Oracle Financial Services Revenue Management and Billing

OR

Oracle Insurance Revenue Management and Billing

Version 8.0.0.0.0

Security Guide

Revision 25.0

G30503-01

April 2025

Oracle Financial Services Revenue Management and Billing/Oracle Insurance Revenue Management and Billing Version 8.0.0.0.0 Security Guide

Note: To improve the content readability, the above two products are collectively referred to as Oracle Revenue Management and Billing throughout this document.

G30503-01

Copyright Notice

License Restrictions

Warranty Disclaimer

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.

Restricted Rights Notice

Hazardous Applications Notice

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.

Trademark Notice

Third-Party Content, Products, and Services Disclaimer

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.

Preface

About This Document

This document describes how you can configure security for the Oracle Revenue Management and Billing (ORMB) application by using the default security features.

Intended Audience

This document is intended for the following audience:

- Product, Database and Security Administrators
- Development Team
- Consulting Team
- Implementation Team

It is also intended for other users who perform the following operations securely and efficiently:

- Design and implement security policies to protect the data of an organization, users, and applications from accidental, inappropriate, or unauthorized actions.
- Create and enforce policies and practices of auditing and accountability for inappropriate or unauthorized actions.
- Create, maintain, and terminate user accounts, passwords, roles, and privileges.
- Develop interfaces that provide desired services securely in a variety of computational models, leveraging product and directory services to maximize both efficiency and ease of use.

To use this document, you need a basic understanding of how the ORMB application works, and basic familiarity with the security aspects of Oracle WebLogic and Oracle Database.

Organization of the Document

The information in this document is organized into the following sections:

Section No.	Section Name	Description
Section 1	What's New in Security?	Describes the new security features introduced in Oracle Revenue Management and Billing.
Section 2	Introducing Security	Lists the different ways in which Oracle Revenue Management and Billing protects access to the product, its functionality and the underlying data stored and managed through the product.
Section 3	Authentication	Explains how the online, batch, and web service authentication is handled in Oracle Revenue Management and Billing.

Section No.	Section Name	Description
Section 4	Authorization	Describes how the identified users are authorized to use specific functions and data within the product. It also provides detailed information about the security authorization model used by Oracle Utilities Application Framework (OUAF).
Section 5	Managing Security	Describes how to manage the security definitions from the product, security infrastructure and from the security repositories.
Section 6	Advanced Security	Lists and describes the advanced security settings that can be configured to support various security requirements.
Section 7	Audit Facilities	Explains how to configure the audit facility to track changes made to the key data.
Section 8	Database Security	Lists a predefined set of database users and roles shipped with the product. It also explains various database security methods that can be used to provide restricted access to the database users.
Section 9	Security Integration	Lists and describes the additional security features or security products that can be integrated with the product to augment the security setup.
Section 10	Keystore and Truststore Support	Explains how to use keystore to encrypt and decrypt data, such as passwords. It describes how to create and modify a keystore and how to synchronize data encryption.
Section 11	Encryption Feature Type	Explains how to create a feature configuration for encryption using the Encryption feature type.
Section 12	Web Services Security	Explains how the inbound web services capability, based upon a JAX-WS implementation provides the WS-Security and WS-Policy support on individual inbound web services from several perspectives.
Section 13	Allow List Support	Explains the use of allow list, which is enforced to protect resources within the oracle cloud implementation. These allow lists can also apply to non-cloud implementations and in some cases can be extended to suit individual needs.
Section 14	Custom Authentication Service Provider	Provides support for complex domain security configurations.
Section 15	Federated Security Support	Explains how Oracle Revenue Management and Billing provides support for the federated security. It explains how to configure the federated online authentication, federated web services, and the federated outbound messages.

Section No.	Section Name	Description
Section 16	Securing JNDI Access	Explains how to provide restricted access to JNDI in Oracle Revenue Management and Billing.
Section 17	Object Erasure Support	Explains how to configure the Object Erasure functionality in Oracle Revenue Management and Billing.
Section 18	Key Ring Support	Explains how to maintain and use the key rings in the system.
Section 19	Redaction Rules	Explains how to maintain redaction functions and rules in the system.
Section 20	Cookies Used by ORMB	Lists and describes the cookies used by ORMB.
Section 21	Garbling Customer Information	Explains how to garble the personal and account related information of a customer.

Conventions

The following conventions are used across this document:

Convention	Meaning
boldface	Boldface indicates graphical user interface elements associated with an action, or terms defined in the text.
italic	Italic indicates a document or book title.
`monospace`	Monospace indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Points to Note:

Menu options in this document assume the use of Alphabetic sorting. If alternatives are used, then adjust the advice accordingly.

Screen images in this document are for illustrative purposes only.

Acronyms

The following acronyms are used in this document:

Acronym	Meaning
ORMB	Oracle Revenue Management and Billing
OUAF	Oracle Utilities Application Framework
JAAS	Java Authentication and Authorization Services
LDAP	Lightweight Directory Access Protocol
SAML	Security Assertion Markup Language
JMX	Java Management Extensions

Acronym	Meaning
TDE	Transparent Data Encryption
DML	Data Manipulation Language
DDL	Data Definition Language
SSO	Single Sign-On
SPNEGO	Simple and Protected Negotiate
CBC	Cipher-Block Chaining
XAI	XML Application Integration
OWSM	Oracle Web Services Manager
IdP	Identity Provider
SP	Service Provider
URN	Universal Resource Name
CSF	Credential Store Framework
KSS	Keystore Service
PII	Personally Identifiable Information
ILM	Information Lifecycle Management

Related Documents

You can see the following documents for more information:

Document Name	Description
Oracle Revenue Management and Billing Version 8.0.0.0.0 Release Notes	Lists the new features and enhancements made in this release. It contains high-level information about the supported platforms, framework upgrade, supported upgrades, unsupported integrations, and media pack downloads. It also lists the discontinued features, bug fixes, and known issues in this release.
Oracle Revenue Management and Billing Version 8.0.0.0.0 Release Overview Guide	Describes the new features, enhancements, user interface level changes, and database level changes made in this release. The new features and enhancements are classified under three categories – Generic (which is intended for both domains), Financial Services, and Insurance. It lists and describes enhancements made in the framework. It also encloses the ORMB and OUAF deprecation notices and enables you to access the product documentation.

Document Name	Description
Oracle Revenue Management and Billing Quick Installation Guide	Provides information about the media packs which are available for the current release. It contains information about the supported platforms, space requirements, and release-specific documentation library. In addition, it contains high-level information on how to install ORMB and selected additional software.
Oracle Revenue Management and Billing Installation Guide	Provides an overview about the application architecture. It contains detailed information about the software and hardware requirements, supported platforms, application server and database space requirements, and application server pre-requisites for supported platforms. It explains the installation and configuration menu options which are available during installation and advanced menu options which are available post installation. It also explains how to install OUAF, OUAF rollup packs, ORMB, and ORMB patches or rollup packs. It contains appendices which lists ORMB rollup packs and notices about third-party JAR and other components.
Oracle Revenue Management and Billing Database Administrator's Guide	Provides information about the supported database platforms and explains how to install database with or without demo data. It contains the standards and conventions that should be followed while working with ORMB database. In addition, it lists various configurations and implementation guidelines with respect to tablespace, encryption, storage, compression, indexes, initialization parameters, listener, table partitioning, performance monitoring, memory resource management, optimizer statistics and segment advisors, etc. It contains various appendices which lists the new objects added in ORMB, application services configured for default user groups, new objects added in OUAF, information about OUAF system tables and ORMB rollup packs, notices about third-party JAR and other components.

Document Name	Description
Oracle Revenue Management and Billing Server Administration Guide	Provides detailed information about the product architecture, native support, directory structures, environment variables, logs, etc. It explains various concepts related to the batch server and lists and describes the scripts required for installing and configuring ORMB. It explains how to monitor the system and configure Web browser, Web application server, business application server, inbound web services, and batch server. It explains how to provide Oracle Cloud Support and how to integrate and monitor Oracle Scheduler. It contains various appendices which lists and describes the parameters available in the ENVIRON.INI, Web.xml, spl.properties, hibernate.properties, submitbatch.properties, threadpoolworker, coherence-cache.config.xml, and tangosol-coherence-override.xml files.

You can also see the following artifacts on My Oracle Support using the respective document ID or part number:

- *Advanced Security Oracle Utilities Application Framework (Document ID: 1375615.1)*
- *Technical Best Practices For Oracle Utilities Application Framework Based Products (Document ID: 560367.1)*
- *Oracle Utilities Application FW - Batch Best Practices (Document ID: 836362.1)*
- *Database Vault Integration (Document ID: 1290700.1)*
- *Migrating from XAI to IWS (Document ID: 1644914.1)*
- *Oracle Utilities Application Framework - Keystore Configuration (Document ID: 2014161.1)*
- *Identity Federation SSO Fails With "Signature verification failed for provider ID ..." (Document ID: 2032605.1)*
- *Web Services Best Practices for Oracle Utilities Application Framework (Document ID: 2214375.1)*
- *Audit Vault Integration (Document ID: 1606764.1)*
- *Question: Does Weblogic CCB Cookies Contain Any Trace Information? (Document ID: 2443627.1)*

In addition, you can see the Oracle Database Advanced Security Guide 19c for more information using the following URL:

https://docs.oracle.com/en/database/oracle/oracle-database/19/asoag/index.html

Critical Patches

Oracle recommends that customers get all their security vulnerability information from **Oracle Critical Patch Update Advisories** available at *Critical Patches, Security Alerts and Bulletins*. It is strongly recommended that all critical patches should be applied in a timely manner.

For more information, see *Oracle Software Security Assurance*.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

1. What's New in Security? 1
   1.1 UI Masking for Numbers 1
2. Introducing Security 2
   2.1 Security Features 2
   2.2 Additional Security Resources 3
3. Authentication 4
   3.1 Online Authentication 4
   3.2 Batch Authentication 5
   3.3 Web Service Authentication 5
4. Authorization 6
   4.1 Authorization Model 6
      4.1.1 Users 6
      4.1.2 User Groups 7
      4.1.3 Access Groups 7
5. Managing Security 8
   5.1 Managing Online Users 8
   5.2 Managing Batch Users 9
   5.3 Managing Web Services Users 9
   5.4 User Authentication 10
6. Advanced Security 12
   6.1 Domain Authentication Group 12
   6.2 Logon Configuration 13
   6.3 Data Ownership Rules 14
   6.4 Configuring JMX Security 15
      6.4.1 Default Simple File-Based Security 15
      6.4.2 SSL-Based Security 16
      6.4.3 Other Security Sources 17
   6.5 Menu Security Guidelines 17
   6.6 Security Types 17
   6.7 Default Generic Application Services 17
   6.8 Administration Delegation 18
   6.9 Secure Communications (SSL) 19
   6.10 Data Masking Support 19
   6.11 Securing Files 19
   6.12 Password Management 20
   6.13 Securing Online Debug Mode 21

6.14 Securing Online Cache Management 21
6.15 Web Services Security 21
6.16 Message Driven Bean Security 23
6.17 SOAP Security 23
6.18 Groovy Support 24
6.19 Oracle Cloud Object Storage Support 24
6.20 HTTP Proxy Support 25
6.21 SYSUSER Account 25
6.22 Embedding User Experience 26
7. Audit Facilities 27
7.1 Audit Configuration 27
7.2 Read Auditing 28
7.3 Integrating to Audit Vault 29
8. Database Security 31
8.1 Database Users 31
8.2 Database Roles 32
8.3 Database Permissions 32
8.4 Using Transparent Data Encryption 32
8.5 Using Database Vault 32
9. Security Integration 33
9.1 LDAP Integration 33
9.2 Single Sign-On Integration 33
9.2.1 Kerberos Support 34
9.3 Oracle Identity Management Suite Integration 34
9.4 OAuth2 Support 35
10. Keystore and Truststore Support 36
10.1 Creating the Keystore and Truststore 36
10.2 Altering the Keystore/Truststore Options 38
10.3 Synchronize Data Encryption 38
10.4 Upgrading from Legacy to Keystore 40
10.5 Importing Keystores/Truststores 41
11. Encryption Feature Type 42
11.1 Configuration of Encrypted Fields 42
12. Web Services Security 43
12.1 Annotation Security 43
12.2 Oracle WebLogic WS-Policy Support 44
12.3 Oracle Web Services Manager Support 44
12.3.1 Access Control Support 44
12.4 Support for Multiple Policies 45

12.5 Importing Certificates for Inbound Web Services 45
13. Allow List Support 46
13.1 URL Allow List 46
13.1.1 Implementing a Custom URL Allow Lists 47
13.2 SQL Allow List 48
13.3 HTML Allow List 48
13.3.1 Implementing a Custom HTML Allow List 48
13.4 Groovy Allow List 48
14. Custom Authentication Service Provider 49
14.1 What does this Security Provider do? 49
14.2 Where would I use this Security Provider? 49
14.3 Implementing the Security Provider 50
15. Federated Security Support 51
15.1 Suggested References 51
15.2 Federated Architecture 52
15.3 Prerequisites for Federated Security 52
15.3.1 Process Flow 53
15.4 Federated Online Authentication 55
15.4.1 Overview 55
15.4.2 Identity Provider Configuration 56
15.4.3 Oracle HTTP Server/WebGate Configuration 57
15.4.4 Define Identity Provider Partner in Oracle Access Manager 58
15.4.5 Enable Just In Time Provisioning in Identity Federation 59
15.4.6 Define WebGate Agent 59
15.4.7 Copy WebGate Agent Configuration to OHS/WebGate 59
15.4.8 Define Authentication Policy for the Product Domain 59
15.4.9 Export the OAM SAML Metadata (optional) 60
15.4.10 Configure the Product Identity Asserter and Authenticators 60
15.4.11 Configure CLIENT-CERT 62
15.5 Federated Web Services 62
15.5.1 Overview 62
15.5.2 Process Flow 62
15.5.3 Setup OAuth Service 63
15.5.4 Configure WebGate for SOAP/REST Communications 63
15.5.5 Create OAuth Client 64
15.5.6 Using Keystores and Credentials 64
15.5.7 Enable OAuth on Product 68
15.5.8 Use Oracle Web Service Manager Policies 68
15.6 Federated Outbound Messages 69

15.6.1 Overview 69
15.6.2 OAuth Policies 70
15.6.3 Extendable Lookup Configuration........ 70
15.6.4 Message Sender Configuration 71
16. Securing JNDI Access 72
16.1 Securing Product Access 72
16.2 Providing Additional Access to the JNDI 73
17. Object Erasure Support 74
17.1 Configuration Of Object Erasure 74
18. Key Ring Support........ 75
18.1 Maintaining Key Rings 75
18.1.1 Generating Keys 75
18.2 Using Key Rings........ 75
19. Redaction Rules 76
20. Cookies Used by ORMB 77
20.1 Cookies Used by Network........ 77
20.2 Cookies Used by Cloud Services 78
21. Garbling Customer Information........ 79

1. What's New in Security?

The following new security feature is introduced with Oracle Utilities Application Framework Version 25.4:

- UI Masking for Numbers

1.1 UI Masking for Numbers

The system is enhanced to support masking numeric values on the user interface. This is useful if your implementation has certain quantities or amounts that are sensitive and should only be available unmasked to certain users.

The system provides a new masking algorithm type (**F1-MASKNBR**) for number masking. The following functionality is provided by this algorithm type:

- It uses the number 9 as its masking character.
- Like the existing string masking algorithm type, this new algorithm type includes configuration for an application service, security type, and authorization level allowing you to configure security for users that are allowed to see the data unmasked.
- For users that do not have the security to see data unmasked, the algorithm type returns a static number of 9s to mask both the digits themselves and the number of digits. See the algorithm type description for more information.

