The `logouturi` 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 Webgate:** Perform steps in "[Configuring Centralized Logout for OAM WebGates](#)".

A.4 Confirming Application-Driven Authentication During Runtime

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 Access Manager SSO.

To confirm application-driven authentication during run time

1. Create the application based on the Oracle ADF framework.
2. Configure the Access Manager SSO Security provider, as described in "[Integrating Access Manager With Web Applications Using Oracle ADF Security and the OPSS SSO Framework](#)".
3. Access the protected field and confirm that the application triggers authentication.

B

Securing Communication

Ensure that the OAM Servers and clients (OAM Agents) can communicate securely across the Access Protocol channel.

You need to perform the following tasks to secure communication:

- [Prerequisites to Setting up a Secure Communication between OAM Servers and Webgates](#)
- [Securing Communication Between OAM Servers and WebGates](#)
- [Securing Communication between OAM Servers and WebGates using OAP over REST](#)
- [Generating Client Keystores for OAM Tester in Cert Mode](#)
- [Configuring Cert Mode Communication for Access Manager](#)
- [Configuring Simple Mode Communication with Access Manager](#)

B.1 Prerequisites to Setting up a Secure Communication between OAM Servers and Webgates

Before you proceed with setting up a secure Communication between OAM servers and webgates ensure the system level requirements are met.

Following are the requirements to perform tasks in this chapter:

- If OAM Server mode is CERT mode, agents must use CERT mode.
- During agent registration, at least one OAM Server instance must be running in the same mode as the agent. After agent registration, you can change the mode of the OAM Server.

See Also:

- [About Communication Between OAM Servers and WebGates.](#)
- For details about the SSL automation tool, managing ports for WebLogic Server, Oracle HTTP Server, and Oracle Fusion Middleware.

See *Administering Oracle Fusion Middleware*.

B.2 Securing Communication Between OAM 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 Servers and WebGates use the same certificates, issued and signed by Oracle CA.
 See [About Simple Mode, Encryption, and Keys](#).
- **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 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. 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.
 See [About Cert Mode Encryption and Files](#).
- **HTTP:** User defined un-encrypted communication mode.
 If the user defined parameter `OAMServerCommunicationMode` is set to HTTP, then the webgate will communicate with the OAM managed servers using HTTP protocols.
- **HTTPS:** User defined encrypted communication mode through the Secure Sockets Layer (SSL) protocol.
 If the user defined parameter `OAMServerCommunicationMode` is set to HTTPS, then the webgate will communicate with the OAM managed servers using HTTPS protocols.
- **OAP:** If the user defined parameter `OAMServerCommunicationMode` is set to OAP then WebGate communicates with the OAM managed servers using the legacy back channel Protocols, OAP over the TCP port, utilizing the communication mode of Open, Simple, or Cert.



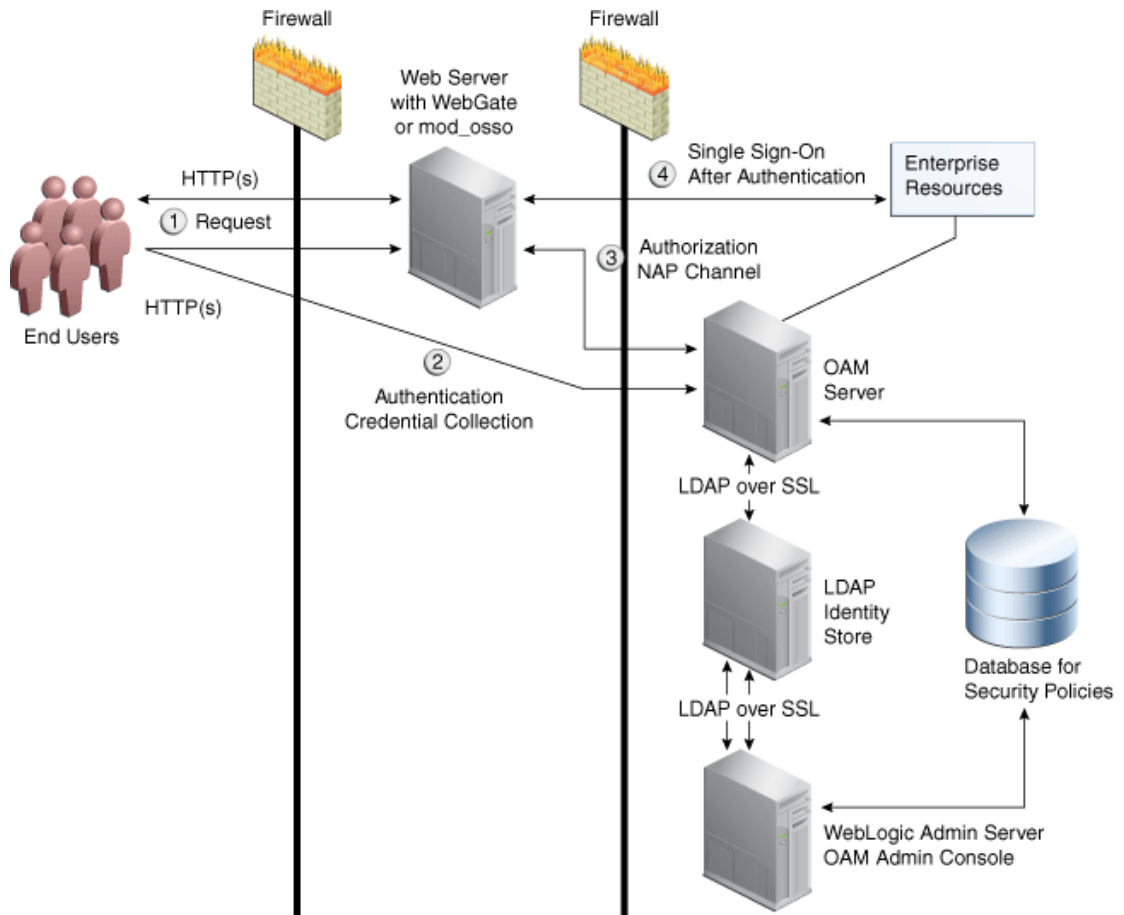
See Also:

[About Certificates, Authorities, and Encryption Keys](#)

Logically the request is to the Access Manager credential collector. However, when you have a Web server proxy in front of the WebLogic AdminServer, with a `<LocationMatch "/*>`, all requests are routed through the proxy. In this case, there is perimeter defense using the proxy.

Figure B-1 illustrates the communication channels used by OAM Servers and WebGates during user authentication and authorization.

Figure B-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.

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. Access Manager components use X.509 digital certificates in DER (binary form of a certificate) format only.

See:

- [About Certificates, Authorities, and Encryption Keys](#)
- [About Security Modes and X509Scheme Authentication](#)
- [The Importcert Tool](#)
- [TLS 1.3 and TLS 1.2 Support in Oracle Access Management](#)

B.2.1 About Certificates, Authorities, and Encryption Keys

Digital certificates can be stored in a registry from which authenticating users can look up the public keys of other users.

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

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](#)
- [About Simple Mode, Encryption, and Keys](#)

B.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://managed_server_host:managed_server_ssl_port/oam/CredCollectServlet/X509`



Note:

If a relative Challenge URL is specified with X509Scheme, the OAM Server uses the specified Server *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"](#)

B.2.3 The Importcert Tool

Administrators use the Oracle-provided `importcert` tool for several different procedures related to keystores, keys, and certificates.

[Table B-1](#) provides the syntax for `importcert` commands.

Table B-1 importcert Command Syntax

Option	Description
keystore	Follow this command with the path to an existing (or new) keystore. For example: <code>/scratch/.oamkeystore</code> or <code>/scratch/clientKey.jks</code>
privatekeyfile	Follow this option with the path to your private key. For example: <code>/scratch/aaa_key.der</code>
signedcertfile	Follow this option with the path to your signed certificate. For example: <code>/scratch/aaa_cert.der</code>
alias	Follow this option with your keystore entry alias. Required with <code>genkeystore</code> .: <code>alias</code>

Table B-1 (Cont.) importcert Command Syntax

Option	Description
storetype	<p>Follow this option with your keystore type. By default, the store type is JCEKS (OAM Server keystore). For example:</p> <p>Server keystore .oamkeystore, of type:</p> <p>JCEKS</p> <p>Client keystore/scratch/clientTrustStore.jks and /scratch/clientKey.jks can be used. Both are type:</p> <p>JKS</p>
genkeystore	<p>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.</p> <p>Specify:</p> <p>Yes or No</p> <p>Yes imports the certificates in a new keystore. No imports certificates into an existing keystore.</p>
Sample for OAM Server	<pre>- java -cp importcert.jar oracle.security.am.common.tools.importcerts.CertificateImport rt -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></pre> <p>Enter the keystore password and alias password when prompted.</p>
Sample for OAM Client See Also Generating Client Keystores for OAM Tester in Cert Mode	<pre>- java -cp importcert.jar oracle.security.am.common.tools.importcerts.CertificateImport rt -keystore <path to clientkey.JKS> -privatekeyfile <path to aaa_key.der> -signedcertfile <path to aaa_cert.der> - storetype <JKS> genkeystore <yes></pre> <p>Enter the keystore password when prompted.</p>

B.2.4 TLS 1.3 and TLS 1.2 Support in Oracle Access Management

Transport Layer Security (TLS) 1.3 and TLS 1.2 are supported with OAM 12c to provide communications security over the Internet. This protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

OAM supports TLS 1.3 across the Front channel only. It supports WebGate communication with the OAM managed servers using OAP over HTTP.

The following communication modes are not supported:

- OAP over TCP
- OHS
- OAM back-channel communication with the Oracle Database

- OAM back-channel communication with LDAP Identity Stores

OAM supports TLS 1.2 across the following channels:

Channel	TLS 1.2 Status
Front	TLS 1.2 is fully supported since incoming traffic is terminated on the Load Balancer, Web Server or Weblogic Server.
OAP Back	The 12c Webgates fully support TLS 1.2.
LDAP Back	TLS 1.2 transport is supported with 11.1.2.3.0 BP8. However, TLS 1.2 is not supported when using an IDS Profile based User Identity Store.
JDBC Back	Databases are abstracted using WLS Datasources, which can be configured to use TLS 1.2 to connect to the database. OES uses JDBC as the Database abstraction and can be configured to use TLS 1.2.
Outbound HTTPS	All outbound calls are done using JSSE and rely on the JDK specific defaults. Starting with JDK 6 121 Update and JDK 7 Update 95, you can control the platform TLS protocols by setting the system property <code>jdk.tls.client.protocols</code> .

TLS 1.2 supports cipher suites that gets installed with JDK.

In a freshly installed environment,

- Consider 12c PS3 OAM server and 12c PS3 webgates, employing latest versions of OAM server and webgate. SSL communication between them will use TLSv1.2 protocol and SHA-2 certificates.
- Consider 12c PS3 OAM server and 12CPS2/ R2PS3 webgates, employing older versions of webgate and OAM server. OAM can support TLSv1.2 and SHA-2 certificates but 12CPS2/ R2PS3 webgates cannot.

Since we have latest 12c OAM server combined with older versions of webgate/OAM servers where TLS1.2 and SHA2 is not supported, we need to relax the security posture in OAM by updating the `java.security` file with the following changes:

- Open the `java.security` file located at `JAVA_HOME/jre/lib/security/` in an editor.
- Remove TLSv1, TLSv1.1, MD5withRSA from the following key:
key - `jdk.tls.disabledAlgorithms`
- Remove MD5 from the following key:
key - `jdk.certpath.disabledAlgorithms`

Steps to enable TLSv1.2 in OAM 12c environment

Enable TLSv1.2 in OHS:

Update the below two files in 12c OHS:

```
/scratch/work/TLStest/OHS12c/Oracle_Home/user_projects/domains/base_domain/
config/fmwconfig/components/OHS/ohs1/ssl.conf
```

```
/scratch/work/TLStest/OHS12c/Oracle_Home/user_projects/domains/  
base_domain/config/fmwconfig/components/OHS/instances/ohs1/ssl.conf
```

```
SSLProtocol -ALL +TLSv1.2
```

Enable TLSv1.2 in DB:

Follow the below steps to enable in DB:

- **Create a wallet for server jnetadmin_s and one for client jnetadmin_c**

```
mkdir root

mkdir jnetadmin_s

mkdir jnetadmin_c

export ORACLE_HOME=/scratch/alice/work/db1427

export T_WORK=$ORACLE_HOME

export ROOT_CERT_DIR=$T_WORK/root

export TNS_ADMIN_SERVER=$T_WORK/jnetadmin_s

export TNS_ADMIN_CLIENT=$T_WORK/jnetadmin_c

./orapki wallet create -wallet $ROOT_CERT_DIR -pwd welcome1

./orapki wallet add -wallet $ROOT_CERT_DIR -dn "CN=Root,C=US" -  
keysize 2048 -self_signed -validity 3650 -pwd welcome1

./orapki wallet export -wallet $ROOT_CERT_DIR -dn "CN=Root,C=US" -  
cert $ROOT_CERT_DIR/b64certificate.txt -pwd welcome1

./orapki wallet create -wallet $TNS_ADMIN_SERVER -auto_login -pwd  
welcome1

./orapki wallet add -wallet $TNS_ADMIN_SERVER -dn  
"CN=Server,OU=ST,O=Oracle,ST=California,C=US" -keysize 2048 -pwd  
welcome1

./orapki wallet export -wallet $TNS_ADMIN_SERVER -dn  
"CN=Server,OU=ST,O=Oracle,ST=California,C=US" -  
request $TNS_ADMIN_SERVER/creq.txt -pwd welcome1

./orapki cert create -wallet $ROOT_CERT_DIR -  
request $TNS_ADMIN_SERVER/creq.txt -cert $TNS_ADMIN_SERVER/cert.txt  
-validity 3650 -pwd welcome1

./orapki wallet add -wallet $TNS_ADMIN_SERVER -trusted_cert -  
cert $ROOT_CERT_DIR/b64certificate.txt -pwd welcome1
```

```

./orapki wallet add -wallet $TNS_ADMIN_SERVER -user_cert -
cert $TNS_ADMIN_SERVER/cert.txt -pwd welcome1

./orapki wallet create -wallet $TNS_ADMIN_CLIENT -auto_login -pwd welcome1

./orapki wallet add -wallet $TNS_ADMIN_CLIENT -dn
"CN=Client,OU=ST,O=Oracle,ST=CA,C=US" -keysize 2048 -pwd welcome1

./orapki wallet export -wallet $TNS_ADMIN_CLIENT -dn
"CN=Client,OU=ST,O=Oracle,ST=CA,C=US" -request $TNS_ADMIN_CLIENT/creq.txt
-pwd welcome1

./orapki cert create -wallet $ROOT_CERT_DIR -request $TNS_ADMIN_CLIENT/
creq.txt -cert $TNS_ADMIN_CLIENT/cert.txt -validity 3650 -pwd welcome1

./orapki wallet add -wallet $TNS_ADMIN_CLIENT -trusted_cert -
cert $ROOT_CERT_DIR/b64certificate.txt -pwd welcome1

./orapki wallet add -wallet $TNS_ADMIN_CLIENT -user_cert -
cert $TNS_ADMIN_CLIENT/cert.txt -pwd welcome1

./orapki wallet add -wallet $TNS_ADMIN_CLIENT -trusted_cert -
cert $TNS_ADMIN_SERVER/cert.txt -pwd welcome1

```

- **Create/Modify wallet.ora files**

```

cd jnetadmin_s
vim wallet.ora

```

content as below:

```

WALLET_LOCATION=

(SOURCE=

(METHOD=FILE)

(METHOD_DATA=

(DIRECTORY=/scratch/alice/work/db1427/jnetadmin_s/)

)

)

```

- **Modify sqlnet.ora**

```

SSL_VERSION=1.2

ssl_client_authentication=false

ifile=/scratch/alice/work/db1427/jnetadmin_s/wallet.ora

```

- **Modify listener.ora**

```

LISTENER =

  (DESCRIPTION_LIST =

    (DESCRIPTION =

      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))

      (ADDRESS = (PROTOCOL = TCP) (HOST = slc03rvu.us.oracle.com)
(PORT = 1521))

      (ADDRESS=(PROTOCOL=tcps) (HOST=slc03rvu.us.oracle.com)
(PORT=5551))

    )

  )

ADR_BASE_LISTENER = /scratch/work

ssl_client_authentication=false

ifile=/scratch/alice/work/db1427/jnetadmin_s/wallet.ora

SSL_VERSION=1.2

SID_LIST_listener=(SID_LIST=

  (SID_DESC=(SID_NAME=db1427) (ORACLE_HOME=/scratch/alice/work/
db1427))

  (SID_DESC=(SID_NAME=db1427) (GLOBAL_DBNAME=db1427.us.oracle.com))

)

```

- **Restart database and listener**

```

restart DB
lsnrctl stop
lsnrctl start

```

- **Use openssl to verify TLSv1.2**

```

openssl s_client -showcerts -tls1_2 -connect
slc03rvu.us.oracle.com:5551

```

Add WLS JDBC TLS Datasources:

- Import DB server cert generated in step 3 to WLS Demo Trust.jks

```
keytool -importcert -trustcacerts -alias dbroot -keystore $MW_HOME/
wlsrserver/server/lib/DemoTrust.jks -storepass DemoTrustKeyStorePassPhrase -
file $ROOT_CERT_DIR/b64certificate.txt
```

- Navigate to wls console->Service->Data Sources, update below JDBC Data Source with TLS connection details.

- Open each data source, click Connection Pool, update URL and Properties as below:

URL example:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS)
(HOST=slc03sfc.us.oracle.com) (PORT=2484)))
(CONNECT_DATA=(SERVICE_NAME=db4758..us.oracle.com)
(SEcurity=(SSL_SERVER_CERT_DN="CN=slc03sfc.us.oracle.com"))))
```

Properties example:

```
javax.net.ssl.trustStoreType=JKS

javax.net.ssl.trustStorePassword=DemoTrustKeyStorePassPhrase

javax.net.ssl.trustStore=/net/slc03sfc/scratch/work/mw4838/wlsrserver_10.3/
server/lib/DemoTrust.jks
```

- Restart the servers.

Enable TLS for OAP channel on WLS using multiple modes:

See, [Configuring Oracle HTTP Server WebGate for Oracle Access Manager to configure 12c Cert and Simple Mode](#).

For Simple mode, remove MD5 from `jdk.certpath.disabledAlgorithms` option in `java.security` file and add the following property in `$OHS_ORACLE_HOME/user_projects/domains/base_domain/config/fmwconfig/components/OHS/ohs1/ohs.plugins.nodemanager.properties` file,

```
environment.ORACLE_SSL_ALLOW_MD5_CERT_SIGNATURES=1
```

Enable TLS1.2 Communication between OAM and the User Identity Store

Follow the below steps:

- Login to OAM console.
- Create a new OAM ID Store.

The screenshot shows the 'OAMIDStore User Identity Store Service' configuration page. The 'Store Name' is 'OAMIDStore' and the 'Store Type' is 'OUD: Oracle Unified Directory'. The 'Location' is 'slc06bku.us.oracle.com:1389' and the 'Bind DN' is 'cn=oamLDAP,cn=SystemC'. The 'Login ID Attribute' is 'uid' and the 'User Password Attribute' is 'userPassword'. The 'User Search Base' is 'cn=Users,dc=us,dc=oracle,dc=com' and the 'Group Search Base' is 'cn=Groups,dc=us,dc=oracle,dc=com'. There are buttons for 'Duplicate', 'Test Connection', and 'Apply'.

- Update the Default store to the ID store you just created.

The screenshot shows the 'User Identity Stores' configuration page. Under 'Default and System Store', the 'Default Store' is set to 'OAMIDStore' and the 'System Store' is set to 'UserIdentityStore1'. There is an 'Apply' button. Below this, there is a section for 'OAM ID Stores' with a 'Sync IDS Profiles' button.

- Update LDAP Authentication Module to the ID Store you just created.

The screenshot shows the 'LDAP Authentication Module' configuration page. The 'Name' is 'LDAP' and the 'User Identity Store' is set to 'OAMIDStore'. There are buttons for 'Duplicate' and 'Apply'.

- Shutdown all servers, and import OUD's certificate to JDK key store.

```
openssl s_client -showcerts -connect slc06bku.us.oracle.com:1636
</dev/null2>/dev/null|openssl x509 -outform PEM >cert.pem
openssl x509 -outform der -in cert.pem -out cert.der
keytool -importcert -alias oud -file cert.der -keystore cacerts -
storepass changeit
```

- To support the TLSv1.2 connection to the LDAP server, add the `LDAP_SSL_PROTOCOL` parameter with value `TLSv1.2` after the `LDAP_URL` parameter in the `oam-config.xml` file.

To set the parameters in the `oam-config.xml`:

1. Export the OAM configuration file to `/tmp/oam-config.xml` using the export method. See [Updating OAM Configuration](#) for details.
2. Locate the IdentityStore section in the exported `/tmp/oam-config.xml` file and search for the following line:

```
<Setting Name="LDAP_URL" Type="xsd:string">ldaps://
myldap.example.com:1636</Setting>
```

 **Note:**

If the `LDAP_URL` does not include the `ldaps` protocol and the LDAPS port of your Directory Services Identity Store, then update it to use LDAPS. For example, update `ldap://myldap.example.com:1389` to `ldaps://myldap.example.com:1636`.

3. Add the `LDAP_SSL_PROTOCOL` parameter after the `LDAP_URL` entry:

```
<Setting Name="LDAP_SSL_PROTOCOL" Type="xsd:string">TLSv1.2</Setting>
```

The entries in the `oam-config.xml` file must look similar to the following example:

```
<Setting Name="LDAP" Type="htf:map">
...
<Setting Name="LDAP_URL" Type="xsd:string">ldaps://
myldap.example.com:1636</Setting>
<Setting Name="LDAP_SSL_PROTOCOL" Type="xsd:string">TLSv1.2</
Setting>
```

4. Import the updated configuration from `/tmp/oam-config.xml` using the import method. See [Updating OAM Configuration](#) for details.
- Adding below lines to `config/fmwconfig/servers/oam_server1/logging.xml` and `config/fmwconfig/servers/AdminServer/logging.xml`

```
<logger name='oracle.oam.user.identity.provider'
level='TRACE:32'useParentHandlers='false'>
<handler name='odl-handler' />
</logger>
```

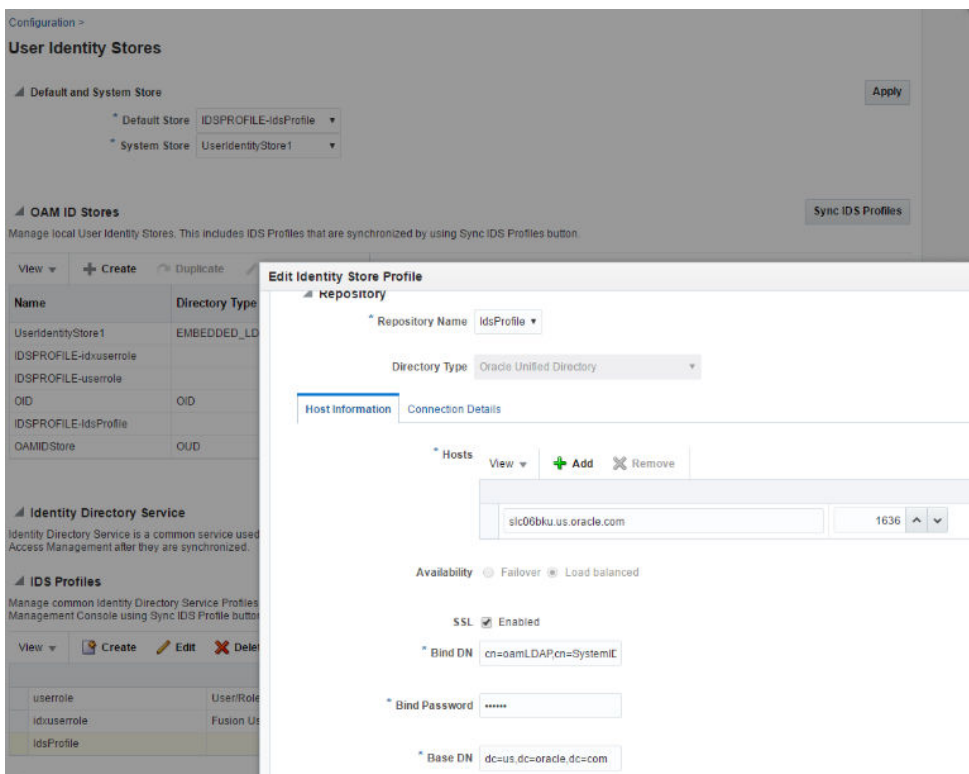
- Start servers and access protected resources, you will see `tls` logs in `oam_server1-diagnostic.log`:

```
[2016-08-09T01:46:49.398-07:00] [oam_server1] [TRACE:32] []
[oracle.oam.user.identity.provider] [tid: [ACTIVE].ExecuteThread: '0' for
queue: 'weblogic.kernel.Default (self-tuning)'] [userId: <anonymous>]
[ecid: 036b5306-7533-4458-ad54-5f5be25adadf-00000106,0] [APP: oam_server]
[partition-name: DOMAIN] [tenant-name: GLOBAL] [SRC_CLASS:
oracle.security.am.engines.common.identity.provider.impl.ids.IDSLDAPConfig
urator] [SRC_METHOD: getIDSInstance] Setting ssl protocol as TLSv1.2
```

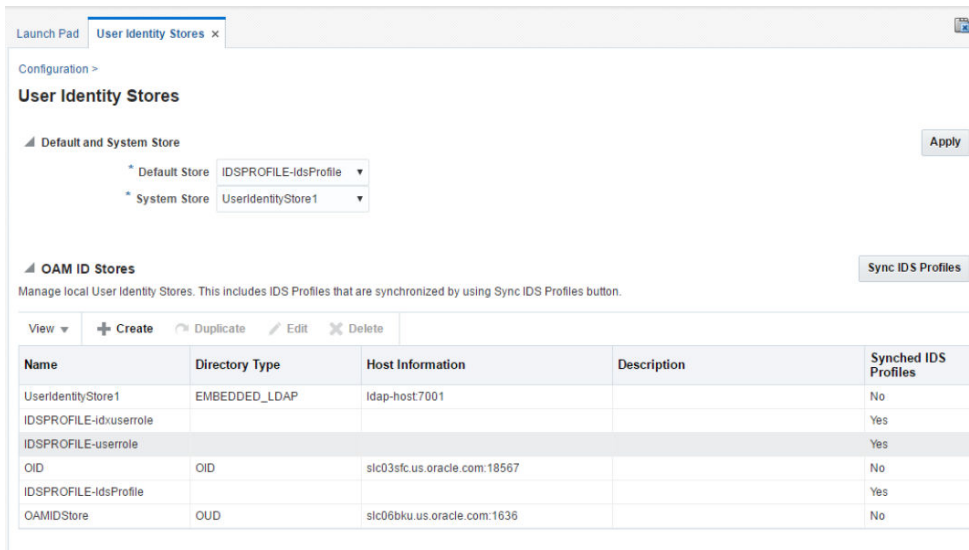
Enable TLS for OAM using IDS Profile

Follow the below steps:

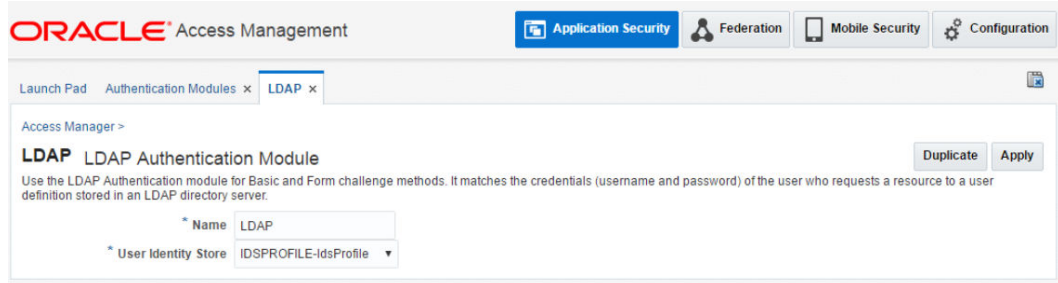
- Log in to OAM console.
- Create an IDS Profile.



- Click Sync IDS Profile and update Default store to the IDS profile you just created.



- Update LDAP Authentication Module to the ID Store you just created.



- Add TLS parameter to ids libovd using wlst,