Masking for alphanumeric data, such as credit card numbers, bank account numbers, and personal identification numbers (for example, social security number) was already supported. This does not impact any existing extensions.

Note: To enable this feature, see the **User Interface Masking** section in the *Oracle Utilities Application Framework Administrative User Guide* for more information.

2. Introducing Security

One of the key aspects of Oracle Utilities Application Framework is security, which not only confirms the identity of an individual user, but also determines the data and functions that user has access to within the ORMB application. This section contains the following topics:

- Security Features
- Additional Security Resources

2.1 Security Features

Security is one of the key features of the Oracle Utilities Application Framework architecture, since it protects the access to your application, its functionality, and the underlying data stored and managed via Oracle Utilities Application Framework. From an architecture point of view, the following summarizes the approach to security:

- **Web Based Authentication** – Oracle Utilities Application Framework provides a default method, using a traditional challenge and response mechanism, to authenticate users.
- **Support for Oracle WebLogic Security** – Oracle WebLogic can integrate into several internal and external security stores to provide authentication services. Oracle Utilities Application Framework can use those configurations, to liaise via Oracle WebLogic, to authenticate users for online and Web Services based security.
- **Operating System Security** – For non-online and non-web service-based channels, Oracle Utilities Application Framework utilizes the operating system security (including any additional products used to enhance the base operating system security).
- **Non-Cookie Based Security** – After authentication the user's credentials form part of each transaction call to correctly identify the user to the internal authorization model to ensure the user is only performing permitted actions. This support is not browser cookie-based.
- **Secure Transport Support** – Transmission of data across the network can utilize the secure encryption methods supported for the infrastructure.
- **Inter-Component Security** - Calls within ORMB and across the tiers are subject to security controls to ensure only valid authenticated and authorized users using Java Authentication and Authorization Services (JAAS).
- **Inbuilt Authorization Model** – Once a user is authenticated, then the internal authorization model is used to determine the functions and data the user has access to within the ORMB application.
- **Native Web Services Security** – Web Services available from Oracle Utilities Application Framework are natively available from Oracle WebLogic. A wide range of security policies are available.
- **Keystore Support** - Keys for encryption can be externalized in JCEKS based keystore.
- **Integration with Other Security Products** – Implementation of security varies from customer to customer, so ORMB allows integration to other security products to offer enhanced security implementations, either directly or indirectly.

2.2 Additional Security Resources

In addition to the security resources described in this guide, Oracle Utilities Application Framework provides the following additional security resources:

- **Oracle Database Vault** – Oracle Database Vault provides fine-grained access control to your sensitive data, including protecting data from privileged users. *Oracle Database Vault Administrator's Guide* and *Database Vault Integration (Document ID: 1290700.1)* describes how to use Oracle Database Vault.
- **Oracle Audit Vault** – Oracle Audit Vault collects database audit data from sources such as Oracle Database audit trail tables, database operating system audit files, and database redo logs. Using Oracle Audit Vault, you can create alerts on suspicious activities, and create reports on the history of privileged user changes, schema modifications, and even data-level access. For more information, see the *Audit Vault Integration (Document ID: 1606764.1)* article *on* My Oracle Support and *Oracle Audit Vault Administrator's Guide* which explains how to administer Oracle Audit Vault.
- **Oracle Advanced Security** - See *Oracle Database Advanced Security Administrator's Guide* for information about advanced features such as transparent data encryption, wallet management, network encryption, and the RADIUS, Kerberos, Secure Sockets Layer authentication.
- **Oracle Identity Management Suite** – Oracle offers a range of specialist security products to manage user identities, password management, single sign-on, access management, identity governance, fraud detection and directory services. The *Oracle Identity Management Suite Administrator Guides* and the *Technical Best Practices For Oracle Utilities Application Framework Based Products (Document ID: 560367.1)* article *on* My Oracle Support provides additional information about these products and integration capabilities.

3. Authentication

From a security point of view authentication is about identification of the user. It is the first line of *defense* in any security solution. In simple terms, it can be as simple as the *challenge-response* mechanism we know as user ID and password. It can be also as complex as using digital certificates as the identification mechanism and numerous other schemes for user identification.

The authentication aspect of security for ORMB is delegated to the infrastructure used to run the product. This is due to the following:

- **Authentication Scheme Support** – Oracle WebLogic supports several industry standard security repositories and authentication methods. These can be native to Oracle WebLogic or additional products that can be integrated.
- **Enterprise Level Identity Management** – Identity Management is typically performed at an enterprise level rather than managed at an individual product level. ORMB is typically not the only application used at any site and therefore managing security across the enterprise is more efficient.

This section describes the following types of authentications:

- Online Authentication
- Batch Authentication
- Web Service Authentication

3.1 Online Authentication

Oracle Revenue Management and Billing delegates the responsibility of authentication of the online users to Oracle WebLogic. This means that any integration that Oracle WebLogic has with specific security protocols or security products can be used with the product for authentication purposes. The configuration of authentication is therefore performed within Oracle WebLogic itself.

Typically, Oracle WebLogic support one or more of the following:

- **Inbuilt Security** – Oracle WebLogic supplies a default basic security store and associated security management capability that can be used if no other security repository exists.
- **LDAP Based Security** – The Lightweight Directory Access Protocol (LDAP) is a protocol for accessing and maintaining distributed directory information services. LDAP is used to standardize the interface to common security repositories (such as Oracle Internet Directory, Microsoft Active Directory etc.). LDAP support may be direct or indirect via Identity Management software like Oracle Virtual Directory or Oracle Identity Federation.
- **SAML Based Security** – Security Assertion Markup Language (SAML) is an XML based data format for exchanging authentication and authorization information between parties.
- **DBMS Based Security** – Oracle WebLogic can store, manage, and retrieve security information directly from a database.
- **Operating System Based Security** - Oracle WebLogic can store, manage, and retrieve security information directly from the underlying operating system.

- **Oracle Utilities Application Framework Security Provider** – The Oracle Utilities Application Framework includes an optional custom WebLogic Security Provider that allows implementations to verify user identity and whether the user is enabled as part of a security domain configuration. If the security provider is not used, these attributes are checked at login time.

These security configurations can be natively support or can be augmented with additional products. See the *Oracle WebLogic Security Guide* for more information about the security configuration process.

3.2 Batch Authentication

The Batch component of the architecture utilizes the operating system-based security (including any extensions to that security) to authenticate users to execute batch processes. From an authentication point of view:

- Batch users must be defined in the operating system and associated with the operating system security group assigned at product installation time. This ensures users have appropriate access to product resources and the ability to write logs.
- Threadpools can be started by any valid operating system user, but ideally threadpools and submitters should be executed by the same operating system user.
- Before any threadpool or submitter is executed, the user must execute the splenviron utility to set the environment variables for the ORMB application correctly. This can be done at the command line for each threadpool and submitter or globally using the logon profile for the operating system user.

3.3 Web Service Authentication

The Web Service component of ORMB is housed in Oracle WebLogic and utilizes the native Web Services security mechanism supported by that server.

From an authentication point of view:

- The Web Service is deployed using an administration account using the utilities provided from the product online (for developers) or using command line utilities.
- The Web Service is managed using the administration account using the administration console provided with Oracle WebLogic.
- Oracle WebLogic allows security policies and security access rules to be configured at an individual Web Service point of view. Any of the valid policies and security rules supported by Oracle WebLogic can be used.
- Web Service management products such as Oracle Web Services Manager can be used to augment security for Inbound Web Services.

4. Authorization

Once a user is identified, they must be authorized to specific functions and data within the ORMB application. Oracle Utilities Application Framework uses an inbuilt security model for authorization. This model contains all the data necessary for the definition of authorizations to function and data.

4.1 Authorization Model

The following data model describes the security authorization model of Oracle Utilities Application Framework:

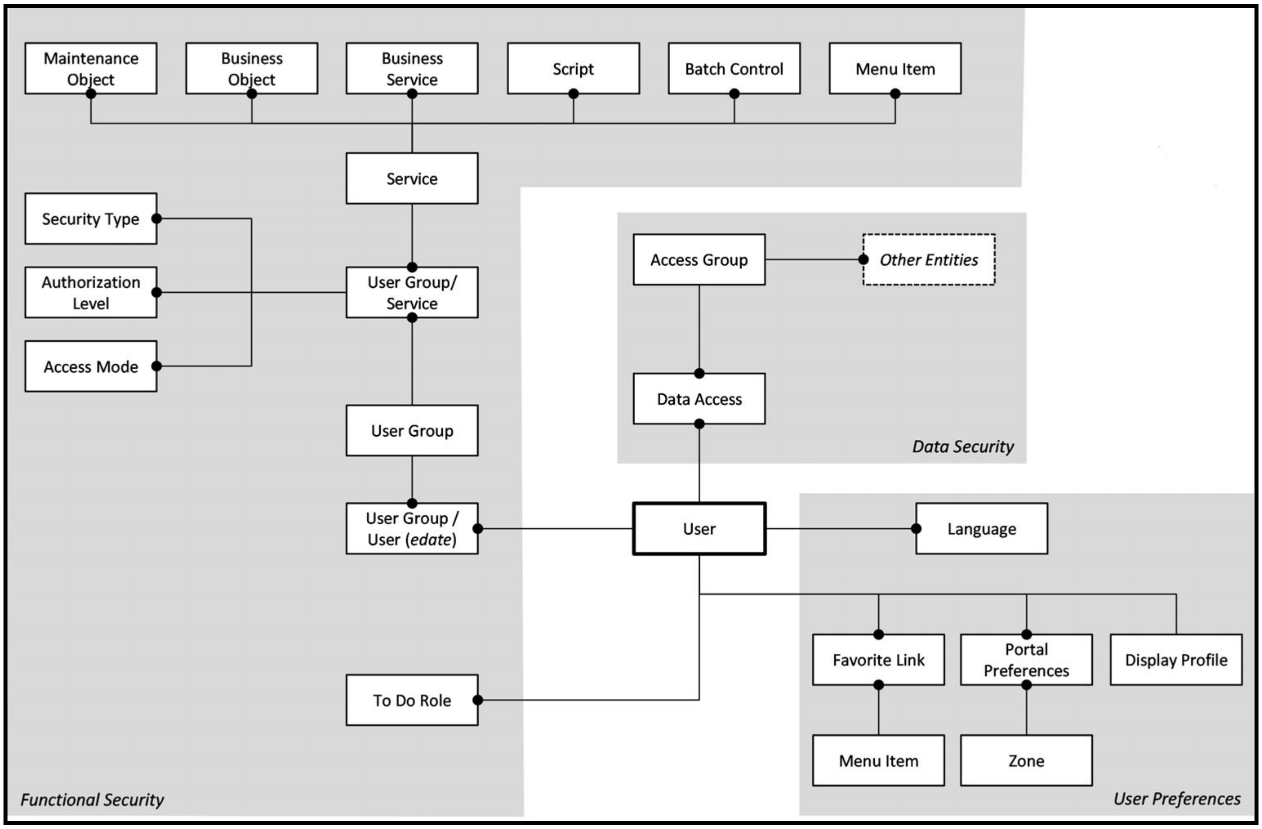


Figure 1: Security Authorization Model

4.1.1 Users

A record of each user is stored in the User entity, which defines the attributes of the user including identifier, name, portal preferences, favorites, display profile (such as format of dates and so on), and language used for screens and messages, and other attributes. Users are attached to the To Do roles that allow the user to process any error records for background processes. For example, if the XXX background process produces an error, it is possible to define the users that will process and address it.

4.1.2 User Groups

User groups are a mechanism for grouping users, usually around job roles. Users are attached to the user groups through a relationship that is effective dated. This can be useful for attaching the temporary employees such as contractors or for people who change roles regularly.

Each user group is authorized to access certain application services, which are the functions within the ORMB application. Loosely, these correspond to each of the accessible screens. The application services have valid access modes, such as Add, Modify, Read, and Delete.

Additionally, it is possible to define authorization levels to functions for the user groups. For example, you may find that a certain group of users can only approve payments of a certain level unless additional authorization is obtained. The authorization level is associated with a security type, which defines the rules for a given application service.

Note: To use security types, the implementation must develop server-side or client-side user exits to design the required code to implement the security level.

Services can be attached to individual Menus, Batch Controls, Maintenance Objects, Business Objects, Business Services and Scripts to denote the service to be used to link user groups to access these objects. In this case Business Object security overrides any Maintenance Object security. The same applies to Business Services security overriding the Application Service it is based upon.

Oracle Utilities Application Framework allows you to limit a user's access to specific data entities to prevent users without appropriate rights from accessing specific data. By granting the user access rights to an account, you are granting the user access rights to the account’s bills, payment, adjustments, orders, and so on.

4.1.3 Access Groups

An access group defines a group of accounts that have the same type of security restrictions. A data access role defines a group of users that have the same access rights (in respect of access to entities that include access roles). When you grant the data access role rights to an access group, you are giving all users in the data access role rights to all entities in the access group.

The following points summarize the data relationships involved with data security:

- Entities reference a single access group. An access group may be linked to an unlimited number of relevant entities.
- A data access role has one or more users associated with it. A user may belong to many data access roles.
- A data access role may be linked to one or more access group. An access group may be linked to one or more data access roles.
- Information in the security model can be manually entered using online transactions and can be imported and synchronized using a LDAP import function provided with the Web Services Adapter. The latter is typically used with customers who have lots of online users to manage.
- The authorization model is used by all modes of access to the product. Native interfaces (java classes) are used by all objects and a PL/SQL procedure is provided for reporting interfaces.

5. Managing Security

Once the security definitions are established, they must be managed from the application itself, using the security infrastructure and security repositories. This section contains the following topics:

- Managing Online Users
- Managing Batch Users
- Managing Web Services Users
- User Authentication

5.1 Managing Online Users

To manage online users, several facilities must be configured:

- The security repository and rules must be configured in Oracle WebLogic to enable authentication. See the *Oracle WebLogic Administration Guide* for more information.
- The product group used to connect users to the Oracle WebLogic resources should be created in the security repository and configured in the product configuration. The default value for this setting is `cisusers`. See the *Oracle Revenue Management and Billing Server Administration Guide* for more information on this setting.
- Users need to be connected to the product group within the security repository to indicate that they can access the Oracle WebLogic resources.

The process for managing online users is outlined in the following figure:

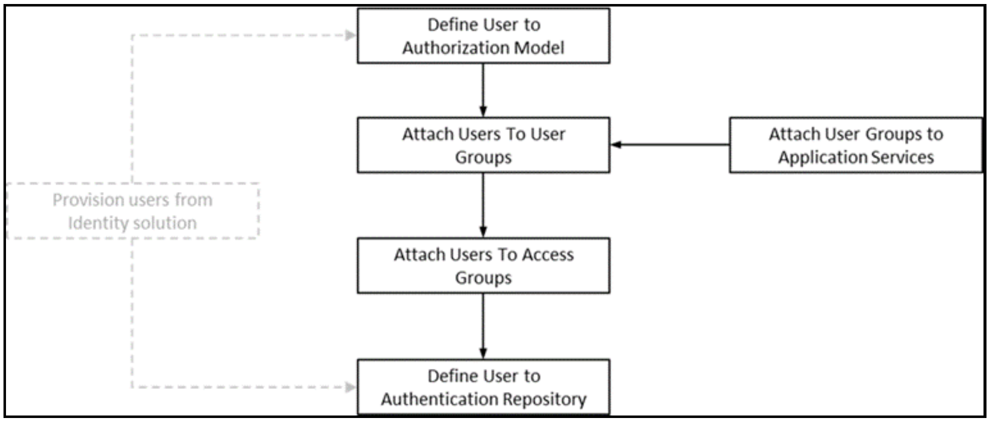

Define User to Authorization Model
Attach Users To User Groups
Attach User Groups to Application Services
Provision users from Identity solution
Attach Users To Access Groups
Define User to Authentication Repository

Figure 2: Process for Managing Online Users

Users should be defined to the authorization model to define their profile and permissions within the ORMB application. For information about defining users and user permissions, see the *Oracle Utilities Application Framework Administrative Guide*.