```
modifyLDAPAdapter(adapterName='IdsProfile', attribute='Protocols',
value='TLSv1.2', contextName='ids')
```
- Create key store for ids libovd,


```
export ORACLE_HOME=/scratch/work/mw169
export WL_HOME=/scratch/work/mw169/wlserver
export JAVA_HOME=/scratch/work/view/nowang_dte8461/bootstrap/java/
1.8.0-51-16-150608.1.8.0.51.0016/jdk

./libovdconfig.sh -host slc03sfc.us.oracle.com -port 22899 -userName
weblogic -domainPath /scratch/work/mw169/user_projects/domains/WLS_IDM -
createKeystore -contextName ids
```
- Add OUD certificate to libOVD key store,


```
openssl s_client -showcerts -connect slc06bku.us.oracle.com:1636 </dev/null
2>/dev/null|openssl x509 -outform PEM >cert.pem

openssl x509 -outform der -in cert.pem -out cert.der

keytool -import -keystore adapters.jks -storepass weblogic1 -alias oud -file
cert.der
```
- Add "-Dssl.debug=true -Dweblogic.StdoutDebugEnabled=true -
 Djavax.net.debug=all" to setDomainEnv.sh and check the logs to see the TLS
 connection messages from the oam_server1 output console redirect file.
- Restart all servers, and access protected resources.



See Also:

Configuring the LDAP and LDAPS Connection Handlers and Configuring Secure Sockets Layer (SSL)

B.2.5 Generating Client Keystores for OAM Tester in Cert Mode

Generate JKS keystores to be used with OAM Tester that is in Cert mode only, else skip this topic.

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:

[The Importcert Tool](#)

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 Access Manager](#)
 - [Configuring Simple Mode Communication with Access Manager](#)
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.

B.3 Securing Communication between OAM Servers and WebGates using OAP over REST

OAP over REST uses HTTP(S) transport to ensure secure communication between OAM Servers and WebGates.

- [About OAP over REST Communication](#)
- [Configuring Load Balancer using Oracle mod_wl_proxy on OHS](#)
- [Configuring Work Manager for OAP over REST](#)
- [Configuring HTTP and HTTPS Communication between WebGate and Access Manager](#)
- [Enabling two-way SSL for OAP over REST](#)
- [Connection Tuning for OAP over REST](#)

B.3.1 About OAP over REST Communication

OAP over REST enables the HTTP(S) transport mechanism between WebGate and OAM server. This transport mechanism reduces the operational cost for both cloud and hybrid deployments, where some components are on-premises and others are moved to cloud.

OAP provides an additional layer of security by encrypting, by default the messages sent to the server using `RESTPayloadEncryption`.

To support HTTP(S) communication, OAM server uses the following:

- REST endpoint as server filter deployed on managed server `/iam/access/binding/api/v10/oap`.
- Work Manager components. See [Configuring Work Manager for OAP over REST](#)

With this 12.2.1.4.0 release of Oracle Access Management, OAP over REST is the default way of communication. Whenever you create an SSO agent, the following additional parameters are set, by default.

```
OAMRestEndPointHostName=host1.com, OAMRestEndPointPort=443, and
OAMServerCommunicationMode=HTTPS
```

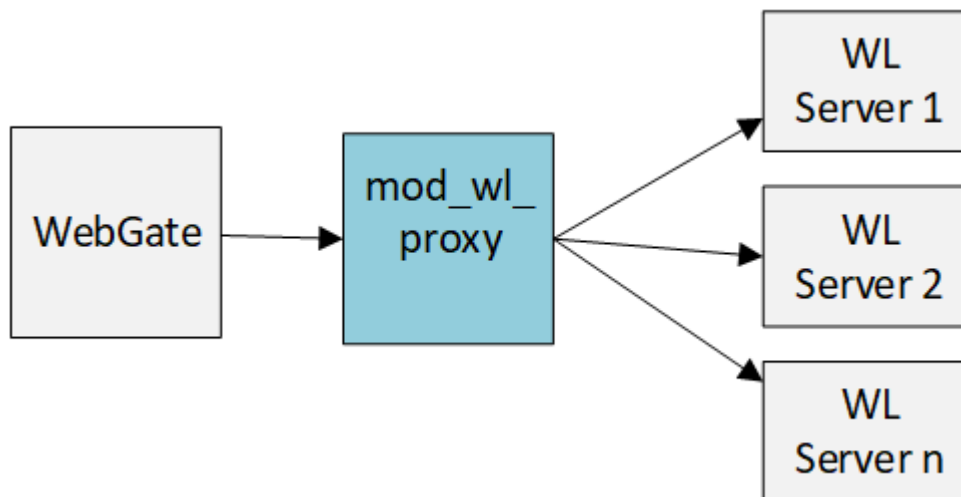
Also see, [Setting Up a Master and a Clone in Multi-Data Center](#)

B.3.2 Configuring Load Balancer using Oracle `mod_wl_proxy` on OHS

Configure `mod_wl_proxy` plug-in as a load balancer between WebGate and Managed Servers, to detect server loads and direct client requests appropriately.

For information about the `mod_wl_proxy` plugin configuration, see [Configuring the WebLogic Proxy Plug-In](#).

Figure B-2 `mod_wl_proxy` as Load Balancer



Example

Consider a deployment where `mod_wl_proxy` is configured on OHS running on SSL port 443 and on `host1.com`, and two Managed server instances running on `clst1.example.com:24100` & `clst2.example.com:24100`

The WebGate configuration looks like:

```
OAMRestEndPointHostName=host1.com
OAMRestEndPointPort=443
OAMServerCommunicationMode=HTTPS
```

The `mod_wl_proxy` configuration looks like:

```
<IfModule weblogic_module>
WebLogicCluster clst1.example.com:24100,clst2.example.com::24100
DynamicServerList ON
MatchExpression /iam/access/binding/api/v10/oap
KeepAliveSecs 90
</IfModule>
```

`mod_wl_proxy` uses `DyanamicServerList` property for load balancing, which detects the load dynamically and perform addition or removal of servers in clusters accordingly.

`mod_wl_proxy` supports the following load balancing algorithms:

- **Round-Robin** - This is the default load balancing strategy that is used when no other algorithms is specified. The round-robin algorithm cycles through a list of server instances that host the clustered servlet in order.
- **Weight-Based** - This load balancing strategy determines carefully the relative weights to assign to each server instance.
- **Random** - Here requests are routed to servers at random, the requests are evenly distributed across server instances in the cluster.
- **Affinity Based** - This algorithm is used in combination with one of the standard load balancing methods such as round-robin, weight-based, or, random.

B.3.3 Configuring Work Manager for OAP over REST

To support HTTP(S) operations, OAM server uses Work Manager `wm/OAPOverRestWM` specific for OAP over REST.

Configure Work Manager to manage work in the application using OAP over REST:

1. Login to the WebLogic Administration Console.
2. In the left pane of the console, expand **Environment** and select **Servers**.
3. Click **oam_server**.
4. In the Settings for `oam_server` page, select **Deployments** tab.
5. Expand **oam_server** and expand **Module**, select `/iam/access/binding` link.
6. Click **Configuration** tab and select **Workload** tab.
7. Select Work Manager `wm/OAPOverRestWM` and below **Application Scoped Work Manager Components**, configure the following parameters:
 - **Capacity** - Capacity includes all requests, queued or executed, from the constrained work set. Work is rejected when the capacity threshold is exceeded.

- **MaxThreadsCount** - Maximum number of concurrent threads that can execute requests sharing this constraint.
8. Click **Save**.
See, Using Work Managers to Optimize Scheduled Work

B.3.4 Configuring HTTP and HTTPS Communication between WebGate and Access Manager

This topic describes how to configure HTTP and HTTPS communication between WebGate and Access Manager.

Prerequisites:

Ensure you have performed the following steps:

1. Configured SSL for the Proxy/Loadbalancer/WebLogic. For details, see Configuring SSL for the Web Tier.
2. Created SSL Wallet and Enabled SSL for OHS. For details, see How to Enable SSL for Oracle HTTP Server by Using Fusion Middleware Control?

When the OAM server is installed and the WebGate is provisioned, by default, the parameters in the following table are configured and the values are automatically populated. The following mandatory configurations are set in the WebGate user defined parameters:

Table B-2 Mandatory Configurations in WegGate User Defined Parameters Field

Parameter Name	Description
OAMServerCommunicationMode	<p>Specifies the communication mode that has to be set between the agent and the server. WebGate will communicate with OAM Managed server using the configured protocol (HTTP/HTTPS).</p> <p>Default value: HTTP/HTTPS (Based on the value configured in Access Manager Settings for WebGate Traffic Load Balancer. For more information, see Managing WebGate Traffic Load Balancer)</p> <p>Values Allowed: HTTP, and HTTPS</p> <p>Ensure OAMServerCommunicationMode is set to HTTPS to enable one-way or two-way SSL channel for OAP over REST.</p> <p>For HTTPS, there is a default trusted certificate file bundled in WebGate at</p> <pre>WEBGATE_ORACLE_HOME/webgate/ohs/tools/curl/cacert.pem</pre> <p>If <code>aaa_chain.pem</code> is not configured then it will default to <code>cacert.pem</code>.</p> <p>You must copy the trusted certificate to</p> <pre>\$WEBGATE_INSTANCE_DIR/webgate/config/cacert.pem</pre>
OAMRestEndPointHostName	<p>Specifies the host name of the server running the service or the Load Balancer URL pointing to the REST endpoint.</p> <p>Default value: Based on the value configured in Access Manager Settings for WebGate Traffic Load Balancer.</p>
OAMRestEndPointPort	<p>Specifies the port of the server running the service.</p> <p>Default value: Based on the value configured in Access Manager Settings for WebGate Traffic Load Balancer.</p>



Note:

The following Webgate configurations are not used when the `OAMServerCommunicationMode` is set to `HTTP` or `HTTPS`:

- Security - The `Open`, `Simple`, and `Cert` modes
- Access Client Password
- Max Session Time
- Access Manager Primary and Secondary server list. The max and min connection pool size can be specified using the `MaxPoolSize` and `MinPoolSize` settings. For details, see [Connection Tuning for OAP over REST](#).

Following table provides the WebGate user defined configuration parameters specific to OAP over REST:

Table B-3 WebGate User Defined Configuration Parameters

Parameter Name	Description
<code>OAMRestEndPointUrl</code>	Specifies the URL of the service. Default value: <code>/oam/services/proxy/oapoverrest/v1</code>
<code>RESTPayloadEncryption</code>	Encrypts the OAP message using WebGate agent key. It is not used in OAP communication mode. Default value: <code>True</code> Allowed values: <code>True/False</code>
<code>IdleConnectionTimeout</code>	Idle time in seconds after which a connection is closed. Default value: <code>60 seconds</code>
<code>MinPoolSize</code>	The minimum number of connections that will be kept open. Default value: <code>2</code>
<code>MaxPoolSize</code>	The maximum number of connections that WebGate can open. Default value: <code>100</code>
<code>SSLVerifyHostname</code>	This option determines whether WebGate verifies that the server cert is for the server it is known as. Default value: <code>True</code>
<code>SSLVerifyPeerCert</code>	This option determines whether WebGate verifies the authenticity of the peer's certificate. Default value: <code>True</code>

B.3.4.1 Enabling two-way SSL for OAP over REST

Perform the following steps to enable two-way SSL channel for OAP over REST between WebGates and the OAM Server

 **Note:**

This feature is not supported for IHS WebGate on AIX platform.

1. Set the configurations as described in [Configuring HTTP and HTTPS Communication between WebGate and Access Manager](#).
2. Generate WebGate Certificate Signing Request (CSR) and get it signed by Trusted CA or Root CA.

 **Note:**

The generated certificate and private key must be in .pem format.

For example,

- a. Generate private key either with passphrase or without passphrase:

- with passphrase:

```
openssl genrsa -des3 -passout pass:1234 -out webgate_key.pem  
2048
```

- without passphrase:

```
openssl genrsa -out webgate_key.pem 2048
```

- b. Generate CSR using the above private key:

```
openssl req -out webgate.csr -new -nodes -key webgate_key.pem -  
sha256
```

- c. Send the CSR to trusted CA to get the signed certificate. The following example shows signing CSR using self-signed CA certificate:

```
openssl x509 -req -days 360 -in webgate.csr -CA ../ca.cert.pem -  
CAkey ../ca.key.pem -CAcreateserial -out webgate_cert.pem -sha256
```

3. Place the generated `webgate_cert.pem` and `webgate_key.pem` files in the `$WEBGATE_INSTANCE_DIR/webgate/config/` directory.
4. If you have used passphrase to generate WebGate key then add the passphrase to `$WEBGATE_INSTANCE_DIR/webgate/config/wallet/cwallet.sso` with map name as `WG_Cert_PassPhrase`, map key name as `phrase_key`, and key name as `phrase`.

Set the WebGate Cert passphrase (if used to create `webgate_key.pem`) in the wallet, using the `mkstore` utility.

- a. Set `JAVA_HOME`

- b. Navigate to `<WebGate_Oracle_Home>/oracle_common/bin/mkstore`

- c. Run the following `mkstore` utility: `./mkstore -wrl $WEBGATE_INSTANCE_DIR/config/wallet -createUserCredential <mapName> <mapkeyName> <name> <Passphrase>`
For example:

```
./mkstore -wrl ./ -createUserCredential WG_Cert_PassPhrase phrase_key
phrase 1234
```

- d. Verify if the map and key entry are stored in the wallet:

```
./orapki wallet display -wallet wallet/
```

This displays the secret store entry. For example,
WG_Cert_PassPhrase@#3#@phrase_key

 **Note:**

If two way SSL is enabled, WebGate prints an INFO level log that shows "OAP over Rest (HTTPS): 2 way SSL config files are present in webgate instance directory"
This log is printed only if both the `webgate_cert.pem` and `webgate_key.pem` files are added in the WebGate instance directory.

B.3.5 Connection Tuning for OAP over REST

WebGate uses a dynamic connection pool for connections between WebGate and the REST endpoint (for example, OAM Server). The number of connections opened by WebGate is not fixed and is controlled by the `MinPoolSize` and `MaxPoolSize`.

`MinPoolSize`. This is the minimum number of connections made by WebGate. WebGate always keeps the connections defined by `MinPoolSize` opened, irrespective of the load.

`MaxPoolSize`. This is the maximum number of connections that are opened in load scenarios. If the connections in excess of `MinPoolSize` are idle for the specified time duration defined by `IdleConnectionTimeout`, the WebGate closes the connections.

For example, WebGate checks after every 60 seconds if the connections that are in excess of the `MinPoolSize` have been idle for more than `IdleConnectionTimeout` defined seconds. If yes, it closes those connections.

The connections established by WebGate with OAM server or Load balancer is always persistent. But some servers or LoadBalancers do not allow connections to be persistent by default.

Therefore, you must change the settings for these Servers for the connection timeout to be greater than what is defined in WebGate `IdleConnectionTimeout`.

For example, for the `mod_wl_proxy` plugin running on OHS, add the following setting into the OHS `httpd.conf` file:

```
KeepAlive on
MaxKeepAliveRequests 0
KeepAliveTimeout 90
```

And add `KeepAliveSecs 90` settings into the `mod_wl_proxy.conf` file:

Assuming it is a two-node managed server cluster, the `mod_wl_proxy` file must look similar to the following sample:

```
<IfModule weblogic_module>
WebLogicCluster
den01cbc.us.oracle.com:24100,den02kra.us.oracle.com:24100
WlSSLWallet /scratch/ranjakha/SSL_Certs
DynamicServerList ON
MatchExpression /oam
KeepAliveSecs 90
</IfModule>
```

B.3.6 Troubleshooting OAP over REST

This section provides troubleshooting steps for problems related to OAP over REST.

- [Error Performing Libcurl Operation](#)

B.3.6.1 Error Performing Libcurl Operation

WebGate and OAM Server throws error after switching to a communication method other than the default, OAP over REST.

Problem

The following error occurs when switching from the default OAP over REST to any other communication method:

```
oracle.security.am.proxy.oam.requesthandler.ObMessageIntegrityFailException: Message Integrity Check Failed
```

To solve the problem, perform the steps provided in either Solution 1 or Solution 2, as necessary.

Solution 1

1. Validate access to `http(s)://<OAMHOST>:<OAMPort>/iam/access/binding/api/v10/oap` through browser or the curl command.

Note:

If a proxy is in front of OAM HTTP/HTTPS port, ensure the proxy setting includes mapping for `/iam`.

2. Update OAM Server setting to reference the REST endpoint:
 - a. Login to the OAM console
 - b. Click **Configuration** tab, and under **Settings**, select **Access Manager**.
 - c. Update the following fields under **WebGate Traffic Load Balancer**, as necessary: **OAM Server Host**, **OAM Server Port**, and **OAM Server Protocol**.

- d. Click **Apply**.
3. Update WebGate agent to reference the REST endpoint:
 - a. In the OAM console, under the **Application Security** tab, click **SSO Agents**.
 - b. In the **Search SSO Agents** window, search for the required 12c agent and click on the agent.
 - c. In the **User Defined Parameters**, verify that the following parameters reference the same data as specified under *WebGate Traffic Load Balancer*.

```
OAMRestEndPointHostName=<hostname>  
OAMRestEndPointPort=<port>  
OAMServerCommunicationMode=<HTTP/HTTPS>
```

- d. Click **Apply**.

Solution 2

1. Login to the OAM console
2. In the OAM console, under the **Application Security** tab, click **SSO Agents**.
3. In the **Search SSO Agents** window, search for the registered agent and click on the agent.
4. From **User Defined Parameters**, remove the following parameters:

```
OAMRestEndPointHostName  
OAMRestEndPointPort  
OAMServerCommunicationMode
```

5. Click **Apply**.
6. Copy the newly created artifact files to the OHS/WebGate location.
7. Delete the **ObAccessClient.xml** from the following cache directory: `$DOMAIN_HOME/servers/<inst>/cache`
8. Restart the OHS/WebGate server.

B.4 Enabling FIPS Mode on Oracle Access Management

Topics

- [Enabling FIPS Mode on OAM Server](#)
- [Configuring SAML Federation for FIPS](#)
- [Enabling FIPS Mode on OAM Clients](#)

B.4.1 Enabling FIPS Mode on OAM Server

To enable FIPS mode on OAM server:

 **Note:**

- As a prerequisite, OAM server must be installed and configured.
- JDK used is Oracle JDK 1.8.0_271-b09.
- See [Enabling FIPS 140-2 Mode From Java Options in Administering Security for Oracle WebLogic Server](#) for detailed steps.

1. Update java security file of the JDK instance referred by your IDM WebLogic domain, as follows:

 **Note:**

You can obtain `JAVA_HOME` reference from the `SetDomainEnv` script.

- a. Add RSA Security Provider to the top of the security file `JAVA_HOME/jre/lib/security/java.security`.
- b. update the sequence number for the remaining providers, as shown:

```
security.provider.1=com.rsa.jsafe.provider.JsafeJCE
security.provider.2=com.rsa.jsse.JsseProvider
```

2. Update WLS Pre-ClassPath Setting with FIPS specific jars. To do so:
 - a. Set WLS `PRE_CLASSPATH` variable to point to `jcmFips.jar` and `sslj.jar`, which is in the `WL_HOME/server/lib/` directory.
 - b. Export `PRE_CLASSPATH` by adding an entry in the `setDomainEnv.sh` script, which is in the `DOMAIN_HOME/bin/` directory. The following is a sample entry:

```
PRE_CLASSPATH="WLS_HOME/server/lib/jcmFIPS.jar:WLS_HOME/server/lib/sslj.jar"
export PRE_CLASSPATH
```

Here, replace `WLS_HOME` with the absolute path of `WLS_HOME` in your environment after confirming that `jcmFIPS.jar` and `sslj.jar` exists in the location specified. This will set the `PRE_CLASSPATH` variable for the entire WLS Domain.

3. Restart the WebLogic Administrative Server and all Managed Servers.

B.4.2 Configuring SAML Federation for FIPS

You must enable FIPS on OAM Server. For details, see [Enabling FIPS Mode on OAM Server](#)

Perform the following steps to update the SAML configuration and profiles to make them FIPS compliant.

1. Enable FIPS on OAM Server. For det
2. The signing key size must be 2048 bits or more. If not, generate keys with key size 2048 by performing the following steps:

- a. Run the following command:

```
<JAVA_HOME>/bin/keytool -genkeypair -alias samlSigning -keyalg
RSA -keysize 2048 -sigalg sha256withrsa -dname cn="ACME SAML Signing"
-validity 1000 -keystore $DOMAIN_HOME/config/fmwconfig/.oamkeystore -
storetype JCEKS
```

- b. Run the following command:

```
<JAVA_HOME>/bin/keytool -genkeypair -alias samlEncryption -keyalg
RSA -keysize 2048 -sigalg sha256withrsa -dname cn="ACME SAML
Encryption"
-validity 1000 -keystore $DOMAIN_HOME/config/fmwconfig/.oamkeystore -
storetype JCEKS
```

- c. Login to the OAM Administration Console: `http(s)://<oam-admin-host>:<oam-admin-port>/oamconsole`

- d. Navigate to **Configuration, Federation Settings**

- e. In the **Keystore** section, create a new entry:

- i. In the Keystore section, click the **+** button
- ii. Enter a KeyID for the new entry (for example, `saml-signing`)
- iii. Select the alias for the new key entry from the drop-down, which lists the key entries in the `.oamkeystore` (for example, `samlSigning`)
- iv. Enter the password for the key entry that you set when creating that key.
- v. Repeat the process for other entries, if needed
- vi. Click **Apply**

- f. In the **General** section,

- i. Select the Signing Key from the dropdown list of key entries (these entries are defined in the Keystore section). For example, select `saml-signing`
- ii. Select the Encryption Key from the dropdown list of key entries (these entries are defined in the Keystore section). For example select `saml-encryption`
- iii. Click **Apply**

- g. Redistribute certificates and/or SAML 2.0 metadata to partners

3. Download the metadata using the following URL: `http(s)://<oam-host>:<manage_server_port>/oamfed/idp/metadata?signid=saml_signing&encid=saml_encryption&sigalgm=SHA-256`. The `signid` and `encid` value is as mentioned in the previous step.

4. Update the partners with new metadata.

5. Connect to WLST and run the following commands:

```
getStringProperty("/fedserverconfig/signaturedigestalgorithm")
```

```
putStringProperty("/fedserverconfig/signaturedigestalgorithm", "SHA-256")
```

6. Restart servers and access the resource.

B.4.3 Enabling FIPS Mode on OAM Clients

WebGate

FIPS mode works only when WebGate OAP is configured in CERT mode. FIPS is not supported in simple mode. For more information see, [User-Defined WebGate Parameters](#)



Note:

For WebGates, only OAP over TCP is FIPS compliant.

ASDK Client

1. Update security provider and Classpath settings. Add the following security providers in the java security file: `<JAVA_HOME>/jre/lib/security/java.security` and modify the sequence of the existing providers accordingly

```
security.provider.1=com.rsa.jsafe.provider.JsafeJCE
security.provider.2=com.rsa.jsse.JsseProvider
```

2. Invoke ASDK Client. Retrieve the `SimpleModeGlobalPassphrase` using WLST from your IDM Domain.

```
<JAVA_HOME>/jre/bin/java -cp .:<ASDK_HOME>/lib/*:<MW_HOME>/
oracle_common/modules/oracle.jps/jps-manifest.jar -
Dopss.tenant.mode=JPS_API -Djava.util.logging.config.file=
<ASDK_HOME>/log/log.properties -
Djava.security.properties=<JAVA_HOME>/jre/lib/security/
java.security -Dkeystore_passwd=<SimpleModeGlobalPassphrase> -
Djava.security.debug=access TestASDK_OAM11g
```

Configuring ASDK clients in Simple and Cert Modes

For the ASDK clients to work in Simple and Cert mode, the `password.xml` file in the ASDK configuration directory must include entries for both `passwd` and `keystore_passwd`.

The `passwd` value is created when the WebGate client is created in Simple and Cert mode. For example:

```
<?xml version="1.0"?>
<ParamsCtlg xmlns="http://www.acme.com" CtlgName="password">
  <ValNameList ListName="">
    <NameValPair ParamName="passwd"
Value="9a5cec58ce96dadf07f68a3616a20a3ebfcff90e05e15db8d47f45f84a78f6cf
775abf3
2c89beeb75ce3b3045a1e6fd0"/>
  </ValNameList>
</ParamsCtlg>
```

To add the `keystore_passwd` value, perform the following:

1. Navigate to `<OAM_DOMAIN_DIRECTORY>/output/webgate-ssl-SHA-256/` and open the `password.xml` file. The following example shows a sample content of the file:

```
<?xml version="1.0"?>
<ParamsCtlg xmlns="http://www.acme.com" CtlgName="password">
  <ValNameList ListName="">
    <NameValPair ParamName="passwd"
Value="02434507010457010c594505535b0d5f0e5b0d534b45025252500c0557"/>
  </ValNameList>
</ParamsCtlg>
```

2. Copy the value from `passwd` and add it under the `keystore_passwd` entry of the `password.xml` in the ASDK configuration directory. For example,

```
<?xml version="1.0"?>
<ParamsCtlg xmlns="http://www.acme.com" CtlgName="password">
  <ValNameList ListName="">
    <NameValPair ParamName="passwd"
Value="9a5cec58ce96dadf07f68a3616a20a3ebfcff90e05e15db8d47f45f84a78f6cf775
abf3
2c89beeb75ce3b3045a1e6fd0"/>
    <NameValPair ParamName="keystore_passwd"
Value="02434507010457010c594505535b0d5f0e5b0d534b45025252500c0557"/>
  </ValNameList>
</ParamsCtlg>
```

3. Save the file.

B.5 Configuring Cert Mode Communication for Access Manager

Configure Cert mode communication for Access Manager with at least one OAM Server instance running in the same mode as the agent.

This topic describes how to configure Cert mode communication for Access Manager. The following tasks apply to Cert mode only.



Note:

In Simple mode, the bundled Access Manager-CA-signed certificates are used and most of the following tasks are not needed.

Prerequisites

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.

Task overview: Adding certificates for the OAM Server includes

1. Reviewing:

- [Securing Communication Between OAM Servers and WebGates](#)
 - [About Cert Mode Encryption and Files](#)
2. [Generating a Certificate Request and Private Key for OAM Server](#)
 3. [Importing the Trusted, Signed Certificate Chain Into the Keystore](#)
 4. [Adding Certificate Details to Access Manager Settings](#)
 5. [Generating a Private Key and Certificate Request for WebGates](#)
 6. [Updating WebGate to Use Certificates](#)

B.5.1 About Cert Mode Encryption and Files

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).

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 Server. The root CA returns the certificates, which must be copied to the Webgate instance area manually after OAM 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 OAM 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`.

B.5.2 Generating a Certificate Request and Private Key for OAM Server

Retrieve the private key, certificate, and CA certificate for the OAM Server.

 **Note:**

The certified tool 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 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 -sha256
```

2. Create CA that generates `rootCA.key` and `aaa_chain.pem` files.

```
openssl genrsa -aes256 -out rootCA.key 4096
openssl req -x509 -new -nodes -key rootCA.key -days 7300 -sha256 -out
aaa_chain.pem
```

3. Submit the certificate request (`aaa_req.pem`) to a trusted CA to get a signed (`aaa_cert.pem`) certificate.

```
openssl x509 -req -in aaa_req.pem -CA aaa_chain.pem -CAkey rootCA.key -
CAcreateserial -sha256 -out aaa_cert.pem -days 500
```

4. Convert `aaa_cert.pem` into DER format.

```
openssl x509 -in aaa_cert.pem -inform PEM -out aaa_cert.der -outform DER
```

5. Convert `aaa_key.pem` into DER format.

```
openssl pkcs8 -topk8 -nocrypt -in aaa_key.pem -inform PEM -out
aaa_key.der -outform DER
```

 **Tip:**

The common name for generating a certificate request for OAM server could be the host name of the OAM cluster's load balancer for clustered environments and the name of the host where OAM server is deployed for the non-clustered environments.

B.5.3 Retrieving the .OAMKeystore password stored in UDM

You can retrieve the keystore credential with Oracle Enterprise Manager Fusion Middleware Control console.

1. Log into the Oracle Enterprise Manager Fusion Middleware Control 12c console using the context URL, `/em`.
2. Navigate to the System MBean Browser page, and select the **Operations** tab.
3. Search for the **credentialFromUDM** operation.
4. Click the **credentialFromUDM** operation link to open the corresponding page.
5. To retrieve the .OAMKeystore password, enter the following parameter values and click **Invoke**.

- p1 = **oracle.oam.OAMStore** (mapname)
- p2 = **JKS** (key)

The password is displayed in the **Return Value** pane.

B.5.4 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 Access Manager 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 Access Manager.
- 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 8 `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, Oracle Access Management and WebLogic Server do not support DSA and you must specify another key pair generation and signature algorithm.

Prerequisites

To import the trusted certificate chain into the keystore:

1. For setting up OAM Server in CERT mode, before making any changes to `.oamkeystore`, download the artifacts using offline WLST command:

```
downloadAccessArtifacts (domainHome="/new/path/base_domain",
propsFile="/path/dbschema.properties")

---- contents of dbschema.properties ----
oam.entityStore.schemaUser=MYPREFIX_OAM
oam.entityStore.schemaPassword=Secret
oam.entityStore.ConnectString=jdbc:oracle:thin:@dbhost.us.oracle.com
:1521/serviceName.us.oracle.com
```

Note:

At every restart of Admin servers , changes are pulled in from DB . Hence we need to `downloadAccessArtifacts` and `saveAccessArtifacts` , to save the cert mode changes .

2. Locate the `keytool` in the following path:

```
$MW_HOME/jdk8/bin/keytool
```

3. Unzip importcert.zip and locate the Readme file in the following location:

```
$ORACLE_IDM_HOME/oam/server/tools/importcert/README
```

4. **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-----
```

5. 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
```

6. When prompted to trust this certificate, type **yes**.

7. **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
```

8. **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
```

9. 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.

10. After making changes, please upload the changes to db using the following offline WLST command: `saveAccessArtifacts(domainHome="/mwhome/user_projects/domains/base_domain", propsFile="/path/dbschema.properties")`.
11. Proceed to [Adding Certificate Details to Access Manager Settings](#).

B.5.5 Adding Certificate Details to Access Manager Settings

After importing the certificates into the keystore, add the alias and password that you specified earlier into Access Manager settings configuration in Oracle Access Management Console.

 **Note:**

No explicit configuration is needed for Simple mode, which is provided out of the box.

 **See Also:**

- [Importing the Trusted, Signed Certificate Chain Into the Keystore](#)
- [Managing the Access Protocol for OAM Proxy Simple and Cert Mode Security](#)

To add certificate details to Access Manager Settings

1. In the Oracle Access Management Console, click **Configuration** at the top of the window.
2. In the **Launch Pad** tab, select **Access Manager** from the **View** drop-down menu in the **Settings** section.
3. In the **Access Protocol** section, 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*

4. Click **Apply** to save the configuration.

5. Close the page.
6. Open the OAM Server registration page, click the **Proxy** tab, change the **Proxy mode** to **Cert**, and click **Apply**.
7. Restart the OAM Server.
8. Proceed to the following topic:
See [Generating a Private Key and Certificate Request for WebGates](#).

B.5.6 Generating a Private Key and Certificate Request for WebGates

Retrieve the private key, certificate, and CA certificate for the WebGate using openssl.

The certified tool 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 -sha256
```

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
```

Tip:

The common name for generating a certificate request for WebGates could be the host name of the Web Server where the agent is deployed.

6. Proceed to [Updating WebGate to Use Certificates](#).

B.5.7 Supporting Two-Way SSL for CERT Mode Communication

In two-way SSL support for CERT mode communication, the user certificate is shared with server while performing the SSL handshake. Therefore, the user certificate must be added to `cwallet.sso`.

You must manually add the user certificate to the `cwallet.sso` using the `orapki` utility. See, [Keystore Management Tools in Administering Oracle Fusion Middleware](#)

Note down the DN of the user certificate before adding the user certificate to the wallet.

1. Check if the trusted certificate exists in `cwallet.sso`.

```
$MW_HOME/oracle_common/bin/orapki wallet display -wallet ./
```

If the command displays the trusted certificate in the wallet content, proceed to next step. If the trusted certificate is not displayed, add it to the wallet:

```
$MW_HOME/oracle_common/bin/orapki wallet add -wallet ./ -trusted_cert -cert
aaa_chain.pem -auto_login_only
```

2. Add user certificate request.

For example,

```
$MW_HOME/oracle_common/bin/orapki wallet add -wallet ./ -dn
'CN=oamserver,OU=OAM,O=Oracle,L=Canada,ST=Canada,C=US' -keysize 2048 -
auto_login_only
```

3. Export the user certificate request

For example,

```
$MW_HOME/oracle_common/bin/orapki wallet export -wallet ./ -dn
'CN=oamserver,OU=OAM,O=Oracle,L=Canada,ST=Canada,C=US' -request certreq.pem
```

4. Submit the user certificate request (certreq.pem) to a trusted CA to get a signed (aaa_cert_file.pem) certificate.

```
openssl x509 -req -days 1825 -in certreq.pem -CA aaa_chain.pem -CAkey
aaa_key.pem -set_serial 01 -out aaa_cert_file.pem
```

Add **-sha256** flag if required.

5. Add the user certificate to the wallet

For example,

```
$MW_HOME/oracle_common/bin/orapki wallet add -wallet ./ -cert
aaa_cert_file.pem -user_cert -auto_login_only
```

6. Verify the user certificate in the wallet

```
$MW_HOME/oracle_common/bin/orapki wallet display -wallet ./
```

Start the server instance and test the two-way SSL for CERT mode.

B.5.8 Updating WebGate to Use Certificates

For all communication modes (Open, Simple, or Cert), the Agent registration should be updated from the Oracle Access Management Console.

- Registering an Agent: If you choose Cert mode when registering an OAM Agent, a field appears where you can enter the Agent Key Password.
- Editing/Updating an Agent: When editing an OAM WebGate registration, password.xml is updated only when the mode is changed from Open to Cert or Simple to Cert.

Editing the agent Key Password does not result in creation of a new password.xml. In Cert mode, once generated, password.xml cannot be updated.

Prerequisites

[Adding Certificate Details to Access Manager Settings](#)

To update the communication mode in the WebGate Agent registration

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the **Launch Pad** tab, click **Agents**.

3. On the Search page, define your criteria and open the desired agent registration, as described in "[WebGate Search Controls](#)".
4. On the agent's registration page, locate the **Security** options and click **Cert** (or **Simple**).
5. **Cert Mode:** Enter the **Agent key Password** as specified in Step 5 of "[Generating a Private Key and Certificate Request for WebGates](#)".
6. Click **Apply** to submit the changes.
7. Copy your updated WebGate files as follows:

OAM WebGate:

```
ObAccessClient.xml
cwallet.sso (OAM WebGate only)
password.xml
```

- **From:** \$IDM_DOMAIN_HOME/output/AGENT_NAME
 - **To:** \$OHS_INSTANCE_HOME/config/OHS/ohs2/webgate/config
8. The following files that were created when [Generating a Private Key and Certificate Request for WebGates](#) are not required to be copied to the WebGate Server, as they were added to the `cwallet.sso` file using the `orapki` command:
 - `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
 9. Restart the OAM Server and the Oracle HTTP Server instance.

B.5.9 About WebGate Usage of PFS and Approved Cipher Suites for OAP Simple/Cert Mode Communication

WebGate ensures that valid and approved cipher suites defined by the admin are used when the Simple/Cert mode OAP communication occurs.

Administrators use WebGate user-defined parameter **TLSCipherSuite** to define ciphers. The default cipher used for Simple/Cert mode OAP communication is **PFS cipher suites**. Following are the supported cipher suites in **PFS cipher suites**:

Table B-4 PFS Cipher Suites

Cipher Name
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA

Following are the supported and approved cipher suites that administrators use for defining ciphers:

Table B-5 Supported Cipher Suites

Cipher Name
TLS_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_AES_128_GCM_SHA256

B.6 Configuring Simple Mode Communication with Access Manager

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.

 **Note:**

Communication between the agent and server works when the WebGate mode matches (or is higher) than the OAM Server mode.

When you register an OAM Agent or a new OAM Server, you can specify the Security mode. However, changing the global passphrase requires that you reconfigure all agents to use the mode and the new global passphrase.

 **Note:**

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.

The highest level of security is Cert mode, the lowest is Open mode. The agent mode can be higher but not lower. For example, Open mode can be updated to SIMPLE or CERT.

This section provides the information you need to configure Simple mode communication.

Task overview: Configuring Simple mode communication includes

1. Reviewing:
 - [About Simple Mode, Encryption, and Keys](#)
 - [The Importcert Tool](#)
2. [Retrieving the Global Passphrase for Simple Mode](#)
3. [Updating WebGate Registration for Simple Mode](#)
4. [Verifying SIMPLE Mode Configuration](#)

B.6.1 About Simple Mode, Encryption, and Keys

For Simple mode encryption, 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.

B.6.2 Retrieving the Global Passphrase for Simple Mode

Retrieve the random global passphrase generated by Access Manager for Simple mode communication during installation.

To retrieve the random global passphrase for Simple mode communication

1. Ensure that the Oracle Access Management Console is running.
2. On the computer hosting the Oracle Access Management Console, locate the WebLogic Scripting Tool in the following path. For example:


```
$ORACLE_IDM_HOME/common/bin
```

Where \$ORACLE_IDM_HOME represents the base 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](#)".

B.6.3 Updating WebGate Registration for Simple Mode

Artifacts generated for Simple Security mode use the Global Pass phrase and any change must be propagated to WebGates.

To update an existing WebGate registration for Simple mode, you can delete the WebGate registration using the Oracle Access Management Console, then re-register it (specifying Simple mode and disabling the automatic generation of policies). Alternatively, you can edit the WebGate registration and then copy the artifacts as described here.



See Also:

- [Viewing or Editing an OAM Agent Registration Page in the Console](#)

To update the WebGate registration for Simple mode

1. In the Oracle Access Management Console, click **Application Security** at the top of the window.
2. In the **Launch Pad** tab, click **Agents**.
3. On the Search page, define your criteria and open the desired agent registration, as described in "[WebGate Search Controls](#)".
4. In the registration page, locate the Security options and click **Simple**.

5. Click **Apply** to submit the changes.
6. Copy the updated WebGate files as follows:

OAM WebGate:

```
ObAccessClient.xml
cwallet.sso (OAM 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
7. Copy the following files, as directed for your WebGate release:

```
aaa_key.pem
aaa_cert.pem
```

OAM WebGate:

For 11g Webgate:

- **From:** \$IDM_DOMAIN_HOME/output/AGENT_NAME
- **To:** \$OHS_INSTANCE_HOME/config/OHS/ohs2/webgate/config/simple

For 12c Webgate:

- **From:** \$IDM_DOMAIN_HOME/output/AGENT_NAME
- **To:** \${Oracle_Home}/user_projects/domains/base_domain/config/fmwconfig/components/OHS/ohs1/webgate/config/simple

8. Restart the OAM Server and the Oracle HTTP Server instance.

B.6.4 Verifying SIMPLE Mode Configuration

Restart the Web server to instantiate the change to SIMPLE mode to validate the results

To validate SIMPLE mode changes:

1. From a command-line window, restart the Web server as shown in the following example:

For 11g Webgate:

```
d:\middleware\ohs_home\instances\ohs_webgate11g\bin
opmnctl stopall
opmnctl startall
```

For 12c Webgate:

```
$(Oracle_Home)/user_projects/domains/base_domain/bin/stopComponent.sh ohs1
( stopComponent.sh ohs1 ) $(Oracle_Home)/user_projects/domains/
base_domain/bin/startComponent.sh ohs1
```

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.

B.7 Configuring SSL in IHSWebServer

To configure the SSL in IHSWebServer run the following commands:

1. Navigate to the path where `./gskcapicmd` is located:

```
<ihsroot>/bin
```

2. From the directory where `./gskcapicmd` is present, execute the following command to create a keystore in the form of keystore file:

Syntax:

```
<ihsroot>/bin/gskcapicmd -keydb -create -db <database file> -pw  
<password> -stash
```

Example:

```
./gskcapicmd -keydb -create -db /opt/IBM/HTTPServer/certs/  
testkey1.kdb -pw Welcome1 -stash
```

3. Run the following command to create certificate request:

Syntax:

```
<ihsroot>/bin/gskcapicmd -certreq -create -db <database> -pw  
<password> \  
-dn <distinguished name> -label <labelname> -size <size> -file  
<outputfilename>
```

Example:

```
./gskcapicmd -certreq -create -db /opt/IBM/HTTPServer/certs/  
testkey1.kdb -pw Welcome1 -label testaix -dn  
"cn=slc00ywe.us.oracle.com,O=ORACLE,OU=IHS,L=RTP,ST=PUN,C=IN" -size  
2048 -file /opt/IBM/HTTPServer/certs/cert.csr
```

4. Run the following command to create authority private key, that helps to sign the certificate. This command can be run from any location:

```
openssl genrsa -out /opt/IBM/HTTPServer/certs/rootCA.key 2048
```

5. Run the following command to create trusted certificate of authority, this command can be run from any location:

```
openssl req -x509 -new -nodes -key /opt/IBM/HTTPServer/certs/  
rootCA.key -sha256 -days 1024 -out /opt/IBM/HTTPServer/certs/  
rootCA.pem
```

6. Run the following command to generate signed certificate in the form of .pem, this command can be run from any location:

```
openssl x509 -req -in /opt/IBM/HTTPServer/certs/cert.csr -CA /opt/IBM/HTTPServer/certs/rootCA.pem -CAkey /opt/IBM/HTTPServer/certs/rootCA.key -CAcreateserial -out /opt/IBM/HTTPServer/certs/aaa_cert.pem -days 500
```

7. Run the following command to list all the certificates present in the keystore file:

```
./gskcapiCmd -cert -list -db /opt/IBM/HTTPServer/certs/testkey1.kdb -pw Welcome1
```

8. Run the following command to list all the labels in the keystore file:

```
To list the label : ./gskcapiCmd -certreq -list -db /opt/IBM/HTTPServer/certs/testkey1.kdb -pw Welcome1
```

9. Run the following command to import signed certificate into the keystore file:

Syntax:

```
<ihsroot>/bin/gskcapiCmd -cert -import -db <inputp12file> -pw <pkcs12password>\ -target <existingkdbfile> -target_pw <existingkdbpassword>
```

Example:

```
./gskcapiCmd -cert -import -db /opt/IBM/HTTPServer/certs/aaa_cert.pem -pw Welcome1 -target /opt/IBM/HTTPServer/certs/testkey1.kdb -target_pw Welcome1
```

10. Run the following command to import Root CA certificate into keystore file:

Syntax:

```
<ihsroot>/bin/gskcapiCmd -cert -import -db <Root CA certificate file> -pw <pkcs12password>\ -target <existingkdbfile> -target_pw <existingkdbpassword>
```

Example:

```
./gskcapiCmd -cert -import -db /opt/IBM/HTTPServer/certs/rootCA.pem -pw Welcome1 -target /opt/IBM/HTTPServer/certs/testkey1.kdb -target_pw Welcome1
```

11. You need to add personal certificate to the keystore .kdb file:

- a. Navigate to the bin directory.