5.2 Managing Batch Users

Each time a batch process is executed, the security components of the ORMB application must authenticate the user against a security repository and authorize the user to access the components that the batch process needs to complete its operations.

The batch component of the architecture uses the following security mechanisms:

Security Mechanism	Description
Authentication	The batch users must be defined in the operating system configured security repository and must be a member of the operating system group assigned to the ORMB application. **Note:** The User ID does not have to match the authorization user.
Authorization	The authorization user is defined within the ORMB application as the online user and is specified as a job parameter at execution time or in configuration files supplied for the batch process. See the *Oracle Revenue Management and Billing Server Administration Guide* for more information about the parameters used for batch processing.

To manage batch users, the following is recommended:

- Add the authentication user used to initiate the thread pool and submitter processes for a batch process to the configured operating system repository.
- Specify a valid user authorization identifier as a parameter for the batch process. This identifier must be authorized to the valid actions against the main objects used in the batch process. See the ORMB documentation about the objects used in each of the batch processes. By default, the user used for authorization is the same as the submission user. The authorization user may be overridden to alternative values if the submission user has the User Override permission for batch job submission.

5.3 Managing Web Services Users

From a product perspective, a Web Service is a channel into the objects within the ORMB application. Any of the objects, services, and scripts available in the product can be exposed as JAX-WS 2.0 based Web Service. From a security perspective, Web Services uses the following security mechanisms:

Security Mechanism	Description
Authentication	The Web Services component of the ORMB application uses the Web Services support native to Oracle WebLogic. This allows security tokens supporting many standards to be used to authenticate individual web service calls.

Security Mechanism	Description
Authorization	The Web Services component uses the same authorization model as that of the online user and batch component. **Note:** The authorization user within the User object is mapped through the authentication user in the same way as the online users are mapped.

To manage Web Services security users the following is recommended:

- Users for authentication are added to the security repository configured with Oracle WebLogic. This should match the Login ID used for the authorization model.
- Security Policies need to be attached to Web Services using Oracle WebLogic. For Oracle WebLogic, the security policies available using Oracle Web Services Manager are available for use with individual Web Services. Multiple policies are supported. See *Securing WebLogic Web Services for Oracle WebLogic Server* for more information and the policies available.
- Users must be defined to the authorization model with appropriate access to underlying services used by the Web Service. For Web Services based upon business objects, business services and scripts, users need appropriate access to the application service defined on these objects.
- Transaction Types in the Web Services translate to Access Modes within the Application Service calls.

For more information about Inbound Web Services, see the *Web Services Best Practices (Document ID: 2214375.1) article on* My Oracle Support.

5.4 User Authentication

There are two different user identifiers, namely User ID and Login ID. The different identifiers have distinct roles:

Identifier	Description
User ID	The User ID, which is up to 8 characters in length, is used internally for authorization and is passed as the CLIENT_IDENTIFIER on the database connection. This user cannot be changed after the user has created any records in the system as it is used for record ownership in some objects and in auditing.
Login ID	The Login ID, which is up to 256 characters in length, is used for authentication to the security repository configured on the Oracle WebLogic domain. The Login ID can be same as the User ID or different to reflect site standards. Unlike the User ID, the Login ID can be changed at any time to reflect changes in the organization, such as name changes or acquisition. **Note:** The Login ID must match, in the same case, as the entry in the configured security repository for Oracle WebLogic.

When maintaining a user, it is important that the Login ID is only changed using the maintenance function, LDAP Import or any Inbound Web Service based upon the user object and not directly using other means (such as direct SQL). This is because a security hash is generated at maintenance time and is checked during login. If the security hash does not match while login, the user is not authorized to access to the ORMB application. To ensure security hashes are correct, use the Synchronize Data Encryption function to reset the user security hash.

6. Advanced Security

While the default security settings are adequate for most sites, there are several additional advanced settings that can be configured to support a wider range of security requirements. This section outlines the various advanced security settings that you can configure for the ORMB application:

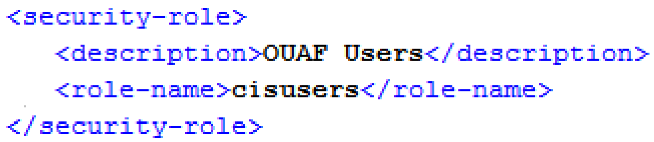
- Domain Authentication Group
- Logon Configuration
- Data Ownership Rules
- Configuring JMX Security
- Menu Security Guidelines
- Security Types
- Default Generic Application Services
- Administration Delegation
- Secure Communications (SSL)
- Data Masking Support
- Securing Files
- Password Management
- Securing Online Debug Mode
- Securing Online Cache Management
- Web Services Security
- Message Driven Bean Security
- SOAP Security
- Groovy Support
- Oracle Cloud Object Storage Support
- HTTP Proxy Support
- SYSUSER Account
- Embedding User Experience

6.1 Domain Authentication Group

The default installation of ORMB includes a default authentication group (role-name) defined within the Web Application descriptor (web.xml). This role name is used by the Web Application to link the authorized users within the ORMB application to the associated domain physical resources (pages and configuration files) within Oracle WebLogic. The specification of the group in the web descriptor is in the security section.

Note: The security role is used in several sections of the Web Application descriptor.

For example:



```
<security-role>
   <description>OUAF Users</description>
   <role-name>cisusers</role-name>
</security-role>
```

By default, this group is set to cisusers, which is configurable for each web component. When the product is deployed to Oracle WebLogic, this group is instantiated ready to be allocated to individual users. Users of the product must be attached to this group to use it.

From a configuration point of view there are several options for this setting:

- The default group may be changed at installation and configuration time using the configuration settings as shown below as outlined in the *Oracle Revenue Management and Billing Server Administration Guide*. The group name should have no embedded blanks.

Component	Principal Name	Role Name
Online Help	WEB_PRINCIPAL_NAME	WEB_ROLE_NAME

- If Oracle WebLogic is configured to use an external security repository the configured administration group must exist in the security repository and the users must be connected to this group.

Note: If the domain administration group is changed after the installation, you need to migrate the users to the new domain administration group either manually or using the tools provided with the security repository or through Oracle WebLogic.

6.2 Logon Configuration

The default configuration for online authentication is using a logon screen for the online product and online help application. The product supplies a prebuilt logon screen for all three components preconfigured.

At logon it detects that a user has not logged on before (the presence of a JSESSIONID cryptographically-secure session cookie issued by the Web Application Server is used). Depending on the configuration (in the web.xml) of the applications, housed in Oracle WebLogic, the following is performed:

- **FORM** – This is the default setting to support a logon screen with an associated error screen in case of unsuccessful logon. ORMB provides a prebuilt logon screen but can be replaced with custom logon screens by setting the following configuration settings appropriately for each web component as outlined in the *Oracle Revenue Management and Billing Server Administration Guide*:

Component	Login Screen	Login Error Screen
Online	WEB_FORM_LOGIN_PAGE	WEB_FORM_LOGIN_ERROR_PAGE

Note: Custom logon screens should be placed in the **cm** directory of the Web Application Server as outlined in the *Oracle Utilities Application Framework Software Development Kit*.

- **BASIC** – The browser will issue a call to the operating system to display the default logon dialog supplied with the operating system. No logon dialog is supplied.

Note: BASIC authentication is considered a relatively weak authentication scheme, and therefore is not recommended for use.

- **CLIENT-CERT** - This is an advanced configuration to allow for certificated (one way or two way) to be used. See the Administering Security for Oracle WebLogic Server documentation for more details on the additional configuration required.

6.3 Data Ownership Rules

On each of the objects (and on selected child objects), an owner flag is included to determine the origin of the data. The owner flag is used by ORMB to determine the maintenance owner of key data as well as to protect from accidental deletion of important data shipped with the product.

The value of the flag is displayed on maintenance screens to visually indicate the data owner. The location of the information varies from the top left of maintenance pages, within lists of information (to apply to individual rows) and within sections of maintenance pages.

The data ownership flag has the following values:

Value	Description
Base	This is crucial information shipped with ORMB and cannot be deleted or modified through the **Delete** or **Medication** functions regardless of user permissions. This value is reserved for use by ORMB to ship and protect key information. **Note**: Deletion of this information directly from a product database will cause unexpected results.
Product Name	The name of the product that owns the data. This value is like the Base value but indicates the component for which the data is applicable. All the rules that apply with the Base value apply to this value.
Customer Modification	This indicates that the data was added by the implementation using the various methods and that it is owned by the implementation. **Points to Note**: Deletion of data is permitted using the valid deletion functions for authorized users. In general, sites can only maintain Customer Modification owned records. Other ownership values are reserved to protect product installation supplied data.

6.4 Configuring JMX Security

The operations interface is based upon Java Management Extensions (JMX), which allows components of the application to be managed and monitored from JSR160 compliant consoles including jconsole or Oracle Enterprise Manager.

See the *Oracle Revenue Management and Billing Server Administration Guide* for more details of the JMX operations interface.

By default, the JMX implementation and configuration uses the default simple file based security as outlined in the JMX Specification.

6.4.1 Default Simple File-Based Security

The default configuration is based upon a properties file containing name/value pairs corresponding to role/password pairs and authorization can be also based on a properties file containing name/value pairs corresponding to role/access pairs where access can be any of readonly access which grants read access to any remote operation and readwrite access which grants access to read and update operations in the interface.

Note: By default, the user (BSN_JMX_SYSUSER) and password (BSN_JMX_SYSPASS) for the administrator are automatically added to the configuration files.

To use this facility, the following file should be maintained using an appropriate editor located in $SPLBASE/scripts directory:

- ouaf.jmx.access.file – This file contains the user ID and access permissions in the format separated by a blank space:

Field	Comments
User ID	Authentication user to access JMX.
Permission	Permission assigned to user. Valid values are: • readonly – No update access • readwrite – Update and update operations access

- ouaf.jmx.password.file - This file contains the user ID and password in the format separated by a blank space:

Field	Comments
User ID	Authentication user to access JMX
Password	Password in plain text or encrypted format.

Note: These files are also tailored using custom templates. The ouaf.jmx.access.file.template and ouaf.jmx.password.file.template are used for the configuration.

6.4.2 SSL-Based Security

To secure communications for JMX using the Java SSL support, the following process must be performed:

- Security must be setup using the Default Simple File-Based Security or Other Security Sources.
- A key pair and certificate need to be setup on your server. See the Monitoring and Management Using JMX Technology or Oracle WebLogic Administration documentation for details and utilities available for this process.
- Set additional java parameters using the **`WEB_ADDITIONAL_OPT`** for the online/Web Services and **`BATCH_MEMORY_ADDITIONAL_OPT`** for Batch. See the *Oracle Revenue Management and Billing Server Administration Guide* for details of these parameters. The following additional system properties must be set:

System Property	Comments
javax.net.ssl.keyStore	Keystore location
javax.net.ssl.keyStoreType	Default keystore type
javax.net.ssl.keyStorePassword	Default keystore password
javax.net.ssl.trustStore	Truststore location
javax.net.ssl.trustStoreType	Default truststore type
javax.net.ssl.trustStorePassword	Default truststore password
com.sun.management.jmxremote.ssl	Set to true
com.sun.management.jmxremote.registry.ssl	Set to true
com.sun.management.jmxremote.ssl.need.client.auth	Set to true

Points to Note:

For a full description of additional options and SSL setup, see Monitoring and Management Using JMX Technology.

Specification of system properties for java is as per the java command-line.

For sites using Oracle WebLogic in native mode, configuration of SSL requires Configuring SSL in WebLogic Server and altering the startup scripts for Oracle WebLogic to include the above options.

In line with industry standards, either HTTP or HTTPS can be used. They cannot be used simultaneously.

6.4.3 Other Security Sources

Whilst, by default, the file-based repository is supported, it is possible to configure the authentication of JMX to use an alternative data source such as an LDAP Server. This involves changing the Java Authentication and Authorization Service (JAAS) configuration stored in the java.login.config file of the $SPLEBASE/splapp/config directory.

In the JAAS configuration file there is a default jmxrealm that contains the default JMX LoginModule. This can be changed, using custom templates, to support an alternative source for authentication. See the LdapLoginModule documentation for information and examples of login configurations.

Note: To implement the custom security source custom templates for java.login.config must be implemented according to the process outlined in the *Oracle Revenue Management and Billing Server Administration Guide*.

6.5 Menu Security Guidelines

By default, a menu option is displayed whenever a user has access to the underlying application service definition attached to objects that are indirectly linked to a menu entry. Whilst this behavior is enough for most needs, it is possible to place an override on an individual menu item to override the lower level security levels. This is particularly useful where implementations wish to replace base supplied menu items with custom menu items.

By linking a menu item to a new service that can reference the underlying objects and specifying an Application Service (optionally also including an Access Mode) would override the permissions on the underlying objects.

It is possible to specify the application service for a menu item on the **Application Security** tab of the **Menu** screen.

6.6 Security Types

For information and details regarding security types, see the *Oracle Utilities Application Framework Administrative Guide*.

6.7 Default Generic Application Services

By default, a whole set of application services are defined against base functions. In line with data ownership rules, some of these records can be altered and new functions added. A set of generic application services are also shipped with the ORMB application to provide a mechanism for defining new zones, new objects, or new menu items for rapid deployment.

The following table lists and describes a generic application service:

Application Service	Description
F1-DFLTS	This is a generic maintenance application service which is designed to secure business objects, zones, and menu items. It supports the Add, Modify, Delete, Inquire, and Execute access methods.

Note: The use of this generic application service is optional.

6.8 Administration Delegation

By default, ORMB provides a single administration account, as configured in the SPLADMIN configuration setting, in the ENVIRON.INI configuration file, to manage the operational aspects of the product. This operating system user is the owner of the product when it is installed and is typically used for all operational aspects of the product.

Note: It is not possible to change the product administration account after installation. If this is desired, it is recommended to remove the product and reinstall using the alternative administration account.

Whilst the single administration account is enough for most needs it is possible to provide additional administration accounts to delegate administration tasks. To delegate administration, the following must be configured:

- Any administration user must be a member of the operating system group allocated to the product as outlined in the SPLADMINGROUP configuration setting in the ENVIRON.INI configuration file.
- If you are using Oracle WebLogic in native mode, then the console will execute the native facilities to start and stop the product. It is recommended that the user allocated to Oracle WebLogic at installation time be a member of the operating system group outlined in SPLADMINGROUP configuration setting in the ENVIRON.INI configuration file.

Note: Customers using Oracle Enterprise Manager, with or without Application Management packs, should use the administration delegation and credential management capabilities of that product to manage administration delegations.

6.9 Secure Communications (SSL)

By default, ORMB uses HTTPS to communicate to the browser and across the tiers. The transport protocol can be encrypted using SSL/TLS to secure transmission of data across networks.