```
<<root directory>>/bin
```

- b. Launch IBM Key Manager utility by running the following command:

```
./ikeyman
```

- c. In the IBM Key Management window, from the **Key Database File** menu, select **Open** to open the .kdb file.

- d. In the **File Name** field, enter the actual file name of the .kdb file and in the **Location** field, enter the path of the directory where the .kdb file is located.
 - e. When prompted, enter the password for .kdb file.
 - f. In the IBM Key Management window, select **Personal Certificates** from the dropdown.
 - g. Click **New-Self Signed** to create a new self-signed certificate.
 - Enter a Label Name in **Key Label** field, ensure that you don't provide a label name that is already used.
 - Enter values in fields, **Common Name, Organization, Organizational Unit, Locality, State/Province, and Zipcode.**
 - Click **OK**
12. Open <<root directory>>/conf/httpd.conf and then modify httpd.conf file.
- Ensure the SSL section in httpd.conf is as shown below:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
Listen ihs_host:ihs_ssl_port
#SSLCheckCertificateExpiration 30
<VirtualHost ihs_host:ihs_sslport>
SSLEnable
SSLServerCert <Provide the labelname mentioned in step 11g>
</VirtualHost>
SSLDisable
KeyFile <path of the .kdb file>
# End of example SSL configuration
```

13. Restart the IHSWebServer and access the resource over https protocol and https port.

C

Troubleshooting

The following topics provide troubleshooting tips:

- [Introduction to Oracle Access Management Troubleshooting](#)
- [My Oracle Support for Additional Troubleshooting Information](#)
- [Oracle Access Management 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](#)
- [OAM unsupported Whole Server Migration](#)
- [Could Not Find Partial Trigger](#)
- [Denial of Service Attacks](#)
- [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](#)
- [Internationalization, Languages, and Translation](#)
- [Login Failure for a Protected Page](#)
- [OAM Metric Persistence Timer IllegalStateException: SafeCluster](#)
- [Partial Cluster Failure and Intermittent Login and Logout Failures](#)
- [RSA SecurID Issues and Logs](#)
- [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](#)
- [Time delay in configuration change](#)

- [Validation Errors](#)
- [Web Server Issues](#)
- [Windows Native Authentication](#)
- [WLST Commands for Multi-Data Centers](#)
- [Comparing Default Parameters and Values used in MDC Configuration for 11g and 12c](#)
- [Safari Browser Does not Display Options Under the Configuration Tab of the OAM Console URL](#)

C.1 Introduction to Oracle Access Management Troubleshooting

Oracle Access Management 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:

- [System Analysis and Problem Scenarios](#)
- [LDAP Server or Identity Store Issues](#)
- [OAM Server or Host Issues](#)
- [Agent-Side Configuration and Load Issues](#)
- [Runtime Database \(Audit or Session Data\) Issues](#)
- [Change Propagation or Activation Issues](#)
- [Policy Store Database Issues](#)

C.1.1 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:

- [System Analysis and Problem Scenarios](#)
- [LDAP Server or Identity Store Issues](#)
- [OAM Server or Host Issues](#)
- [Agent-Side Configuration and Load Issues](#)
- [Runtime Database \(Audit or Session Data\) Issues](#)
- [Change Propagation or Activation Issues](#)
- [Policy Store Database Issues](#)

C.1.2 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 time outs 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

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 [Logging Component Event Messages](#) (alternatively, in [Monitoring Performance and Logs with Fusion Middleware Control](#)).
5. Try to access Oracle Access Management 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 Management Console.
10. Correct the issue based on the requirements in your environment.

C.1.3 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 time outs 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 time outs 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
3. Bring up the OAM Server
4. Use the Access Tester to test authentication and authorization as described in [Validating Connectivity and Policies Using the Access Tester](#) .
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: AdminServer

1. Shut down the 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 [Registering and Managing OAM Agents](#) (this should fail).
4. Startup OAM AdminServer.

C.1.4 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.

C.1.5 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 [Logging Component Event Messages](#) (alternatively, in [Monitoring Performance and Logs with Fusion Middleware Control](#)).

C.1.6 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)

Diagnosis

See "[Policy Store Database Issues](#)"

C.1.7 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 Management Console to edit policies, and then observe errors in the AdminServer log file.

C.2 My Oracle Support for Additional Troubleshooting Information

You can use My Oracle Support (formerly MetaLink) to help resolve Oracle Fusion Middleware problems. My Oracle Support contains several useful troubleshooting resources, such as:

- Knowledge base articles

- Community forums and discussions
- Patches and upgrades
- Certification information

**Note:**

You can also use My Oracle Support to log a service request.

You can access My Oracle Support at <https://support.oracle.com>.

C.3 Administrator Lockout

Problem

Administrator cannot successfully log in to the Oracle Access Management Console. The following message appears:

```
Manually Change Identity Store Settings at OPSS Level and configure the IDMDomainAgent.
```

Cause

Access Manager secures the Oracle Access Management Console based on authentication information in the IAM Suite Application Domain: OAM Admin Console Policy. This policy relies on a single Authentication Scheme (OAMAdminConsoleScheme), which uses a Form challenge method and LDAP Authentication Module. The LDAP Authentication Module must be pointing to the User Identity Store designated as the System Store.

If, for example, your deployment is configured to use Oracle Internet Directory (with all Administrators, users, and groups defined therein) ensure that the LDAP Authentication Module points to this user identity store and that this is designated as the System Store.

Solution

1. Insert a user identity into both your designated system store and the Embedded LDAP store.
2. Log in to Oracle Access Management Console.
3. Configure the LDAP Authentication Module used by the designated System Store to point to the appropriate User Identity Store, as described in "[Managing Native Authentication Modules](#)".

C.4 Error During Federation Configuration After Upgrade from PS1 to PS2

IAM Suite is the OOTB Application Domain created when OAM 11.1.2 is installed. This Application Domain can be renamed after installation but when upgrading OAM to 11.1.2.2.0, it must be renamed back to **IAM Suite**. If it has been renamed, the upgrade operation will fail with this error seen in the WLS admin logs.

```
java.lang.NullPointerException  
at
```

```

oracle.security.am.common.policy.tools.upgrade.r2ps2.bootstrap.FedR2PS2Bootstr
apHandler.createFedAuthnResource (FedR2PS2BootstrapHandler.java:505)
at
oracle.security.am.common.policy.tools.upgrade.r2ps2.bootstrap.FedR2PS2Bootstr
apHandler.doBootstrap (FedR2PS2BootstrapHandler.java:151)
at
oracle.security.am.common.policy.tools.upgrade.r2ps2.bootstrap.R2PS2BootstrapH
elper.doBootstrap (R2PS2BootstrapHelper.java:70)
at
oracle.security.am.common.policy.tools.PolicyComponentLifecycle.initialize (Pol
icyComponentLifecycle.java:99)

```

If the **IAM Suite** Application Domain has been renamed, it is required to rename it back to its original **IAM Suite** name prior to beginning the upgrade process. After the upgrade, the name can be changed back to a custom name.

C.5 Oracle Access Management Console Inconsistent State

Problem

Administrators performing updates concurrently will result in an inconsistent state within the system configuration of the Oracle Access Management Console.

Cause

Concurrent configuration updates are not supported.

Solution

Only one Administrator should be allowed to modify the system configuration at any given time.

C.6 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 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\jdkxxx) 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

```

C.7 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, 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.

C.8 Application URL Requirements

The number of characters allowed in a URL are based on browser version.

The main attribute that affects the size of a cookie is the length of the requested URL. Some of the system generated URLs for ADF applications are quite long and can cause the cookie to exceed the maximum size.

Another case is when using custom plug-ins. The data that a plug-in adds to the authentication context is persisted in the cookie and can cause the cookie size to grow.

Multiple wrong password attempts can also add more context data to the cookie. Combined with one of the above cases, the cookie size can rapidly grow.

Solutions

Ensure that your applications do not use URLs that exceed the length that Access Manager and the browser can handle.

The cookie cache mode can be changed to `FORM` mode from default `COOKIE` mode. `FORM` mode works with long URLs. The only difference in behavior is for programmatic authentication, which requires a proper form Submit to pass the `OAM_REQ` parameter set to the form. Custom credential collection pages need to handle the `OAM_REQ` parameter that is submitted with the form.

Also, to support long URLs, set the `serverRequestCacheType` parameter to `FORM` using the `configRequestCacheType WLST` command. For more information about its affect on ECC or DCC configurations, see `TempStateMode` in [Table 22-25](#)

C.9 Authentication Issues

This section provides the following information:

- [Anonymous Authentication Issues](#)
- [X.509Scheme and SSL Handshake Issues](#)
- [X.509 Protected Resource and Single Sign Off](#)
- [X509CredentialExtractor Certificate Validation Error](#)

C.9.1 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.

C.9.2 X.509Scheme and SSL Handshake Issues

The Access Manager X.509 Authentication Scheme relies on SSL to deliver the user's X.509 certificate to the OAM Server. The X.509 Authentication Scheme requires the X.509Plugin as the value of the Challenge Method (not the Authentication Module).

Problem

User has selected his certificate in the Browser but the Certificate is not available to the OAM Server.

Solution

The specific solution will depend on the reason for the SSL Handshake failure. For instance:

- For debugging SSL connections terminating on the WebLogic Server, please refer to http://docs.oracle.com/cd/E12840_01/wls/docs103/secmanage/ssl.html
- For debugging SSL connections terminating on the OHS server, see http://docs.oracle.com/cd/E12839_01/web.1111/e10144/under_mods.htm#i1007687.

Determine the reason for the SSL Handshake failure and the peer that is terminating the SSL Handshake. The solution will fall into the following categories:

- [Configuration Issues](#)
- [Trust Issues](#)
- [Certificate Validation Issues](#)

C.9.2.1 Configuration Issues

If you are encountering problems establishing a SSL connection with the default WebLogic server SSL implementation, switch to using the JSSE SSL implementation which is supported with WLS 10.3.3+.

The following list identifies other possible configuration issues.

- OHS plugin is incorrectly configured and not sending the user certificate to the WebLogic server.
- Cipher suites: As configured, are not compatible with the user certificate.
- Smart cards: The browser is not communicating with the smart card reader.

- PKCS#11 (or hardware cryptography): Ensure that the devices are in working order.

C.9.2.2 Trust Issues

The server name within the certificate does not match the host name. This check can be disabled through configuration.

The server does not contain a CA certificate on the user certificate path in its trust store.

C.9.2.3 Certificate Validation Issues

The following list identifies possible configuration issues.

- Certificate has expired.
- Certificate has been revoked.
- Certificate validation is not working because this is incorrectly configured or there are connectivity issues.

C.9.3 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 then access the X.509-protected resource:

From the browser window, open the Tools menu, click Internet Options, choose Content, and then Clear SSL state.

C.9.4 X509CredentialExtractor Certificate Validation Error

Problem

Client certificate authentication works fine using the standard X509 Authentication Module after importing the root and sub CA certificates into the WebLogic Server and .oamkeystore keystores.

However, a certificate validation error can occur when using a Custom X509Plugin Authentication Module and root and sub CA certificates into the WebLogic Server and .oamkeystore keystores.

Solution

With the Custom X509Plugin Authentication Module the root and sub CA certificates must be added to the DOMAIN_HOME/config/fmwconfig/amtruststore because the X509CredentialExtractor plug-in loads certificates from this location.

C.10 Authorization Issues

This section provides the following topics:

- [Authorization Condition Error](#)
- [LDAP Search Filter Test Results](#)
- [Authorization Header Response Names](#)

C.10.1 Authorization Condition Error

An error is logged in the oam-server diagnostic log file whenever you create or edit an IPv4 range or temporal condition:

```
.... refreshPolicy specified but no response collector supplied
```

Cause

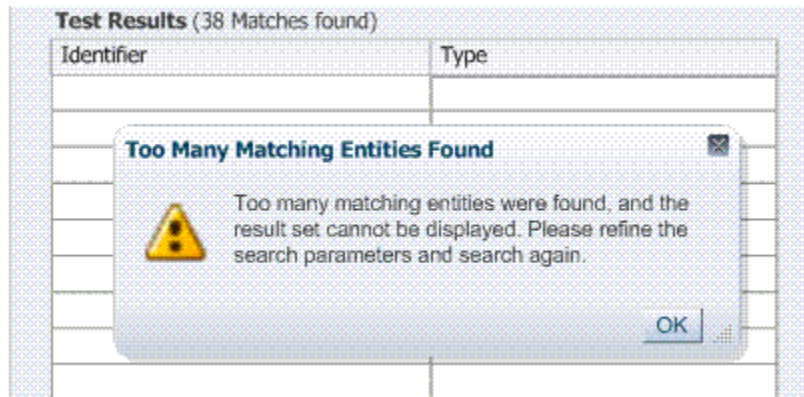
This is a message that is erroneously being logged at the ERROR level.

Solution

The correct level of the message is INFO.

C.10.2 LDAP Search Filter Test Results

If too many results are returned, you are informed as follows:



Solution

1. Click **OK**.
2. Click **Test Filter** to initiate a new test.
3. In the Edit Search Filter dialog, make your changes.
4. Check the Test Results.

C.10.3 Authorization Header Response Names

Some characters might not be usable within header response names or values, depending on whether the client receiving these responses is a Webgate, and if so which Web server is protected. Certain characters might be subject to automatic conversion to other characters in a server-specific way.

Oracle recommends that you refer to your Web server documentation for more details.

C.11 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.

C.12 Cannot Find Configuration

This section lists the following errors and provides a solution or a workaround.

- [Configuration Does Not Exist ...](#)

C.12.1 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 Access Manager.

C.13 OAM unsupported Whole Server Migration

Whole server migration is not supported for retaining virtual IP with OAM.

Solution

Whole server migration is not supported for OAM.

C.14 Could Not Find Partial Trigger

In the Administration Server output, you might see a "Could Not Find Partial Trigger" error (multiple times for each clicked policy configuration node or host identifier node) and also when you click any of other nodes in the navigation tree. This does not block functionality.

C.15 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 limits 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](#)

C.15.1 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

C.15.2 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.

 **See Also:**

- [Controlling Network Latency in *Performance and Tuning Guide*.](#)
- [Validating Connectivity and Policies Using the Access Tester](#)

C.15.3 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
```

C.16 Diagnosing 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](#)

C.16.1 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 [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.

C.16.2 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 [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.

C.16.3 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 jrockit for jrockit jvm located under JAVA_HOME/bin.

For jrockit jvm

```
jrockit pid <command>
/jrockit_160_14_R27.6.5-32/bin/jrockit 16775 heap_diagnostics
/jrockit_160_14_R27.6.5-32/bin/jrockit 16775 print_threads
/jrockit_160_14_R27.6.5-32/bin/jrockit 16775 jrarecording ....
```

For Sun jvm

```
jmap -histo <pid>  
jmap -dump:live,format=b,file=heap.bin <pid>
```

C.17 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 Access Manager's authentication.

For example, on the first request from an Internet Explorer (IE) browser to a resource on IIS protected by 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 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.

C.18 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.

C.19 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](#)

C.19.1 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.

C.19.2 IIS and General Web Component Guidelines

Following are some general guidelines to follow when installing Access Manager Webgates with IIS Web servers.

Account Privileges: The account that performs 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 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.

C.19.3 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 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.public.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-W3SVC-failing-to-start-with-Win32-Error-193-on-64bit-Windows.aspx>

C.19.4 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.

C.19.5 Removing and Reinstalling IIS DLLs

When Access Manager is running with Microsoft's IIS Web server, you must manually uninstall and reinstall the following ISAPI filters when reinstalling Access Manager.

- tranfilter.dll
- oblixlock.dll (if you installed Webgate)
- webgate.dll (if you installed Webgate)

To remove and reinstall IIS DLLs

1. Uninstall Access Manager.
2. Manually uninstall the preceding DLLs.
3. Reinstall 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.

C.20 Import and File Upload Limits

The UPLOAD_MAX_MEMORY and UPLOAD_MAX_DISK_SPACE is set to "50mb". To upload more than 50mb, manually change these settings in web.xml.

To reset the memory and disk space parameters

1. Locate web.xml in WEB-INF/lib/ngam-ui.war.
2. Edit the file to change UPLOAD_MAX_MEMORY. For example:

```
<context-param>
  <param-name>org.apache.myfaces.trinidad.UPLOAD_MAX_MEMORY</param-name>
  <param-value>104857600</param-value>
</context-param>
```

3. Edit the file to change UPLOAD_MAX_DISK_SPACE. For example:

```
<context-param>
  <param-name>org.apache.myfaces.trinidad.UPLOAD_MAX_DISK_SPACE</param-name>
  <param-value>104857600</param-value>
</context-param>
```

4. Save the file.
5. Restart the OAM Server.

**See Also:**

"Providing File Upload Capability" in the Oracle Application Development Framework Developer's Guide.

C.21 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.

C.22 Internationalization, Languages, and Translation

This section provides the following topics:

- [Automatically Generated Descriptions Are Not Translated](#)
- [Console Looks Messy](#)
- [Authentication Fails: Users with Non-ASCII Characters](#)
- [Access Tester Does Not Work with Non-ASCII Agent Names](#)
- [Locales, Languages, and Oracle Access Management Console Login Page](#)

C.22.1 Automatically Generated Descriptions Are Not Translated

The automatically generated Description for some components are 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.

C.22.2 Console Looks Messy

The Oracle Access Management Console displays policies and resources oddly when the input 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 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 Management Console might display policies and resources oddly following successful inband registration.

C.22.3 Authentication Fails: Users with Non-ASCII Characters

Configure Access Manager to use Kerberos Authentication Scheme with WNA challenge method, and create a non-ASCII user in Microsoft Active Directory.

Problem

An exception occurs when trying to get user details to populate the subject with the user DN and GUID attributes. Authentication fails and an error is recorded in the OAM Server log when a non-ASCII user in Active Directory attempts to access an Access Manager-protected resource:

```
... Failure getting users by attribute : cn, value ....
```

Cause

The username in the attribute is passed without modification as a java string.

Solution

Non-ASCII users can access the resource protected by Kerberos WNA scheme now by applying this JVM system property (for the built-in WebLogic SPNEGO support):

```
-Dsun.security.krb5.msinterop.kstring=true
```

C.22.4 Access Tester Does Not Work with Non-ASCII Agent Names

Register a Webgate with Access Manager using a non-ASCII name. In the Access Tester, enter the valid IP Address, Port, and Agent ID (non-ASCII name), then click Connect.

Connection testing fails.

C.22.5 Locales, Languages, and Oracle Access Management Console Login Page

When the browser locale is not supported, the Oracle Access Management 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 Management falls back to the server locale.
- If the server locale is not supported, Oracle Access Management falls back to English.

When users select an unsupported language and come to the 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 Access Manager SSO page and the original Access Manager login page also falls back to English.

C.23 Login Failure for a Protected Page

Problem

After installing OAM and protecting a page using a physical host and port, register the 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.

C.24 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 Management 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](#)".

C.25 Partial Cluster Failure and Intermittent Login and Logout Failures

Problem

In the event of a partial outage of 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 Access Manager application must be removed from service (shutdown) and brought back up upon recovery.

C.26 RSA SecurID Issues and Logs

Each OAM SecurID Server must be registered as a separate agent with the RSA Authentication Manager. This provisions the OAM SecurID Server with its own node secret file. Every OAM SecurID Server must have its configuration file stored under `$DOMAIN_HOME/config/fmwconfig/servers/$SERVER_NAME/oam`.

If the RSA SecurID authentication plug-in returns an error, it is logged in the OAM Server log. Web Server logs can also provide clues as to what might be going wrong. Be sure the enable logging on your Web server.

If communication has been established between the Access Server and Authentication Manager, the `sdadmin` tool provides access to logs under the Report menu. Both Activity and Exception reports may give you helpful information.

Verify Authentication Manager Logging Configuration and Reports

1. Confirm that you have added the user and assigned a token using the Authentication Manager Administrator tool, `sdadmin`.
2. Verify that you have copied the `sdconf.rec` file to the OAM Server.
3. In the Authentication Manager console, Report menu, open Activity and Exception reports for helpful information.

Check SecurID Plug-In Parameters with Modified HTML Fields

If you have modified the HTML field names in the HTML forms, ensure that the RSA SecurID plug-in parameters are configured to match.

Remove the @ character From any Login Attribute Value

User login can fail if there is an at-sign (@) in the login attribute value. This is a known issue with SecurID.

C.27 Registration Issues

This section provides the following information.

- [Problem: Remote Registration Tool Failure](#)
- [Problem: No ObAccessClient.xml File Generated](#)
- [Problem: Partner Registration Failure](#)
- [Problem: Remote Registration Failure in upgraded OAM12c environment](#)

C.27.1 Problem: Remote Registration Tool Failure

Solution

Ensure that the agent name is unique (does not already exist) and that the AdminServer is running.

C.27.2 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 "[Resource URL, Prefixes, and Patterns](#)".

C.27.3 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.

C.27.4 Problem: Remote Registration Failure in upgraded OAM12c environment

Remote registration fails if the user is not using Oracle Home of OAM12c and might encounter the following error:

```
SEVERE: Exception encountered: RemoteAgentRegistrationException. Specific
exception:Cannot reach admin server at : http://
slc12ors.us.oracle.com:7001. Please make sure the server url provided is
valid and that the server is up before trying again.
```

Solution

Ensure that Oracle Home used in the remote registration is of OAM12c and not R2PS3.

C.28 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.

C.29 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 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 Webgate):

OAM Webgate

```
$Webgate_OH/webgate/ohs/lib/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 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
```

C.30 Session Issues

This section provides the following details:

- [Session Impersonation Not Enabled by Default](#)
- [Sessions with Oracle Access Manager 11.1.1 Integrated with Oracle Identity Federation 11.1.1](#)

C.30.1 Session Impersonation Not Enabled by Default

Session impersonation is not enabled by default. You can update the value in `oam-config.xml`, then 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`.

C.30.2 Sessions with Oracle Access Manager 11.1.1 Integrated with Oracle Identity Federation 11.1.1

Expected Behavior: Oracle Identity Federation 11.1.1 session is not cleared

When Oracle Access Manager 11.1.1 is integrated with Oracle Identity Federation 11.1.1, and you clear the session using the console, only the Oracle Access Manager session is cleared. The Oracle Identity Federation session is not cleared.

C.31 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.

C.32 Start Up Issues

This section lists the start up issues and its solution.

- [AdminServer Startup \(or Remote Registration Tool Failure\) on AIX Platforms](#)

C.32.1 AdminServer Startup (or Remote Registration Tool Failure) on AIX Platforms

Problem

AdminServer start up fails with following message:

```
"java.net.SocketException:
No buffer space available".
```

Configuration for the number of AIX file descriptors set for the operating system is substantially high (`ulimit` file descriptor) 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
```

C.33 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.

C.34 Time delay in configuration change

If any configuration change is made, there may be a time delay for that change to be refreshed in the runtime. This time delay is because the servers will need few seconds to refresh the configuration after it was changed.

For Example, Suppose you create a new password after setting the “Disallowed Previous Passwords” option in the password policy, the previous password should not be allowed to access the protected resource. In this case, the previous password is valid if the user tries to use the password without waiting for 60 seconds.

Solution

When a configuration change is made, you should wait for a minimum of 60 seconds for the changes to reflect.

C.35 Validation Errors

The following sections provide information on the Validation Errors and solution to the problems.

- [Resource not added to Authentication or Authorization Policy](#)
- [Validation Failure - "description" attribute is not valid](#)

C.35.1 Resource not added to Authentication or Authorization Policy

Problem

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.

C.35.2 Validation Failure - "description" attribute is not valid

Problem

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.

C.36 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](#)

C.36.1 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). stacktrace:  
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

C.36.2 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

C.36.3 Apache v2 Bundled with Red Hat Enterprise Linux 4

Problem

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 Access Manager Webgates.

C.36.4 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 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"

C.36.5 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 multi-threaded 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_palloc(pool, sizeof(apr_threadattr_t));
4-----> (*new)->attr = (pthread_attr_t *)apr_palloc(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`.

C.36.6 Domino Web Server Issues

Failure Authentication Event: For Domino Web servers, the redirection of a URL through 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.

C.36.7 Errors, Loss of Access, and Unpredictable Behavior

Symptom

If you installed Access Manager on UNIX under a different user ID than you used to create your Web server instance, 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 Access Manager directory to the same user ID that you used to create your Web server instance.

C.36.8 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 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.

C.36.9 Oracle HTTP Server Fails to Start with LinuxThreads

Problem

After installing a Webgate instance on an Oracle HTTP Server, the server does not start up.



Note:

When running 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 Access Manager, you do not set `LD_ASSUME_KERNEL` to 2.4.19.9

Cause

This occurs because Access Manager uses an older Linux threading model.

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">`
- 4.

Add the following information under the line you found in the previous step:

```
<environment>  
<variable id="LD_ASSUME_KERNEL" value="2.4.19" />  
</environment>
```

5. Save this file.
6. Run the following commands to implement your changes:

```
opmnctl stopall  
opmnctl startall
```

C.36.10 Oracle HTTP Server Webgate Fails to Initialize On Linux Red Hat 4

This situation might arise whether you are using 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.

C.36.11 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
```

C.36.12 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 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.public.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-W3SVC-failing-to-start-with-Win32-Error-193-on-64bit-Windows.aspx>

C.36.13 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

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 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

C.36.14 Removing and Reinstalling IIS DLLs

When Access Manager is running with Microsoft's IIS Web server, you must manually uninstall and reinstall the following ISAPI filters when reinstalling Access Manager.

- `tranfilter.dll`
- `oblloxlock.dll` (if you installed Webgate)
- `webgate.dll` (if you installed Webgate)

To remove and reinstall IIS DLLs

1. Uninstall Access Manager.
2. Manually uninstall the preceding DLLs.
3. Reinstall 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.

C.37 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 Management might not point to Windows Active Directory. By default, the identity store is Embedded LDAP.

Solution

1. In the Oracle Access Management Console, review the identity store configuration: System Configuration, Data Sources, User Identity Store.
2. Confirm the LDAP store settings point to Active Directory.

C.38 WLST Commands for Multi-Data Centers

The following WebLogic Scripting Tool (WLST) commands are specific to Multi-Data Center deployment. More information is in the following sections.

- `enableMultiDataCentreMode`
- `disableMultiDataCentreMode`
- `addPartnerForMultiDataCentre`
- `removePartnerForMultiDataCentre`
- `setMultiDataCenterType`
- `setMultiDataCenterWrite`

- [setMultiDataCentreClusterName](#)
- [validateMDCConfig](#)
- [exportAccessStore](#)
- [importAccessStore](#)



See Also:

WebLogic Server WLST Online and Offline Command Reference

C.38.1 enableMultiDataCentreMode

Online command used to enable Multi-Data Center mode.

Description

This command enables Multi-Data Center mode. It takes a value equal to the full path to, and name of, the MDC.properties file.



Note:

Setting the SSO Token version to 5 is not supported from the administration console. To do this, modify the Access Manager Settings page and run the enableMultiDataCentreMode WLST command to set.

Syntax

```
enableMultiDataCentreMode(propfile="../MDC_properties/oamMDCProperty.properties")
```

Argument	Definition
<i>propfile</i>	Mandatory. Takes a value equal to the full path to, and name of, the oamMDCProperty.properties file. Table C-1 documents the properties that comprise the file. The example (following the table) is a sample oamMDCProperty.properties file.