Note: Oracle strongly recommends that customers use SSL to secure transmission for production environments.

To implement SSL the following process must be completed:

- Configure Oracle WebLogic to use the SSL protocol. For Oracle WebLogic, see the Configuring SSL section in Oracle Fusion Middleware Securing Oracle WebLogic Server.
- Set the SSL Port Number using the WEB_WLSSLPORT configuration parameter as outlined in the *Oracle Revenue Management and Billing Server Administration Guide*.
- Once the setup has been tested and verified, see the console documentation on disabling insecure protocols.

6.10 Data Masking Support

For information about data masking, see the *Oracle Utilities Application Framework Administrative Guide*.

6.11 Securing Files

Note: The utilities mentioned in this section only apply to Linux environment.

The file structure of ORMB is protected by permissions set at the operating system level. By default, the settings provided upon installation comply with Oracle standards in respect to permissions. For more details of the individual user permissions on product directories and subdirectories, see the *Oracle Revenue Management and Billing Server Administration Guide*.

You can use the following process to reset the environment permissions to the default settings:

- Execute the splenviron.sh utility to set the environment variables for the product environment to reset. See the *Oracle Revenue Management and Billing Server Administration Guide* for details of this process.
- Execute the setpermissions.sh utility to reset the environment permissions back to the defaults.

6.12 Password Management

Oracle Revenue Management and Billing requires several passwords, which must be changed on a regular basis to conform with security best practices:

Password Owner	Location	Comments
Online User	Domain Authentication Source	No configuration changes. User changes password in security repository directly or indirectly using security products. Security repository is configured in Oracle WebLogic. **Note: `WEB_SPLPASS`** specifies the default password for the initial user. If this user is used past the installation, the password may need to be changed. See the *Oracle Revenue Management and Billing Server Administration Guide* for more details.
Web Service User	Domain Authentication Source	No configuration changes. User changes password in security repository directly or indirectly using security products. Security repository is configured in Oracle WebLogic.
Batch User	Operating System	No configuration changes. User changes password in security repository directly or indirectly using security products. Security repository is configured in Oracle WebLogic.
Database Users	BATCH_DBPASS DBPASS XAI_DBPASS	The database users are stored in ENVIRON.INI. See the *Oracle Revenue Management and Billing Server Administration Guide* to understand how to change values. New passwords need to be re-encrypted.
JMX Users	BSN_JMX_SYSPASS	The default JMX user is stored in ENVIRON.INI. See the *Oracle Revenue Management and Billing Server Administration Guide* to understand how to change values. New passwords need to be re-encrypted.

Password Owner	Location	Comments
Administration Account	WLS_WEB_WLSYSPASS WEB_WLSYSPASS	The default administration users are stored in ENVIRON.INI. See the *Oracle Revenue Management and Billing Server Administration Guide* to understand how to change values. New passwords need to be re-encrypted.

6.13 Securing Online Debug Mode

Oracle Revenue Management and Billing features an online debug mode which is used for problem solving and development personnel to trace their code or to diagnose problems. Along with other functions within the product, the debug function is security-controlled.

To use this facility for any of the user groups, an individual user must include Inquire access to the F1DEBUG application service. This will enable the debug facility from the URL.

For more information about the Debug facility see the *Oracle Revenue Management and Billing Server Administration Guide*.

6.14 Securing Online Cache Management

The online cache management function resets the online cache to force new values to be loaded. Along with other features within the product, it is security controlled.

To use this facility for any of the user groups, you must include Change access to the F1ADMIN application service. This will enable the cache management facility from the URL.

For more information about the cache management facility see the *Oracle Revenue Management and Billing Server Administration Guide*.

6.15 Web Services Security

Points to Note:

This section outlines the Inbound Web Services security facility only.

See the *Migrating from XAI to IWS (Document ID: 1644914.1)* article for more information.

Inbound Web Services allow external web service based integrations to access functionality within the ORMB application. The security settings for the Inbound Web Services can be summarized as follows:

- Inbound Web Services rely on Web Services standards supported by Oracle WebLogic for authentication support.
- Inbound Web Services support the WS-Policy standards supported by Oracle WebLogic to provide both transport and message security. See the Oracle WebLogic documentation for details of the WS-Policies supported. The following rules apply to those policies:
    - Oracle WebLogic policies are supported if the corresponding setup is performed within Oracle WebLogic. For example, encryption is supported if keystores are configured for encryption keys.

  - WS-Policies are attached within the Oracle WebLogic console or Oracle Fusion Middleware Control after deployment. These policies are maintained independently as per the console documentation.
  - Element Level policies are not supported in the current release.
  - Security policies at the operation level are not supported directly but are supported via authorization.
  - The product ships an internal policy for backward compatibility (UserToken).
- Inbound Web Services use the underlying business objects, maintenance objects, business services and service scripts to determine authorization of records. This includes authorization for specific operations.
- Inbound Web Services can use Oracle Web Service Manager for additional WS-Policy support and web service access controls.
- Security policies can vary between individual Inbound Web Services.
- Multiple WS-Policies are supported per Web Service. The clients calling these services must conform to at least one of the policies attached.

By default, the WS-Client calling the product must supply an authentication token in the format configured on the WS-Policy of individual web service. By default, there is no default user on Inbound Web Services transactions. A default user may be configured on the ouaf.ws.defaultUser setting in the spl.properties file for the Inbound Web Services. See the *Oracle Revenue Management and Billing Server Administration Guide* for details of the process.

Note: Setting of a default user is not recommended for implementations unless backward compatibility is required for older XML Application Interface-based services.

For backward compatibility, there are several additional settings that cover Inbound Web Services:

Setting	Comments
ouaf.ws.defaultUser	Default user for authorization of Web Services calls
ouaf.ws.superusers	Delimited set of effective users used to translate calls from authentication users not known to the system.
ouaf.ws.deploy.user	Administration user for deployment activities. This setting is only specified if it differs from the administration settings.

6.16 Message Driven Bean Security

Note: See the *Oracle WebLogic JMS* documentation for detailed information about JMS facilities provided.

See the *Web Services Best Practices for Oracle Utilities Application Framework (Document ID: 2214375.1)* article on My Oracle Support for more information about the JMS integration implementation.

The Message Driven Bean (MDB) within the Inbound Web Services implementation allows JMS resources (such as JMS Queues or JMS Topics) to be read using the MDB and sent to an Inbound Web Service to be processed.

By default, the Message Driven Bean uses the JMS (JMSX) message property fields for authentication and authorization purposes such as JMSXUserid.

If the JMS message security is not used, then a default user can be set in the ouaf.ws.defaultUser parameter in the spl.properties file.

6.17 SOAP Security

In this release, additional SOAP Header security, for outbound communications, has been added to support additional facilities in Service Oriented Architecture integrations and Oracle Web Services Manager. The following additional facilities are now supported in the SOAP Header:

Facility	Description
SOAP Insert Time Stamp	A set of timestamps can be added to the transaction to support WS-Security to avoid replay attacks.
Additional SOAP Security Inbuilt	The SOAP Header can now have inbuilt support for TEXT and DIGEST headers in addition to the BASIC support provided in past releases. This feature can be replaced using Oracle Web Services Manager with WS-Policy support for other advanced security configurations.
SOAP Expiration Delay	It is possible to set a transaction expiration, in seconds, to control resource usage of the transaction.

Note: The only HTTP/HTTPS method supported in this release is POST.

See the online documentation for a more detailed description of these settings.

6.18 Groovy Support

Oracle Revenue Management and Billing supports Groovy for extensions, via the script engine. This support was added to augment the Java and Scripting support to offer an alternative. The implementation of Groovy has some limitations for security reasons:

- Groovy APIs that have direct access to operating system functions have been blocklisted for security reasons and therefore cannot be used. Alternative functions are provided to offer safe access to selected operating system functions.
- It is possible to implement a custom allow list for non-cloud implementations. See the *Oracle Revenue Management and Billing Server Administration Guide* for more information.

See the online documentation for more details of Groovy support.

6.19 Oracle Cloud Object Storage Support

Note: Prior to using this capability, the Oracle Cloud Object Storage Service must be purchased and configured. Networking between on-premises or other cloud services must be installed, configured and operational before using this facility.

By default, use of FILE-PATH batch variable was restricted to local mounted storage where it is possible to use network storage through mapped directories. It is now possible to use Oracle Cloud Object Storage Service as a source of import files or locations to write files. To use this feature, Oracle recommends the following:

- Create or edit a lookup value for the F1-FileStorage extendable lookup for each cloud service using the following connection details:

Connection Details	Comments
File Adapter	Use *Oracle Cloud Object Storage*
Tenancy	OCI Tenancy for Oracle Cloud Object Storage
Compartment	Object Storage Compartment
Namespace	Object Storage Namespace
Key Ring	Key associated with Oracle Cloud Object Storage
Region	Oracle Cloud Infrastructure Region
Bucket Name Prefix	Prefix to use for Bucket.
Reporting Configuration	Whether this setting is for BI Publisher files (Oracle Revenue Management and Billing Cloud Services use only).

- To use the definition, the parameter should be used in the FILE-PATH variable, in either the Batch Control definition or batch configuration file for relevant batch controls in the format:

  file-storage://<ExtendableLookupValue>

  Where:

  - <ExtendableLookupValue> - Lookup value of the F1-FileStorage extendable lookup.

Note: Additional settings may be added to FILE-PATH to support virtual folders, if necessary.

6.20 HTTP Proxy Support

If HTTP Proxies are used for networking these can be configured at the JVM level for all JVMS using the Java Networking and Proxy settings. These settings can be set in the following areas:

- For online, Inbound Web Services, REST and outbound messages, the settings may be specified on the Oracle WebLogic Server settings or using the GLOBAL_JVMARGS configuration parameter.
- For Batch in Oracle Coherence, the settings may be specified using the GLOBAL_JVMARGS configuration parameter.

See the *Oracle Revenue Management and Billing Server Administration Guide* for additional information.

6.21 SYSUSER Account

By default, an account named SYSUSER is created during the ORMB installation. This account is defined by default in the default security realm in the provided templates, is provided as the initial User object in the authorization model and is used as the default user in some transactions.

You cannot physically remove the SYSUSER account as it is used by the initial installation and owned by ORMB, but it can be disabled under the following conditions:

- Alternative identities have been configured for the authentication and authorization components of the product.
- Every facility in the implementation that uses the SYSUSER account as the default identity has been changed to an alternative to prevent misconfiguration of the facility.

Note: Oracle recommends that you use the appropriate alternatives for transactions instead of the SYSUSER account.

The following facilities use SYSUSER as the default identity, if used:

Facility	Comments
Default User in Message Option	This is used for XAI and MPL (older releases only).
HTTP Login User context type on *Message Sender*	Used for Utilities P2P integrations. Ensure HTTP User Password is also set correctly for alternative.
WEB_IWS_SUPER_USERS in ENVIRON.INI file	Used for supporting IWS user proxy connections.
WEB_IWS_MDB_RUNAS_USER in	Used to support user proxy connections in Message Driven

Facility	Comments
ENVIRON.INI file	Bus.
Batch Controls	Replace SYSUSER as the user used for submission of any batch controls in batch control configuration files, batch edit configuration files or in the Oracle Scheduler configuration.

You can deactivate the SYSUSER account by:

- Removing SYSUSER from configured security realm for authentication, preventing the user from authenticating.
- Setting the User Enable attribute of the SYSUSER user to Disable. This will deactivate the account from any unauthorized activity in ORMB.

6.22 Embedding User Experience

Oracle Revenue Management and Billing supports embedding application and being embedded in applications. To use this facility, several settings must be configured appropriately including:

- Set embedded behavior using the following values in the spl.properties file:

Setting	Comments
com.oracle.ouaf.web.csp.enable	Set to true
com.oracle.ouaf.web.allowCORS	Set to true
com.oracle.ouaf.web.csp.allowedFrameAnsMaxNumber	Set to 2 or more
com.oracle.ouaf.web.disableSecureCookie	Set to false
com.oracle.ouaf.uriSubstitutionVariables.file	Set to the location of substitution file

- Set the URI substitutions using the substitution variables for your embedded or embedding application to avoid hardcoding in the com.oracle.ouaf.uriSubstitutionVariables.file file. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<substitutionVariables>
…
<uriVariable>
<name>CSP_FRAME_ANS_HOST1</name>
<value>https://[FRAMING-HOST-URL]</value>
</uriVariable>
<uriVariable>
<name>CSP_FRAME_ANS_HOST2</name>
<value>https://[FRAMING-HOST-URL]</value>
</uriVariable>
</substitutionVariables>
```

7. Audit Facilities

Oracle Revenue Management and Billing has an inbuilt configurable auditing facility that provides the capability to register accesses to data from online and Web Services users. Batch processing is not audited by default but can be enabled using the *Oracle Utilities Application Framework Software Development Kit* using programmatic methods.

Auditing allows for the configurable tracking of changes to key data and allows authorized users to track changes on an individual user. Use of this facility is optional and can be switched on or off at any time. This section contains the following topics:

- Audit Configuration
- Read Auditing
- Integrating to Audit Vault

7.1 Audit Configuration

Note: This section covers the soft-table implementation of auditing. There is a specialist Audit algorithm support on Business and Maintenance objects to add information to log entries attached to these objects. See the *Oracle Utilities Application Framework Software Development Kit* and online Administration documentation for a description of programmatic implementation of auditing.

Audit configuration for ORMB is performed at the table level. You can enable auditing on a table by navigating to the **Table** screen from the **Administration** menu, and then configuring the following fields for the table:

- **Audit Table** – You need to configure a database table to store the audit information. By default, the CI_AUDIT table can be used for this purpose. If a custom table is used to store the audit information, ensure that the structure of the custom table is similar to the CI_AUDIT table to ensure compatibility.
- **Audit Program** – You must configure a class or program that will record and process the audit information. By default, several pre-built audit programs are available for use:
  - com.splwg.base.domain.common.audit.DefaultTableAuditor – This is the default java-based audit class provided by the product. It audits any changes to any fields configured to track auditing information.
  - com.splwg.base.domain.common.audit.ModifiedTableAuditor – This is an alternative to the DefaultTableAuditor class. However, it does not audit inserts or deletions of empty string field data. For example, changes from null values to empty spaces or empty spaces to null values are not logged.

Note: It is possible to implement custom Audit handlers using the base classes as parent classes. See the *Oracle Utilities Application Framework Software Development Kit* documentation on how to extend the product.

- **Audit Conditions** – A set of switches are configurable on each field you wish to include in auditing to determine the conditions of auditing. At least one of these switches must be enabled for auditing to be registered:
  - **Audit Delete Switch** – Enable this switch to audit delete operations against this field.
  - **Audit Insert Switch** – Enable this switch to audit insert operations against this field.
  - **Audit Update Switch** - Enable this switch to audit update operations against this field.

To maintain the audit information, navigate to the **Table** menu option on the **Administration** menu and specify the table to enable auditing against.

Specify the Audit Table, Audit Program (and associated type) and configure the Audit Switches on the fields you wish to track.

Note: To enable auditing on a running version of the product, the online data cache must be flushed, or the product restarted. See the *Oracle Revenue Management and Billing Server Administration Guide* for more details.

7.2 Read Auditing

Whilst the inbuilt audit facility is mainly used to register changes in data, it can also be used to register whenever data is accessed for auditing purposes. Read auditing is different from the standard auditing as it focuses on zones. On the zone configuration, there is an ability to configure an Audit Service Script which is called whenever the zone is displayed to determine which criteria and result records are displayed. In the current release, the read auditing feature is available for the following zone types:

- F1-DE
- F1-DE-QUERY
- F1-DE-SINGLE
- F1-MAPDERV
- F1-MAPEXPL

The information audited can be determined by using programs and logged based on your requirements. See the *Oracle Revenue Management and Billing Online Help* for descriptions and samples for read auditing.

Note: Products ship with sample generic inquiry Audit code specific to the product. These can be reused or altered to suit your needs. See the product documentation for details of these samples.

7.3 Integrating to Audit Vault

Note: Customers using Oracle 12.x and above, should use Unified Auditing to ensure consistent capture of audit information. See the *Technical Best Practices For Oracle Utilities Application Framework Based Products (Document ID: 560367.1)* article on My Oracle Support.

Oracle Utilities Application Framework contains an internal audit facility that provides a basic audit facility for recording changes and optionally, inquiring, data by online users. Whilst this facility is enough for most needs it may be replaced with using Oracle Audit Vault to provide an enterprise-wide audit facility.

Oracle Utilities Application Framework supports the use of Audit vault in association or as a complete replacement for the inbuilt Auditing feature.

Audit Vault collects audit information at the database level, using the Database Auditing features of the Oracle Database, and loads them into a separate Audit Vault database. The information in that database can be queried, reported, and managed using the Audit Vault front end. For example:

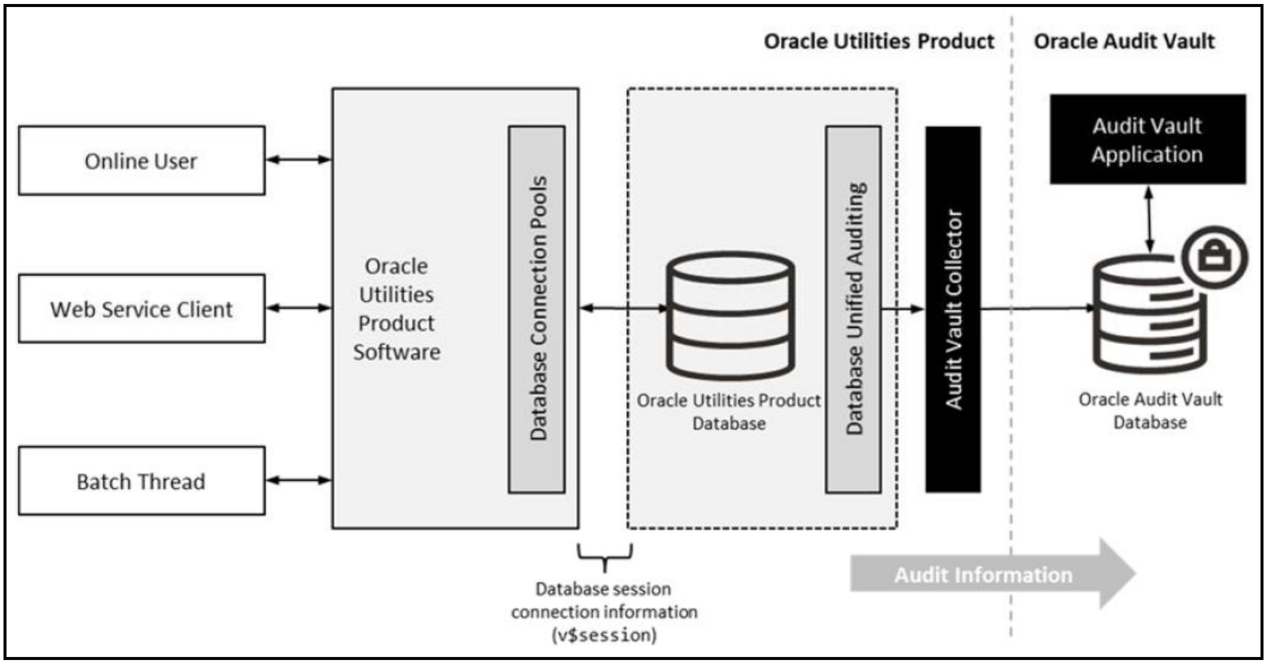

Oracle Utilities Product
Oracle Audit Vault
Online User
Web Service Client
Batch Thread
Oracle Utilities Product Software
Database Connection Pools
Oracle Utilities Product Database
Database Unified Auditing
Audit Vault Collector
Audit Vault Application
Oracle Audit Vault Database
Database session connection information (v$session)
Audit Information

Figure 3: Using Audit Vault

To use Audit Vault, the following must be configured:

Configuration	Description
Setup Database Auditing	The Database auditing feature must be enabled to store the relevant audit information. The level of auditing information and the location of the audit information is configurable. See the *Oracle Database Security Guide* for more information about database auditing and *Best Practices of Auditing* for more information about the various methods available.

Configuration	Description
Design Database Auditing	The tables, users, and SQL statements to audit need to be specified on the product database. This is typically done by the database administrator using the AUDIT statement.
Install and Configure Audit Collector	On the host holding the product database an Audit Vault Agent needs to be configured to pass audit information to Audit Vault and implement data retention policies for audit information.
Configure Audit Vault	Audit Vault can be configured to implement policies, alerts, and reports on the Audit data. Audit Vault can be configured to set an Audit Data Retention Policy for its internal audit information.

8. Database Security

Oracle Database supports a wide range of security configurations natively or via additional options available. For more information about the security options available for the database, see the *Oracle Database Security Guide*.

This section contains the following topics:

- Database Users
- Database Roles
- Database Permissions
- Using Transparent Data Encryption
- Using Database Vault

8.1 Database Users

Oracle Revenue Management and Billing ships a predefined set of users to be used by the product at configuration and runtime. These users are specified in the installation of the product to build the database and load its initial dataset.

The following users are available:

- SPLADM – This is the default DBA administration account which owns the product schema. This user is used to create and maintain the structures of the database. It is used by DBA personnel to maintain the product schema and indexes.
- SPLUSER – This is the default main product user used by the product to access the SPLADM schema. The product uses this physical user ID as a pooled user with pooled connections to the database. Variations on this account can be created for each channel of access using the following configuration settings.

Configuration Parameter	Comments
BATCH_DBUSER	Database User for Batch
DBUSER	Database User for online (Default: SPLUSER)
XAI_DBUSER	Database User for Web Services

- SPLREAD – This is the default read only user available for reporting tools or external direct interfaces to use on the product database. This user is not used by the product.

Note: For customers on older versions of ORMB, this user was also used for the ConfigLab component.

- CISOPR, OPRPLUS – These are optional operator users that can be used to delegate backup and restore operations on the product.

Note: The values of these users can be altered to customer specific values at installation time. See the *Oracle Revenue Management and Billing Installation Guide* and *Oracle Revenue Management and Billing Database Administrator's Guide* for more information.

8.2 Database Roles

Oracle Revenue Management and Billing ships a set of database roles to allow administrators to allocate new database users to the relevant components of the product. The following roles are shipped by default for the product:

- SPL_USER – This role is available for database users who require update, insert, delete, and select access to the product schema. This role is used for product users.
- SPL_READ – This role is available for database users who require read only access to the product schema.

To use the roles the DBA grants the role to the database user to connect them to the schema in the desired fashion.

8.3 Database Permissions

Database permissions for ORMB are allocated at the role level with the role setting permissions to the schema objects. By default, the roles have full access to all the objects in the product schema, as dictated by the role.

Unless otherwise stated, it is not recommended to alter the database users used by the product to specific additional permissions on the product schema as this may cause permission issues.

Customers wishing to restrict external parties, such as external tools or reporting engines, to specific objects may use all the desired security facilities available in the database to implement those restrictions.

8.4 Using Transparent Data Encryption

Transparent Data Encryption (TDE) allows data to be encrypted at the storage level to protect the data files at the lowest level. From a product perspective, the implementation of Transparent Data Encryption requires no product configuration changes on the application server.

Note: To implement Transparent Data Encryption, DBAs will have to execute appropriate alter statements on product tables to indicate the level of encryption.

For more information about implementing Transparent Data Encryption, see the *Oracle Database Advanced Security Guide*.

8.5 Using Database Vault

By default, the database administration account as SQL Data Manipulation Language (DML) access to the ORMB schema, as dictated by the default permissions of the Oracle Database. It is possible to restrict the permissions of the DBA to SQL Data Definition Language (DDL) statements only using Database Vault. See the *Database Vault Administrator's Guide* for details of this facility.

The product includes a prebuilt database vault solution, see the *Technical Best Practices For Oracle Utilities Application Framework Based Products (Document ID: 560367.1)* article on My Oracle Support.

9. Security Integration

Whilst Oracle Utilities Application Framework applications provide a set of security facilities natively or via Oracle WebLogic, it is possible to augment the security with additional security features or security products. This section contains the following topics:

- LDAP Integration
- Single Sign-On Integration
- Oracle Identity Management Suite Integration
- OAuth2 Support

9.1 LDAP Integration

By default, Oracle WebLogic includes an internal security repository that uses the Lightweight Directory Access Protocol (LDAP) to provide authentication facilities. It also provides authorization services, but these are not typically utilized by ORMB. It is possible to replace the internal security repository with another LDAP compliant security source.

To use an alternative source as a security repository the following process must be used:

- Oracle WebLogic must be configured to use the external LDAP security source for authentication. See the *Oracle WebLogic* documentation for more information. Oracle WebLogic customers can see the Configuring LDAP Authentication Providers section of the *Oracle Fusion Middleware Securing Oracle WebLogic Server Guide*.
- The product LDAP import feature can be used to initially populate the authorization model from the LDAP source as outlined in the *Technical Best Practices For Oracle Utilities Application Framework Based Products (Document ID: 560367.1)* article on My Oracle Support.

Note: Whilst LDAP sources are the most common security repository, it is possible to use alternative security authentication sources as supported by Oracle WebLogic. See the documentation provided with Oracle WebLogic for more details.

9.2 Single Sign-On Integration

One of the common security integrations is the ability to implement Single Sign-On with ORMB. This enables end users to access the product minimizing the need to re-authenticate each time.

Oracle WebLogic in association with other technologies can be configured to support Single Sign-On. For more information, see the *Technical Best Practices For Oracle Utilities Application Framework Based Products (Document ID: 560367.1)* article on My Oracle Support.

9.2.1 Kerberos Support

Single Sign-On (SSO) with Microsoft clients allows cross-platform authentication between Web applications running in Oracle WebLogic and .NET Web service clients or browser clients (for example, Microsoft Internet Explorer) in a Microsoft domain. The Microsoft clients must use Windows authentication based on the Simple and Protected Negotiate (SPNEGO) mechanism.

See Configuring Single Sign-On with Microsoft Clients for details of configuring Oracle WebLogic to use Kerberos.

9.3 Oracle Identity Management Suite Integration

Oracle offers a comprehensive set of security products as part of the Oracle Identity Management Suite that can be used to augment the security setup at your site. The product can be integrated with the following components of Oracle Identity Management Suite:

Component	**Description**
Oracle Identity Manager	Oracle Identity Manager can be used to centralize user provisioning to the product, password rule management and identity administration.
Oracle Access Manager	Oracle Access Manager can be used to provide authentication, single Sign-On, access controls and user tracking.
Oracle Adaptive Access Manager	Oracle Adaptive Access Manager can be used to provide fraud tracking and multi-faceted authentication.
Oracle Virtual Directory	Oracle Virtual Directory can be used to provide virtualized LDAP security access to LDAP and non-LDAP security sources.
Oracle Internet Directory	Oracle Internet Directory can be used as a LDAP security store.

For more information, see the *Technical Best Practices For Oracle Utilities Application Framework Based Products (Document ID: 560367.1)* article on My Oracle Support.

9.4 OAuth2 Support

Note: For customers using OAuth2 for Federated Security, see the dedicated section in this guide for detailed instructions.

OAuth2 is supported for delegated access to APIs within the ORMB application. From the configuration perspective, the following is required for Oauth2:

- The `CLIENT-CERT` setting should be specified for security in the product. This indicates that the security is coming from an external source to the domain.
- Configure Oracle WebLogic domain security realm to support OAuth2 using the instructions provided with the relevant version of Oracle WebLogic.
- If you are using a service provider like Oracle Identity Suite or Oracle Access Manager in association with Oracle WebLogic, then see the documentation provided with those products for domain specific setup instructions.
- If you are using a third-party provider in association with Oracle WebLogic, see the documentation provided with that provider with additional setup instructions.

10. Keystore and Truststore Support

Oracle Revenue Management and Billing supports the ability to store cryptographic keys and certificates. Keystores are used to encrypt and decrypt data such as passwords and for the data encryption component of Oracle Utilities Application Framework. Because of their importance, it is vital that keystore-related files be secured and only be accessible by administrators. It is also possible to implement a truststore to ensure the integrity of certificates.

This section contains the following topics:

- Creating the Keystore and Truststore
- Altering the Keystore/Truststore Options
- Synchronize Data Encryption
- Upgrading from Legacy to Keystore
- Importing Keystores/Truststores

10.1 Creating the Keystore and Truststore

Points to Note:

For backward compatibility, customers on older versions will have a default keystore and truststore created upon upgrade with backward compatible values.

If the keystore and truststore is not present, Oracle Utilities Application Framework will revert to the internal cryptography used in previous releases.

Passwords encrypted using this keystore will be prefixed with ENCKS and legacy password encryption uses prefix ENC.

The settings in this section are defaulted in installation and can be altered using overrides in templates/FW_spl.properties.keystore.truststore.include as outlined in the *Oracle Revenue Management and Billing Server Administration Guide*.

Typically, a keystore and truststore are created using the java keytool utility manually but the Oracle Utilities Application Framework utilities have been extended to allow customers to create and manage the keystore from the command line.

Before creating the keystore the following settings must be set in the installation, as per the *Oracle Revenue Management and Billing Server Administration Guide*:

- KS_ALIAS - The alias used to encrypt/decrypt passwords by Oracle Utilities Application Framework to access the keystore. By default, it is set to ouaf.system.
- KS_ALIAS_KEYALG - The algorithm used by the KS_ALIAS entry in keystore to encrypt the passwords. By default, it is set to AES (i.e. Advanced Encryption Standard).
- KS_ALIAS_KEYSIZE - The strength of the keystore for the KS_ALIAS entry. By default, it is set to 256.
- KS_HMAC_ALIAS - The HMAC alias used by the Encryption Feature Type of Oracle Utilities Application Framework. By default, it is set to ouaf.system.hmac.

- KS_HMAC_ALIAS_KEYALG - The algorithm used by the KS_HMAC_ALIAS entry in keystore to encrypt the data. By default, it is set to HmacSHA256.
- KS_HMAC_ALIAS_KEYSIZE - The strength of the keystore for the KS_HMAC_ALIAS entry. By default, it is set to 256.
- KS_KEYSTORE_FILE - Location of the keystore file.
- KS_MODE - Keystore Padding mode. By default, it is set to CBC (i.e. Cipher-block Chaining).
- KS_PADDING - Key padding algorithm used for keystore. By default, it is set to PKCS5Padding.
- KS_STOREPASS_FILE - Keystore Password file.
- KS_STORETYPE - Keystore type. By default, it is set to JCEKS.
- TS_ALIAS – Alias used for trust store. By default, it is set to ouaf.system.
- TS_ALIAS_KEYALG - The algorithm used by the TS_ALIAS entry in truststore. By default, it is set to AES.
- TS_ALIAS_KEYSIZE - The strength of the truststore for the TS_ALIAS entry. By default, it is set to 256.
- TS_HMAC_ALIAS - The HMAC alias used by the truststore. By default, it is set to ouaf.system.hmac.
- TS_HMAC_ALIAS_KEYALG - The algorithm used by the TS_HMAC_ALIAS entry in truststore to encrypt the data. By default, it is set to HmacSHA256.
- TS_HMAC_ALIAS_KEYSIZE - The strength of the truststore for the TS_HMAC_ALIAS entry. By default, it is set to 256.
- TS_KEYSTORE_FILE - Location of the truststore file.
- TS_MODE - Truststore Padding mode. By default, it is set to CBC.
- TS_PADDING - Key padding algorithm used for truststore. By default, it is set to PKCS5Padding.
- TS_STOREPASS_FILE - Truststore Password file.
- TS_STORETYPE - Truststore type. By default, it is set to JCEKS.

Once these settings are specified, the keystore or truststore is created using the following command:

```
initialSetup.sh -k|-K
```

This generates the keystore (-k) or truststore (-K) using the credentials outlined in the Keystore or Truststore Password file.

10.2 Altering the Keystore/Truststore Options

Note: This process should be used for any keystore/truststore change including copying keystores/truststores across environments.

After creating the keystore, if any of the keystore values need to be changed, then the system needs to be realigned to the new configuration. The following process must be performed:

- Logon to the machine where you wish to make the changes to the settings.
- Execute the splenviron.sh -e <environment> command where <environment> is the environment on the machine to change.
- Shutdown the environment.
- Alter the keystore parameters to suit the new desired configuration using the configureEnv.sh -a utility.
- Execute the initialSetup.sh -k utility to recreate the keystore (-k) or initialSetup.sh –K to recreate the truststore (-K) with the new settings.
- Execute the configureEnv.sh once more and press enter on each password prompt to re-encrypt the passwords with the new settings.
- Execute the initialSetup.sh command to apply the changes to the configuration files.

Note: For customers using native installation, update the Deployments using the Oracle WebLogic console or Oracle Enterprise Manager to load the new versions of the product EAR files.

- If the encryption values have changed, the data encrypted in the database must be re- encrypted to match the new settings using the process outlined in the Synchronize Data Encryption section.

10.3 Synchronize Data Encryption

Points to Note:

Failure to synchronize data when encryption values change will cause outages and unexpected behavior in ORMB.

You should shut down the ORMB application while running this process.

If at any time the encryption values change, the values that are encrypted using the old value must be updated to reflect the new settings. A new utility named com.splwg.shared.common.ChangeCryptographyKey is provided to synchronize data changes. The following keys are updated using this utility:

- Database Passwords used in Feature configurations such as Database Update features.
- Message Sender and Receiver Passwords (depending on Sender and Receiver type)
- Reporting tool integration passwords
- Multi-Purpose Listener passwords (for selected products)
- Email Adapter configuration.
- Web Services Passwords (legacy only)
- Security Hashes on user records

The following process is to be used:

- Logon to the machine you have made the changes upon as the product administrator.
- If you have not already done so, use the splenviron utility to set the environment variables for the product environment.
- Execute the following command:

```
perl <SPLEBASE>/run_java_standalone.plx
com.splwg.shared.common.ChangeCryptographyKey  [-t|-l|-h|-p] [old-settings]
```

The following table lists and describes the command line options:

Command Line Option	Description
-t	Test Mode (no commit of changes)
-l	Convert Legacy/OUAF System key
-h	Convert User hashes only
-p	Convert encrypted passwords only
[old settings]	You can use the following old settings: - Dcom.oracle.ouaf.system.old.keystore.file= <oldfile> - Dcom.oracle.ouaf.system.old.keystore.passwordFileName =<oldpassfile> - Dcom.oracle.ouaf.system.old.keystore.type= <oldtype> - Dcom.oracle.ouaf.system.old.keystore.alias= <oldalias> - Dcom.oracle.ouaf.system.old.keystore.padding= <oldpadding> - Dcom.oracle.ouaf.system.old.keystore.mode= <oldmode> Where, • <oldfile>: Original Keystore file • <oldpassfile>: Original Password Store file • <oldtype>: Original Keystore type • <oldalias>: Original alias • <oldpadding>: Original Padding • <oldmode>: Original Mode **Note:** The command line options such as –t, -l, -h, and –p should not be used with these old settings.

Points to Note:

Only specify the values that have been changed.

This command must be run once for each alias.

After executing the com.splwg.shared.common.ChangeCryptographyKey utility, you must execute $SPLEBASE/bin/invokeDBUpdatePatch.sh to reset the database patching credentials as follows:

- If you have not already done so, use the splenviron.sh utility to set the environment variables for the product environment.
- Run the command with the –b option to go into interactive mode and reply to the prompts. Use the –h option to get help.

```
$SPLEBASE/bin/invokeDBUpdatePatch.sh -b
```

10.4 Upgrading from Legacy to Keystore

When upgrading from past releases of Oracle Utilities Application Framework and adopting the new keystore it is recommended to use the following process to adopt the keystore:

- Ensure all passwords have been updated by executing the configureEnv and pressing enter at each password prompt.
- Execute the process outlined in the Synchronize Data Encryption section by running the com.splwg.shared.common.ChangeCryptographyKey utility with the -l option to convert old keys to new keys. For example:

```
java ChangeCryptographyKey -l
```

- Ensure that you also execute the invokeDBUpdatePatch.sh utility mentioned in the Synchronize Data Encryption section.
- Optionally, it is possible to update the passwords using the LegacyCryptographerUpdater utility on individual passwords using the following command:

```
java LegacyCryptographyUpgrader [-f <file>| -p <password>]
```

Where options are:

Command Line Option	Description
-f <file>	Read <file> for password and re-encrypt to stdout
-p <password>	Decrypt old password <password> and re-encrypt to stdout. The password should be already in the ENC format.

For more information and examples, see the *Oracle Revenue Management and Billing Installation Guide and the Oracle Utilities Application Framework - Keystore Configuration (Document ID: 2014161.1) article on* My Oracle Support.

10.5 Importing Keystores/Truststores

While ORMB supplies a default keystore and truststore it is possible to import existing keystores and truststores from alternative sources (such as a corporate level set of stores or from a trusted CA authority).

To import a keystore or truststore the following process should be followed:

- Logon to the machine you have made the changes upon as the product administrator.
- If you have not already done so, use the splenviron utility to set the environment variables for the product environment.
- Ensure the KS_IMPORT_KEYSTORE_FOLDER or TS_IMPORT_KEYSTORE_FOLDER is set in the ENVIRON.INI prior to continuing. These are the locations the files to be imported will be located for keystores and truststores respectively.
- Copy the new keystore or truststore to the locations specified in the KS_IMPORT_KEYSTORE_FOLDER or TS_IMPORT_KEYSTORE_FOLDER respectively.
- Execute the initialSetup.sh –s | -S to import the keystore (-s) or truststore (-S) successfully.

11. Encryption Feature Type

One of the major features of Oracle Revenue Management and Billing is the ability to mask and encrypt data to protect sensitive information. This encryption is implemented in a Feature Configuration using the Encrypted Feature Type.

Oracle Utilities Application Framework supports Feature Configuration which store specific configuration settings for features in the product to be implemented. Feature Configurations allow simple configurations to be implemented for specific features.

Feature Configurations can be maintained using the **Feature Configuration** menu option on the **Administration** menu.

For the Encryption feature, one Feature Configuration should exist for the Encryption Feature Type with an option per field to encrypt.

Note: If the product does not ship a Feature Configuration for Encryption, then it can be created as a Customer Modification. Prefix the name of the Feature Name with CM.

11.1 Configuration of Encrypted Fields

For more information about encrypting data, see the *Oracle Utilities Application Framework Administrative Guide*.

12. Web Services Security

In ORMB, the Inbound Web Services capability, based upon a JAX-WS/JAX-RS implementation, provides the WS-Security and WS-Policy support on individual Inbound Web Services from several perspectives.

Points to Note:

This section is applicable for the REST and SOAP based services which are defined as inbound web services in the ORMB application.

See the *Web Services Best Practices for Oracle Utilities Application Framework (Document ID: 2214375.1) article on* My Oracle Support for additional implementation advice on web services security.

This section contains the following topics:

- Annotation Security
- Oracle WebLogic WS-Policy Support
- Oracle Web Services Manager Support
- Support for Multiple Policies
- Importing Certificates for Inbound Web Services

12.1 Annotation Security

It is possible to implement custom WS-Policy support using inbuilt annotation support on individual Inbound Web Service definitions. This allows the policies to be implemented within the ORMB application and allows for backward compatibility for sites using XML Application Integration (XAI).

ORMB supports custom WS-Policy files using the following method:

- The WS-Policy formatted XML file containing the policy definition is installed in the `$SPLEBASE/splapp/iws/resources/policies` or `%SPLEBASE%\splapp\iws\resources\policies` directory.
- A Web Service Annotation of type F1POLICY is defined using the Web Service Annotation maintenance function. In the annotation entry, the following should be configured:
  - uri: The name of the policy XML file located in the policies directory in the format "policy:<policyname>" where <policyname> is the name of the file containing the policy. For example: "policy:UsernameToken.xml".
  - attachToWsdl: Whether the WS-Policy file is attached to the WSDL (for SOAP Web Services only).
  - direction: Indicates when to apply the policy as per weblogic.jws.Policy.Direction. For example: Direction.both
- The annotation is attached to the relevant Inbound Web Services to implement the policy using the Inbound Web Services maintenance function.

Note: Multiple policies can be added as documented in the Support for Multiple Policies section.

12.2 Oracle WebLogic WS-Policy Support

Note: Customers wishing to use Oracle WebLogic policies should not configure policies via the annotations on the same web service. Use of Oracle WebLogic policies and annotations are mutually exclusive and can cause unintentional security violations.

Oracle WebLogic has inbuilt WS-Policy Message-Level and Transport-Level support with predefined policies that can be used with Inbound Web Services at the container level. These policies can be attached within the container on individual services using the Oracle WebLogic console. For more information, see Attach a WS-Policy file to a Web Service.

Note: Message level policy support is restricted to the whole message not to the parts of the message.

Multiple policies can be supported directly by specifying additional policies in the Chosen Inbound Message Policies.

12.3 Oracle Web Services Manager Support

Points to Note:

Customers using annotations can also use Oracle Web Services Manager policies by specifying the F1-OWSM policy type as an annotation.

Customers wanting to use Oracle Web Services Manager within the product domain should use the Full JRF Profile not the Restricted JRF Profile as recommended as a minimum for the product installation.

Oracle Web Services Manager can be used to secure individual Inbound Web Services providing additional WS-Policies and access control support. With the Oracle WebLogic WS-Policy support, configuration is performed using the Oracle WebLogic console by specifying individual policies on individual Inbound Web Services by using the policy type of Oracle Web Services Manager (OWSM).

See Using Oracle Web Service Manager Security Policies for additional information about Oracle Web Services Manager.

12.3.1 Access Control Support

Note: By default, all Inbound Web Services are accessible to all valid users.

Oracle Web Services Manager allows to configure access control for Inbound Web Services to provide additional security access controls. This facility allows for multiple rules to be configured implementing access rules across the following areas:

- **Basic Policies**: Policies relating to identity, group, role, environment mode, and generic global rules.
- **Date and Time Policies**: Policies relating to specific dates and times including periods of access.
- **Context Element Policies**: Policies relating to data within the service itself.

Policies can be individually specified per service operation or combined to implement complex access requirements. For more information, see Security Policy Conditions about specific policies and how to enable policies on services.

12.4 Support for Multiple Policies

Within each policy regime (annotation, Oracle WebLogic, or Oracle Web Services Manager), it is possible to configure multiple policies where web service clients must conform to at least one policy specified. The following methods are supported:

- **Annotations**: In the Inbound Web Service, each policy can be added as an individual annotation with the sequence number designating the order they are checked. It is also possible to delegate Oracle Web Services Manager for policies as part of an annotation using the `F1-OWSM` annotation type.
- **Oracle WebLogic/Oracle Web Services Manager**: In the web service deployment, it is possible to designate multiple policies in the Chosen Policies column of the individual web service WS-Policy configuration. The order ofthe policies is dictated by the position in the list of Chosen Policies.

12.5 Importing Certificates for Inbound Web Services

If your implementation uses certificates for security, the certificate must be deployed with the Inbound Web Services deployment. To perform this activity:

- Ensure the certificate is valid and is installed on the Oracle WebLogicservers used for deployment of the Inbound Web Services.
- Login to the machine where you have made the changes as the product administrator.
- If you have not already done so, use the `splenviron` utility to set the environment variables for the product environment.
- Execute the initialSetup[.sh] -i [<host>:<port>] to import the certificate into the Inbound Web Service deployment.

13. Allow List Support

In the Oracle Cloud implementations of ORMB, the use of allow list is enforced to protect resources within the implementation. These allow lists can also apply to non-cloud implementations and in some cases can be extended to suit individual needs.

Note: Custom allow lists are not supported on Oracle Revenue Management and Billing Cloud Services.

This section contains the following topics:

- URL Allow List
- SQL Allow List
- HTML Allow List
- Groovy Allow List

13.1 URL Allow List

Points to Note:

The URL allow list is blank by default unless otherwise configured as part of your customizations.

For this facility to be used the com.oracle.ouaf.uriValidation.enable parameter in the spl.properties file must be set to true.

It is possible to limit the values of URLs within the product for key objects with the configuration. This is implemented as a allow list that can filter on scheme (aka protocol), hosts and ports. These are checked at runtime and can generate an error if they do not adhere to the allow list.

The feature allows for the following:

- Individual scheme hosts and port combinations can be configured to limit runtime access for specific features.
- Specification of the '*' wildcard is supported for scheme, hosts and ports.

The allow list is configured using the following settings in the spl.properties file:

Configuration Parameter	Comments
com.oracle.ouaf.uriValidation.enable	Enable or disable URI validation.
com.oracle.ouaf.whitelist.file	Location and name of product allow list file.
com.oracle.ouaf.customer.whitelist.file	Location and name of custom allow list file.

The format of the allow list file is as follows:

XML Tag	Comments
"<Parameter>"	Feature within product to limit. This is a preset string linked to a URI parameter in the product.
uri	URI tag

XML Tag	Comments
scheme	Protocol supported by <Parameter>. Valid values will vary depending on the <Parameter> value. For example, URL's support file, http, https, etc.
host	Host name(s) or IP Address (es) to filter upon.
port	Port number(s) to filter upon.

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<whitelist>
…
  "Message Sender HTTP URL properties"
  <uri>
      <scheme>https<scheme/>
<host>myhost.mycompany.com<host/>
      <port>*<port/>
  </uri>
…
</whitelist>
```

See the *Oracle Revenue Management and Billing Server Administration Guide* for more details of the usage and format of this file.

13.1.1 Implementing a Custom URL Allow Lists

It is possible to establish a custom allow list to implement URL allow lists for non-cloud implementations. The format of the allow list is the same as the above allow list and this will augment and enhance the existing allow list (if present). To support a custom allow list perform the following steps:

- Logon to the machine you have made the changes upon as the product administrator.
- If you have not already done so, use the splenviron utility to set the environment variables for the product environment.
- If it does not already exist, clone the product allow list located as indicated in the com.oracle.ouaf.whitelist.file parameter. Ensure the custom file is in the same location as this allow list file and is prefixed with "cm" to indicate it is a customization. The file name can be valid for your operating system and site preferences and must be suffixed with ".xml". Uncomment any section you want to set.
- Create a custom spl.properties template or use the templating function to set the following parameter with the filename and location of the custom allow list.

  ```
  com.oracle.ouaf.customer.whitelist.file=<filename>
  ```
- Execute the initialSetup.sh to implement the new setting.