Table C-1 oamMDC.properties Properties

Property	Definition
SessionMustBeAnchoredToDataCenterSe rvicingUser	Takes a value of True (Invalidate) or False (No Invalidation).
SessionDataRetrieval OnDemand	Takes a value of True (Cross DC retrieval) or False (No). Data retrieval can be turned off without disabling MDC. If False, session data is not transferred but SSO is still performed as the user moves across DCs. NOTE: SessionDataRetrievalOnDemand must be set to False when deploying in Co-existence mode.
Reauthenticate	Takes a value of True (force reauthentication) or False (No forced reauthentication).

Table C-1 (Cont.) oamMDC.properties Properties

Property	Definition
SessionDataRetrievalOnDemandMax_retry_attempts	Takes a value equal to a binary that represents the number of times to retry data retrieval when it fails. Default is 2.
SessionDataRetrievalOnDemandMax_connection_wait_time	Takes a value equal to a binary that represents the total amount of time in milliseconds to wait for a connection. Default is 1000.
SessionContinuationOnSyncFailure	Decides the session adoption action on fail over. When set to 'true', the session will continue on the DC servicing the current request even though the parent DC is down/not reachable. The session will be created in the DC servicing the current request from the mandatory minimal information available in the incoming token. When set to 'false', the user will be challenged on fail-over scenarios.
MDCGitoCookieDomain	Specifies the domain with which the OAM_GITO cookie should be set. In MDC deployments where a common cookie domain hierarchy cannot be derived, this setting should be commented or removed as described in Inactivity time outs scenario.

Sample oamMDCProperty.properties File

```
SessionMustBeAnchoredToDataCenterServicingUser=false
SessionDataRetrievalOnDemand=true
Reauthenticate=true
SessionDataRetrievalOnDemandMax_retry_attempts=3
SessionDataRetrievalOnDemandMax_conn_wait_time=80
SessionContinuationOnSyncFailure=true

#MDCGitoCookieDomain=.example.com <This setting should be provided only if there
is a common cookie subdomain across the WGs and DCs>
```

Example

The following command enables this data center.

```
enableMultiDataCentreMode(propfile="../MDC_properties/oamMDCProperty.properties")
```

C.38.2 disableMultiDataCentreMode

Online command used to disable Multi-Data Center mode.

Description

This command disables Multi-Data Center mode.

Syntax

```
disableMultiDataCentreMode()
```

There are no arguments for this command.

Example

The following command disables Multi-Data Center mode.

```
disableMultiDataCentreMode()
```

C.38.3 addPartnerForMultiDataCentre

In an MDC deployment with n number of Data Centers, each Data Center has a registered partner to communicate with each of the other $(n-1)$ Data Centers. This makes the total number of partner registrations $(n) \times (n-1)$. This online command is used to add a partner for inter Data Center OAP communication.

Note:

An MDC partner profile is exposed by each data center and used by other data centers to communicate with it. Registering an MDC partner is a two step process. Consider an MDC with three data centers. In DC1, expose an MDC partner profile by creating an OAM WebGate (DC1_MDC_Partner). Then, register DC1_MDC_Partner in DC2 and DC3 using `addPartnerForMultiDataCentre`. See [addPartnerForMultiDataCentre](#) for details.

Description

This command adds a partner to the Data Center. It takes a value equal to the full path to, and name of, the `partnerInfo.properties` file.

Syntax

```
addPartnerForMultiDataCentre(propfile="../MDC_properties/partnerInfo.properties")
```

Argument	Definition
<i>propfile</i>	Mandatory. Takes a value equal to the path to, and name of, the <code>partnerInfo.properties</code> file.
<i>RESTEndpoint</i>	Optional. Takes as a value the HTTP/HTTPS URL from which the Access Manager REST services can be accessed.

[Table C-2](#) documents the properties that comprise `partnerInfo.properties`. See [Multi-Data Center Security Modes](#) for properties file samples.

Table C-2 partnerInfo.properties Properties

Property	Definition
<code>remoteDataCentreClusterId</code>	Cluster id of the remote Data Center with which the OAP communication needs to be established.
<code>oamMdcAgentId</code>	Partner ID of the registered partner profile in the remote Data Center. The "allow management operations" flag for this partner should be set in the remote Data Center.
<code>PrimaryHostPort</code>	Takes a <i>fully-qualified-host-name:OAM-port</i> for the primary Access Manager server corresponding to the remote DC identified by <code>remoteDataCentreClusterId</code> ; for example: <code>PrimaryHostPort=abc.example.com:5575</code>

Table C-2 (Cont.) partnerInfo.properties Properties

Property	Definition
SecondaryHostPort	<p>Takes a <i>fully-qualified-host-name:OAM-port</i> for the secondary Access Manager server corresponding to the remote DC identified by remoteDataCentreClusterId; for example: SecondaryHostPort=abc.example.com:5577</p> <p>Consider an OAM MDC member Data Center with two managed servers at abc.example.com with ports as follows: oam_server1 (5575) and oam_server2 (5577). High availability/failover of the OAP SDK partner can be achieved by setting the PrimaryHostPort and SecondaryHostPort as below. PrimaryHostPort=abc.example.com:5575 SecondaryHostPort=abc.example.com:5577</p>
AccessClientPasswd	The access client password of the MDC partner registered in the remote Data Center.
oamMdcSecurityMode	<p>Defines the MDC security mode. Takes a value of OPEN/SIMPLE/CERT. (CERT Mode is preferred, SIMPLE is fine but OPEN is discouraged.)</p> <p>For SIMPLE and CERT modes, the following values should be supplied appropriately. For OPEN mode, these values are not applicable. See Multi-Data Center Security Modes.</p>
agentVersion	Valid agent version 11g.
trustStorePath	Absolute path to the truststore file [SIMPE/CERT].
keyStorePath	Absolute path to the keyStore file [SIMPLE/CERT].
globalPassPhrase	Global passphrase set during the partner registration [SIMPLE/CERT].
keystorePassword	Key store password set during partner configuration [SIMPLE/CERT].

Example

The following command defines this data center as a Master.

```
addPartnerForMultiDataCentre (propfile=" ../MDC_properties/partnerInfo.properties")
```

C.38.4 removePartnerForMultiDataCentre

Online command used to remove a registered remote partner from the Data Center configuration.

Description

This command removes a registered remote partner from a configured Data Center. It takes a value equal to a valid remoteDataCentreClusterId.

Syntax

```
removePartnerForMultiDataCentre=("<cluster_ID>")
```

Argument	Definition
<i>cluster_ID</i>	Mandatory. Takes a string value equal to the cluster ID.

Example

The following command defines the partner to be removed.

```
removePartnerForMultiDataCentre("99bf9-adc2120609")
```

C.38.5 setMultiDataCenterType

Online command used to set the type of data center - either Master or Clone.

Description

In an MDC deployment one Data Center is designated as the Master and the others as a Clone. Essentially all MDC wide global configurations and policy updates should be applied to the Master and propagated to the Clones. This command sets the type of the data center. Values are Master or Clone.

Syntax

```
setMultiDataCenterType(DataCenterType="<Master|Clone>")
```

Argument	Definition
<i>DataCenterType</i>	Mandatory. Takes a string value of Master or Clone.

Example

The following command defines this data center as a Master.

```
setMultiDataCenterType(DataCenterType="Master")
```

C.38.6 setMultiDataCenterWrite

Online command used to set write protection for modifications to system and policy configurations on the Clone Data Center.

Description

A Clone Data Center can be write protected by setting `WriteEnabledFlag` to false. In this case, the Clone Data Center will not allow updates through the Oracle Access Management Console or WLST commands. Data synchronization will still continue to update as the command is used to write protect the Clone Data Center against accidental updates after the initial set up is complete.

Syntax

```
setMultiDataCenterWrite(WriteEnabledFlag="<true|false>")
```

Argument	Definition
<i>WriteEnabledFlag</i>	Mandatory. Takes a string value of true or false.

Example

The following example protects the Clone Data Center from accidental overwrites.

```
setMultiDataCenterWrite (WriteEnabledFlag = "false")
```

C.38.7 setMultiDataCentreClusterName

Online command to set the cluster name of the Data Center to the supplied string.

Description

This command sets the Multi-Data Center cluster name. Value is a string.

Syntax

```
setMultiDataCentreClusterName (clusterName="<string_value>")
```

Argument	Definition
<i>clusterName</i>	Mandatory. Takes a string equal to the cluster name.

Example

The following command enables this data center.

```
setMultiDataCentreClusterName (clusterName="MyCluster")
```

C.38.8 validateMDCConfig

Online command used to insure the Multi-Data Center configuration is correct.

Description

This command validates that the required entries in the Multi-Data Center configuration are present in `oam-config.xml`. For the MDC solution, a new Access Manager event named `mdc_session_update` is required to create or update MDC sessions during authorization. The Access Manager event model requires a set of configurations to be present in the `oam-config.xml` configuration file. The required configurations cannot be added statically so `validateMDCConfig` validates the required entries for `mdc_session_update` and seeds any configurations not already present.

Syntax

```
validateMDCConfig()
```

There are no arguments for this command.

Example

The following command validates the MDC configuration.

```
validateMDCConfig()
```

C.38.9 exportAccessStore

Online command to create a ZIP file of the Master Data Center UDM metadata.

Description

This command will create a ZIP file of the Master Data Center UDM metadata.

Syntax

```
exportAccessStore(toFile="<name and location of ZIP", namePath="/")
```

Example

```
exportAccessStore(toFile="/master/location/dclmetadata.zip", namePath="/")
```

C.38.10 importAccessStore

Online command to import a ZIP file of the Master Data Center UDM metadata to a Clone Data Center.

Description

This command will import a ZIP file of the Master Data Center UDM metadata to the Clone Data Center.

Syntax

```
importAccessStore(fromFile="<name and location of ZIP", namePath="/")
```

Example

```
importAccessStore(fromFile="/master/location/dclmetadata.zip", namePath="/")
```

C.39 Comparing Default Parameters and Values used in MDC Configuration for 11g and 12c

Configuring MDC in 12c environment is made simple with the new set of MDC Admin REST APIs. Many of the parameters and properties that were manually set in 11g are now defaulted in 12c.

The following table details the parameters in 12.2.1.3.0 which are defaulted. You can override the default setting, if required.

Parameter and Description	Mandatory / Optional	Default Value
mdcTopologyType (ACTIVE_ACTIVE or DISASTER_RECOVERY)	Mandatory	NA
masterMDCAgentID (MDC NAP Agent Name for Master)	Mandatory	NA
cloneMDCAgentID (MDC NAP Agent Name for Clone)	Mandatory	NA
accessClientPassword (password to be used for the MDC NAP agents in Master and Clone)	Mandatory	NA
cloneServerURL (URL of the clone admin server or the URL of the reverse proxy front-ending the clone admin server)	Mandatory	NA

agentKeyPassword (Agent Key Password used to register partners in CERT mode)	Mandatory	NA
certModeKeystorePassword (Keystore Password used to protect clientTrustStore.jks and clientKeyStore.jks)	Mandatory	NA
masterServerURL (URL of the master admin server or URL of the reverse proxy front ending the master admin server)	Optional for Master configuration; Mandatory for Clone configuration	If not provided, during Master configuration it retrieves Admin server's host and port details from MBean.
artifactPassword (password used for protecting the cloning artifacts.)	Mandatory	NA
cloneAdminUserNamePassword (UserName:Password of Clone DC Administrator; specify only when the Master and Clone Admin Users/Password are different)	Optional	UserName:Password of Master data center.
trustStorePath (path to clientTrustStore.jks file; specify only when the clientTrustStore.jks file is present in any folder other than DOMAIN_HOME/config/fmwconfig/oam-mdc-cert-artifacts)	Optional	CERT mode: %DOMAIN_HOME%/config/fmwconfig/oam-mdc-cert-artifacts SIMPLE mode: %DOMAIN_HOME%/output/webgate-ssl-SHA-256/
keyStorePath (path to clientKeyStore.jks file; specify only when the clientTrustStore.jks file is present in any folder other than DOMAIN_HOME/config/fmwconfig/oam-mdc-cert-artifacts)	Optional	CERT mode: %DOMAIN_HOME%/config/fmwconfig/oam-mdc-cert-artifacts SIMPLE mode: %DOMAIN_HOME%/output/webgate-ssl-SHA-256/
artifactsZipLocation (location where artifacts are stored; specify if artifacts are stored in a location other than /tmp)	Optional	/tmp
isMultiDataCenterEnabled (used to disable MDC)	Mandatory	NA
isBackwardCompatible (used to enable backward compatibility when master and clone data centers have different OAM versions.)	Mandatory	NA
clusterName (cluster name of a data center)	Mandatory	NA

masterArtifactsZipLocation (location where cloning artifacts are present in Master; specify only when artifactsZipLocation was used in input while configuring the Master data center)	Optional	/tmp
--	----------	------

C.40 WADL Generation Does not Show Description

Issue

WADL generation fails and a `java.lang.IllegalStateException: ServiceLocatorImpl` is returned.

```
Exception thrown when provider
class
org.glassfish.jersey.server.internal.monitoring.MonitoringFeature$StatisticsL
istener
was processing MonitoringStatistics. Removing provider from further
processing.
java.lang.IllegalStateException:
ServiceLocatorImpl(__HK2_Generated_6,9,221656053) has been shut down
at
org.jvnet.hk2.internal.ServiceLocatorImpl.checkState(ServiceLocatorImpl.java:
2393)
```

Also, when the WADL generation fails, the description field shows **Root Resource**, instead of a proper description in the following URLs.

```
http://<Host>:<AdminServerPort>/oam/services/rest/11.1.2.0.0/ssa/policyadmin/
application.wadl
http://<Host>:<ManagedServerPort>/iam/access/api/v1/health/application.wadl
```

Resolution

Restart the Admin server and managed servers to resolve the wadl issue.

C.41 Safari Browser Does not Display Options Under the Configuration Tab of the OAM Console URL

Problem

The drop-down option under **Configuration > Settings** does not appear on Safari version 15 when OAM is deployed on Windows 2022. However, it works on other browsers.

Solution

If you are using a Windows or Mac machine, you can fix the Safari 14+ version drop-down issue by clicking the drop-down once and hitting Enter/Return.