13.2 SQL Allow List

Points to Note:

Custom SQL Allow List are not supported at this time.

Sites not wanting to use this SQL function allow list should ensure that the spl.runtime.customSQLSecurity is set to false in the spl.properties file.

The SQL used in query zones and in Groovy scripts can be limited in terms of what SQL functions are supported to prevent performance issues or inappropriate access to the database via functions.

The product provided allow list is implemented as a Managed Content object named F1-SQLFunctionWhiteList. This allow list is not changeable and lists the supported functions that are allow listed for use. Any function used that is not used in this allow list, if the facility is enabled, will generate a runtime error when the SQL is executed.

13.3 HTML Allow List

The HTML used in UI Maps can be limited using a HTML allow list. This product managed allow list lists the valid HTML tags that can be used in the HTML objects. The allow list is implemented as a Managed Content allow list named F1-HTMLWhiteList.

Any attempt to run a UI Map with a tag that is not listed in the F1-HTMLWhiteList will be ignored (as comments) which may lead to unexpected behavior.

13.3.1 Implementing a Custom HTML Allow List

It is possible to replace the inbuilt HTML allow list by creating a custom HTML allow list to extend the tags supported at runtime in UI Maps.

To implement a custom HTML allow list, copy the F1-HTMLWhiteList Managed Content to CM-HTMLWhiteList Managed Content and extend the allow list.

Points to Note:

Do not remove any tags that exist within F1-HTMLWhiteList in your custom allow list. This may cause unexpected behaviour across base UI Maps.

Any upgrades to F1-HTMLWhiteList must be also manually reflected in CM-HTMLWhiteList for any subsequent upgrade.

13.4 Groovy Allow List

Note: The current implementation of the Groovy Allow List is dynamically generated and cannot be altered.

The Groovy language has been added as an alternative language used for scripting. As the Groovy language can access low level APIs it has been allow listed to exclude parts of the language not appropriate for cloud implementations. The allow list confirms to the Oracle Cloud SDK Supported Groovy Classes and Methods. ADF extensions to Groovy are not supported. See the online documentation for additional advice and examples.

14. Custom Authentication Service Provider

In Oracle Utilities Application Framework, a custom Oracle WebLogic Authentication Service Provider has been provided to support complex domain security configurations.

This section contains the following topics:

- What does this Security Provider do?
- Where would I use this Security Provider?
- Implementing the Security Provider

14.1 What does this Security Provider do?

The Oracle Utilities Application Framework Service Provider allows Oracle WebLogic domains to be extended with the following additional checks:

- The provider will check that the authentication user has been defined as a User object within the product. If the user does not exist, then the provider will issue an error to be processed according to the rules in the domain security setup (including using the Oracle WebLogic Adjudicator Providers to implement access rules).
- If the user exists as a User object within the product, the user record is checked that the user is enabled for use. It is possible to disable a user at the authorization level as well as the authentication level. This provider performs the authorization level check. If the user is disabled, the provider will issue an error to be processed according to the rules in the domain security setup (including using the Oracle WebLogic Adjudicator Providers to implement access rules).

14.2 Where would I use this Security Provider?

This provider is designed to be used in several security scenarios:

- If the domain security realm uses several authentication sources then the Security Provider can be used to decide the order where the checks provided by this provider are executed. By default, the login even performs the same checks after all providers are called, the security provider allows that event to be done earlier in the chain.
- If the implementation is using single Sign-On (SSO) or federated security, this security provider can be used to decide when the checks performed are done in relation to these configurations.

By default, the checks performed by this provider are done automatically by the product login process. Use of the provider allows implementations to perform these checks earlier in the security checking process.

14.3 Implementing the Security Provider

Points to Note:

Before using the provider ensure a data source has been created to connect to the product database to access the SC_USER table.

Each Plugin Properties must exist on a separate line.

The Oracle Utilities Application Framework security provider is provided in the $SPLEBASE/tools/bin/auth subdirectory as ouaf-dbmsauth-<version>.jar. This jar file must be copied to the $DOMAIN_HOME/lib directory. After restarting the Administration server, the following must be configured to use this security provider:

- Login to the Oracle WebLogic Administration console using the appropriate administrator account.
- Navigate to the **Security Realms** → **myrealm** → **Providers** tab from the console.
- Select New to add a new Provider.
- Assign an appropriate name for the provider according to your site standards.
- Use the CustomDBMSAuthenticator for the Provider type.
- Use the Ok button to save the authenticator definition.
- Select the Name you assigned the provider to complete the configuration.
- Select the appropriate Control Flag for your site standards to determine the how the provider fits into the login sequence.
- Select the Provider Specific tab to configure the provider using the following settings:
    - Specify the data source created to connect to the database created earlier in the Data Source Name attribute.
    - Specify com.oracle.ouaf.fed.OuafDBMSAuthenticator for the Plugin Class Name.
    - Specify the userGroup=<usergroupname> where <usergroupname> is the realm group created for the product in the Plugin Properties. By default, this is set to cisusers if parameter not present. For example:

      ```
      userGroup=cisusers
      ```
    - Optionally, specify the users you wish to bypass from this Security provider by specifying the excludeUser=<listofusers> where <listofusers> is a list of authentication users delimited by "," to be excluded. For example:

      ```
      excludeUsers=system,weblogic,OracleSystemUser
      ```
- Save the Provider configuration.
- Optionally, use Reorder to set the order of check.
- Optionally, configure the Adjudicator Provider for additional rules.

15. Federated Security Support

In some security architectures, the identity used by an individual user can be shared across identity systems. Typically, this is used for cloud implementations where the product or identity is housed on a cloud system and needs to be shared across on-premises and cloud systems. This is the basis for the support of Federated Security within the product.

The Federated Security Support is supported for the following:

- Federated Single Sign-On Support for the online channel.
- Support for OAuth2 tokens for Inbound Web Services.
- Support for OAuth2 tokens for inbound RESTful Web Services.
- Support for OAuth2 tokens for outbound calls to external SOAP based web services.
- Support for OAuth2 tokens for outbound calls to external RESTful based web services.

Points to Note:

This capability supports Security Assertion Markup Language (SAML) 2.0 but is limited to authentication only. Authorization is supported in the product using the Security Model.

WLST commands in this section are for illustrative purposes only and assume that the user has connected to the relevant domain using the relevant WLST commands prior to execution of the command with the relevant credentials.

This section contains the following topics:

- Suggested References
- Federated Architecture
- Prerequisites for Federated Security
- Federated Online Authentication
- Federated Web Services
- Federated Outbound Messages

15.1 Suggested References

Note: It is assumed that sites wanting to use this capability are familiar with Federated Security and the products discussed in this section. Oracle strongly recommends reading relevant documentation related to this topic prior to using this advice.

The following references are recommended to be read before proceeding using this capability:

- Oracle Identity Federation Overview
- Fusion Middleware Administrating Oracle Access Management
- Oracle Unified Directory (optional)
- Fusion Middleware Administering Security for Oracle WebLogic Server
- Using Identity Federation in Oracle Access Management

- Fusion Middleware Understanding Oracle Web Services Manager

15.2 Federated Architecture

The Federated Architecture is based upon the following components:

- An *Identity Provider* (IdP) which authenticates the SAML 2.0 based identity. This is typically an on-premises (or third-party) provider that provides the ability to validate and share identity across applications/requesting systems.
- A *single sign-on* product, acting as a Service Provider (SP), to detect logins and appropriately process SAML 2.0 requests and responses.
- For Web Services a SAML 2.0 security-based provider or WS-Policy compliant policy.

15.3 Prerequisites for Federated Security

Note: This support has been verified with specific Oracle technology (and selected third-party providers). The instructions in this section are specific to that group of products. Alternatives may be used but instructions will need to be adjusted for those products.

The following products were used in the implementation of Federated Security:

Product	Minimum Version	Comments
Oracle Identity Management	11.1.2.3.0+	This version includes Oracle Identity Federation. If earlier version is used, then Oracle Identity Federation may need to be installed separately.
OHS/WebGate	12.2.1.4+	Includes Apache HTTP Server 2.4 which is the recommended version. The instructions in this document are based upon OHS/Webgate 11.1.2.3.0.
Oracle Utilities Application Framework	4.3.0.5.0+	-
Oracle Web Services Manager	12.2.1+	Installed with Oracle Fusion Middleware Infrastructure.
3rd Party Identity Provider (IdP)	-	This document uses Shibboleth as an example only. Any 3rd party provider supporting SAML can be used. See Installation Reference for Oracle WebLogic and Oracle Interoperability. **Note:** This is not a recommendation from Oracle for implementing this capability. Shibboleth is recognized as a reference infrastructure.

15.3.1 Process Flow

When a user logs into the ORMB application, there are two scenarios in relation to single sign-on:

- **Standard Single Sign-On** – This is where the single sign-on infrastructure and related security repository are either all in the cloud or all on-premises. In this case, the Oracle Access Management product is used as the Single Sign-On solution without the need for any federation.
- **Federated Single Sign-On** – This is where the sign-on and security provider are in a hybrid approach where one is on-premises (usually the identity provider) and one is on a cloud instance (usually the single sign-on solution).

The latter has the following flow when logging in:

1. The user logs into the product using the provided login to the proxy (OHS/Webgate) which houses the access rules and detects that the user has not been authenticated yet.
2. Oracle Access Manager is called to determine that validation of the authentication is the responsibility of the external Identity Provider (IdP). It formats a SAML 2.0 request to be sent to the Identity Provider.
3. It responds to the user via the product as a redirect.
4. The browser redirects the request to the Identity Provider configured by Access Manager.
5. The SAML 2.0 request is sent to the Identity Provider for processing.

   This part of the flow is shown below:

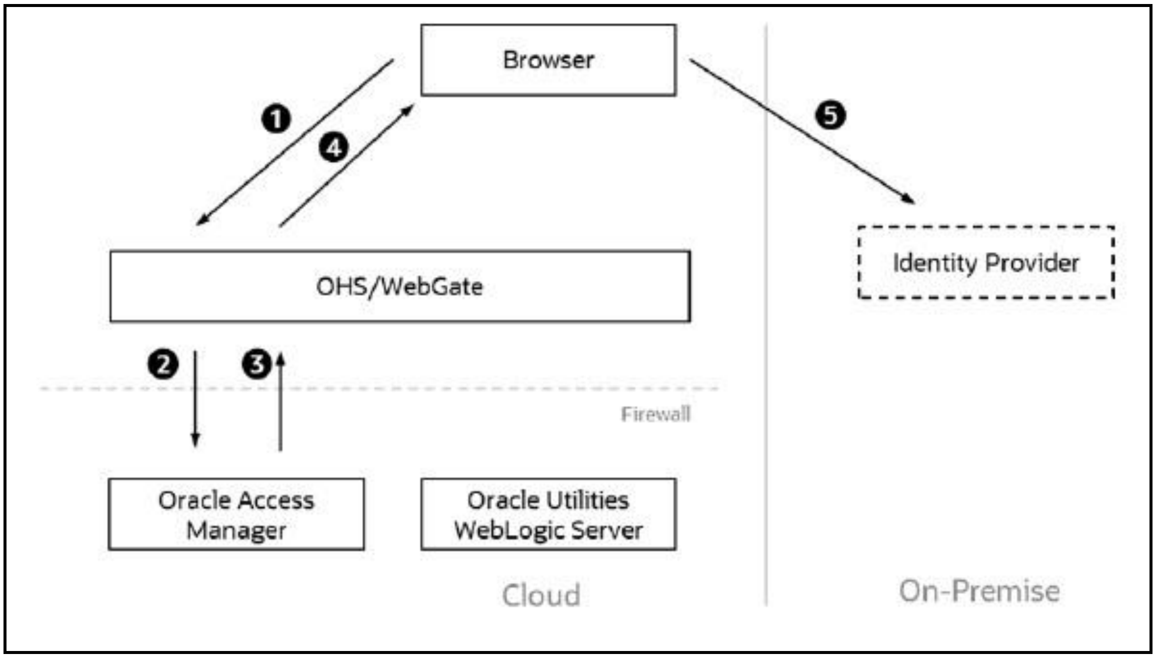


Figure 4: Federated Single Sign-On Process Flow – Part I

6. The Identity Provider authenticates the user and responds with a SAML 2.0 assertion which includes the authentication data.
7. The browser sends the SAML 2.0 assertion to Oracle Access Manager.
8. Oracle Access Manager validates the assertion and responds with an appropriate identity assertion for the SSO session.

9. The browser sends the SSO session to Webgate which now detects the user is valid and sends all the subsequent requests in that session to the product server(s).
10. The product server(s) respond with the appropriate responses for the transactions for the duration of the session.

    This part of the flow is shown below:

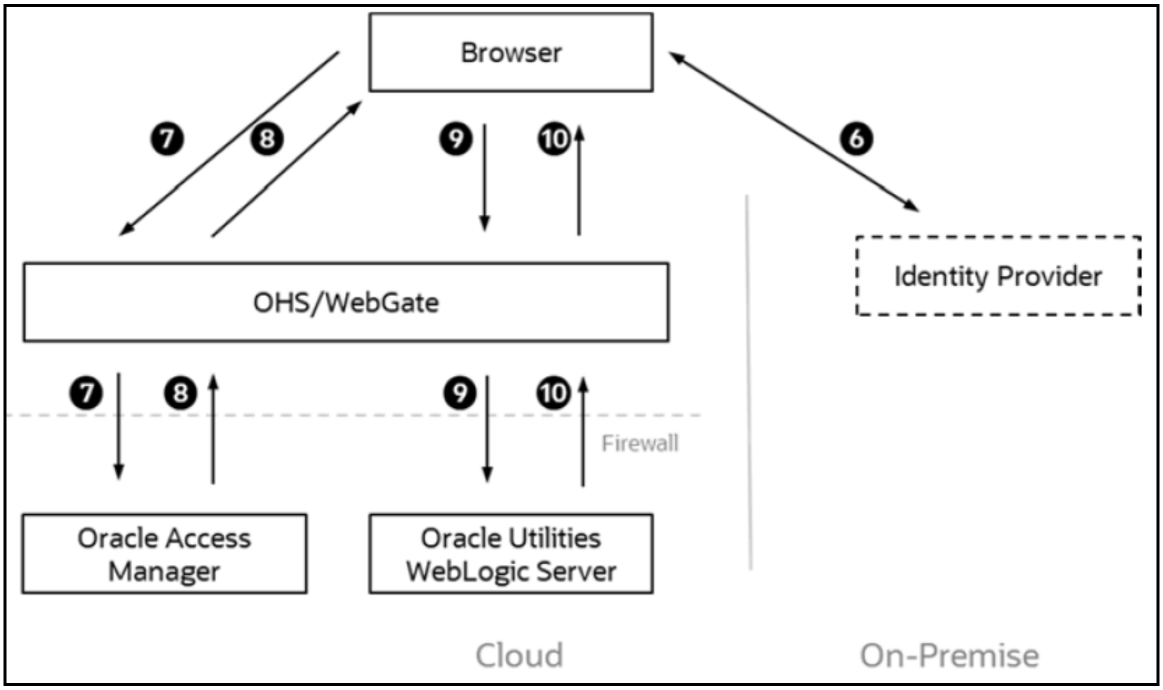


Figure 5: Federated Single Sign-On Process Flow – Part II

In summary:

- WebGate is the proxy and gatekeeper of the session.
- If it detects the user is valid, it will use the session information as authentication tokens for the product.
- If Webgate detects the user has not been authenticated, it will use the rules in Access Manager to determine how to authenticate the user.
- Access Manager should be configured to point to the Identity Provider and format/process any SAML 2.0 requests/responses transmitted to/from the Identity Provider. It also provides the SSO credentials for valid users.
- The Identity Provider is responsible for taking the SAML 2.0 request, validating the user in the request and sending an appropriate SAML 2.0 response.
- The browser acts as the interface, known as the User Agent, between the various components as it is the common point in the hybrid architecture. The feature uses standard features of HTTP to redirect traffic as needed.

15.4 Federated Online Authentication

The most common scenario is to federate the authentication of online users to a hybrid cloud solution.

15.4.1 Overview

The process for configuring an architecture for federation is as follows:

- **Install/Provision All the Software on the Various Platforms Ready for Configuration** - The installation and configuration documentation will outline the steps involved in getting the software installed and configured at a basic working level.
- Identity Provider Configuration - Setup your Identity Provider (IdP) to point to your security repositories and accept the necessary SAML 2.0 request to format the appropriate SAML 2.0 response.
- Oracle HTTP Server/WebGate Configuration - Configure your proxy servers to accept requests and, optionally, support reverse proxy (to detect the client connection).
- Define Identity Provider Partner in Oracle Access Manager – Configure the access management solution with the details of the Identity Provider.
- Enable Just In Time Provisioning in Identity Federation – Whilst the user would exist in the security repository and the product, it needs to be in the access management repository on first use.
- Define WebGate Agent – In a hybrid solution agents need to be configured to enable communication across partners.
- Copy WebGate Agent Configuration to OHS/WebGate – As agents need to communicate effectively ensure that the agent configurations match.
- Define Authentication Policy for the Product Domain – The domains must be configured with the linking policies to link all the components together.
- Export the OAM SAML Metadata (optional) – Optionally, each provider may differ SAML formats. This step exports the suggested format from the access management software into the Identity Provider to ensure SAML responses are formatted correctly.

The steps outlined above are described in the next subsections.

Note: All configuration files shown in this section are examples for illustrative purposes only. See the documentation provided with products for details of related settings.

15.4.2 Identity Provider Configuration

The Identity Provider (IdP) in the solution needs to be configured to allow SAML requests and appropriate responses to be received and sent to the service providers (SP) accessing the solution. The actual steps involved will vary from provider to provider but in general the following must be performed:

- The Identity Provider (or its networking component) must be configured to allow connections from the browsers access to the provider. This can be hostnames, IP addresses or IP Address ranges allocated to your users. These addresses can be via Virtual Private networks or direct according to your networking policies. For example, in a Shibboleth installation, the allowedRanges in the access-control.xml configuration file needs to be set:

```
<entry key="AccessByIPAddress">
    <bean parent="shibboleth.IPRangeAccessControl"
    p:allowedRanges="#{ {'127.0.0.1/32', '::1/128', '<validIpRangePattern>} }" />
</entry>
```

- Oracle Identity Federation uses the uid LDAP attribute for user identification. Ensure your Identity Provider uses this identifier in any communication to the solution. For example, in a Shibboleth installation, the Requester should be Oracle Access Management (with Oracle Identity Federation installed) with uid as the AttributeRule in the attribute-filter.xml configuration file:

```
<AttributeFilterPolicy id="<host>">
<PolicyRequirementRule xsi:type="Requester"
value="http://<OAMhost>:<port>/oam/fed" />
    <AttributeRule attributeID="uid">
        <PermitValueRule xsi:type="ANY" />
    </AttributeRule>
</AttributeFilterPolicy>
```

- Some Identity Providers require that the IdP be defined explicitly. For example, in a Shibboleth installation, the idp.entityID property must be set in the idp.properties configuration file:

```
idp.entityID= https://ouaf.oracle.com/idp/shibboleth
```

Note: See Entity Naming for a description of this setting.

- Configure your Identity Provider to connect to your LDAP security repository. See the documentation with your Identity Provider for detailed information on this process. For example, in a Shibboleth installation, the ldap.properties file controls the interface to the LDAP.

Note: If your implementation uses trust stores, ensure that the trust is also configured correctly for handshaking between the Identity Provider and LDAP security repository.

- Some Identity Providers need to understand the SAML request and response from/to the Oracle Access Management solution. The provider needs to understand how to download the SAML metadata to understand the interface. For example, in a Shibboleth installation, the **metadata-providers.xml** configuration file needs to be configured:

```
<MetadataProvider id="HTTPMetadata"
    xsi:type="FileBackedHTTPMetadataProvider"
    backingFile="%{idp.home}/metadata/localCopyFrom_https_<OAMhost>_<port>.xml"
    metadataURL="https://<OAMhost>:<port>/oamfed/sp/metadata">
</MetadataProvider>
```

- Some Identity Providers specify their meta data for use with the Service Provider. Ensure the Identity Provider also has correct certificate usage (including X.509) for secure transmission of data. It is important that signing and encryption certificate match the certificates used in the architecture to avoid issues (such as "Signature verification failed for provider ID..." errors). See the *Identity Federation SSO Fails With "Signature verification failed for provider ID ..."* (Document ID: 2032605.1) article on My Oracle Support for more details. For example, in a Shibboleth installation, the X509 signing and encryption certificates in the **idp-metadata.xml** configuration file should match the certificates in **idp-signing.crt** and **idp-encrpytion.crt** files. In a default installation of Shibboleth the certificates are in the **credentials** subdirectory. Correct if necessary.
- This should complete the configuration of the Identity Provider.

15.4.3 Oracle HTTP Server/WebGate Configuration

Install and configure the Oracle HTTP Server (aka Web Tier) and the Oracle Access Manager Web Gate as per the *Oracle Revenue Management and Billing Installation Guide*.

It is recommended that reverse proxy be configured for Oracle HTTP Server to be used with the ORMB application to enable content served by different servers to appear as if coming from a single server.

To enable the reverse proxy functionality, alter the Oracle HTTP Server **httpd.conf** configuration file with the following example section:

```
<VirtualHost *:<portnumber>>
   ProxyPreserveHost On
   ProxyPass "<context>" "https://<producthost>:<productport>/"
   ProxyPassReverse "<context>" "https:// <producthost>:<productport>/"
</VirtualHost>
```

Where:

Command Line Option	Description
<port>	Port Number allocated to proxy (default: 7777)
<producthost>	Host Name or cluster address for product
<productport>	Port or cluster port for product
<context>	The Product context as configured in the **WEB_CONTEXT_ROOT** setting of the **ENVIRON.INI** configuration file (For example: **/spl**)

For more information and alternatives see the *Reverse Proxy Guide*.

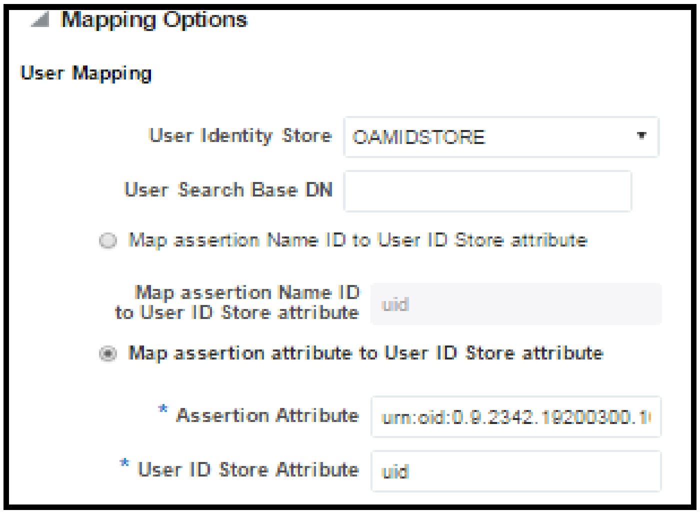
15.4.4 Define Identity Provider Partner in Oracle Access Manager

The Oracle Access Manager with Oracle Identity Federation needs to define the Identity Provider for the redirect and the integration between the products.

Oracle Identity Federation needs to be configured to identify the IdP and its interface. See the Managing Identity Federation Partners documentation for details of the configuration settings.

Using the **Federation → Launch Pad → Service Provider Management → Identity Provider Partner** option, ensure that the following is configured:

- Enable the partner using the Enable Partner function in the Identity Provider Partner portal within Oracle Identity Federation/Oracle Access Manager.
- Load the SAML 2.0 Metadata that was exported from your IdP (or enter it manually). For example, Shibboleth interface is in the **idp-metadata.xml** configuration file.
- Configure the *User Mapping* to see the following:
    - Select the appropriate Identity Store that was configured for your Access Management solution. This is used for Single Sign-On.
    - The userid needs to be explicitly identified via the *Universal Resource Name (URN)* in the SAML metadata from the IdP metadata. For example:



Mapping Options

User Mapping

User Identity Store: OAMIDSTORE

User Search Base DN:

○ Map assertion Name ID to User ID Store attribute

Map assertion Name ID to User ID Store attribute: uid

● Map assertion attribute to User ID Store attribute

* Assertion Attribute: urn:oid:0.9.2342.19200300.1(

* User ID Store Attribute: uid

Figure 6: Mapping Options

-
    - The above attribute must be mapped to the **uid** attribute in Oracle Access Manager
    - Optionally, set Enable global logout and HTTP POST SSO Response Binding for SSO integration.
    - Within Oracle Access Manager, use the Create Authentication Scheme and Module option to generate and implement the definitions. If necessary, this generated configuration can be viewed or altered using **Access Manager → Authentication Schemes and Plug-ins → Authentication Module**s respectively.

For the Shibboleth interface example, see Oracle Interoperability for additional detailed instructions.

15.4.5 Enable Just In Time Provisioning in Identity Federation

The Oracle Access Manager authenticator requires that the SAML assertion from the IdP exists in the Oracle Access Manager Identity Store.

When a new user logs into the solution for the first time, the user exists in the IdP and ORMB, but it also needs to exist in the Identity Store to complete the login. As the user already is in the IdP and user object, it is recommended to provision the user in the Identity Store automatically (also known as *Just In Time Provisioning*) upon first login.

This is achieved by setting the userprovisionenabled configuration setting to true within Oracle Access Manager. See JIT User Provisioning in OIF/SP and Oracle Access Manager Administration for more information.

Configure the credentials for access by setting the Bind DN to the connection group and the Identity Store on the *User Identity Stores* as per the Registering and Managing User Identity Stores.

15.4.6 Define WebGate Agent

Within Oracle Access Management, the OAM Agent for ORMB must be registered and configured with the following settings at a minimum:

- Set the Security to *Simple* (or *Cert* for two-way certificate implementations.).
- Set *Auto Create Policies* to create default access policies.
- Set the *Base URL* to the Oracle HTTP Server/Webgate used for the product.
- Set the *Access Client Password* to the password for the Webgate.
- Set the *Relative URI* to the WEB_CONTEXT_ROOT setting in the ENVIRON.INI (prefixed with /) in the *Protected Resource List*. For example, /spl.
- It is recommended to set the *User Defined Parameter* authorizationResultCacheTimeout to 0 to avoid *Invalid SAML Assertion* errors.

See Introduction to Agents and Registration for more information.

15.4.7 Copy WebGate Agent Configuration to OHS/WebGate

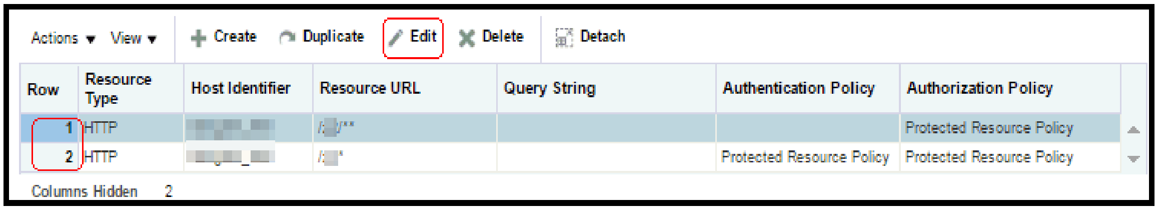
Ensure that the configuration in the previous step is available to the Oracle HTTP Server/Web Gate configuration is transferred to the config directory as per the Registering an OAM Agent Using the Console.

15.4.8 Define Authentication Policy for the Product Domain

To link the Oracle Access Manager and the IdP, an *Authentication Policy* must be configured to connect the WebGate to the IdP for communications. The following process is recommended at a minimum:

- An *Application Domain* is automatically created by the WebGate configuration implementation. This will be used by the process.
- Create an *Authentication Policy* for the selected domain with the authentication scheme appropriate for the IdP.

- Set the *Resources* to the protocol and resource URLs (set to the WEB_CONTEXT_ROOT setting in the ENVIRON.INI, prefixed with / and suffixed with *, for example /ouaf*) as a *Protected Resource Policy*.
- Repeat the above step for subdirectories under the context, for example, /ouaf/**.



Row	Resource Type	Host Identifier	Resource URL	Query String	Authentication Policy	Authorization Policy
1	HTTP	[redacted]	/[redacted]/**			Protected Resource Policy
2	HTTP	[redacted]	/[redacted]*		Protected Resource Policy	Protected Resource Policy

Columns Hidden 2

Figure 7: Authentication Policy for Product Domain

- Set the *Operations* to **All** and ensure the Authentication Policy is set correctly for the IdP for each policy.

15.4.9 Export the OAM SAML Metadata (optional)

If the IdP requires to understand the format of the messages from the Oracle Access Manager SAML request and format the response it is recommended to export the definition using the Export SAML 2.0 Metadata option on the provider. See Managing Settings for Identity Federation for more information.

See the documentation of the IdP for how to import the settings into that product. For the example Shibboleth interface, this is not required as this is automatically performed as part of the handshaking process.

15.4.10 Configure the Product Identity Asserter and Authenticators

The security realm in the product Oracle WebLogic domain must be set to do the following:

- Configure the Oracle Access Manager Identity Asserter to provide the identity from the transaction flow to the product.
- Configure the Custom Authentication Service Provider that is available with the product to participate in the authentication process.
- If Oracle Unified Directory is used, as a supplemental security repository, then its Authentication Adapter must be configured to participate in the authentication process.
- If the Default Authenticator is used, as a supplemental security repository, then it must be configured to participate in the authentication process. This Authenticator is used by the administration consoles to administrate the product.

15.4.10.1 Oracle Access Manager Identity Asserter

For Oracle Access Manager to participate in the authentication the OAMIdentityAsserter must be added to your security realm as an Authentication Provider as per the Implementing the Security Provider. The following should be setup for federated security:

- Set the *Type* to OAMIdentityAsserter.
- Set the *Control Flag* to SUFFICIENT. This tells the security system that if the user existence is confirmed by this adapter then it is a valid user.

- Set the *Active Types* to OAM_REMOTE_USER and OAM_IDENTITY_ASSERTION. This denotes the login types used by federation.

15.4.10.2 Custom Authenticator

Note: Use of the Custom Authenticator Service Provider is optional and only recommended for sites with more than one security repository that requires adjudication.

Oracle Revenue Management and Billing ships a Custom Authentication Service Provider, which checks that the user is defined in the application and that it is still active. Set the *Control Flag* for this Authenticator Service Provider to SUFFICIENT. See Implementing the Security Provider for more details of this adapter.

If you are using Unified Authenticator and/or the Default Authenticator for system and administration account then the following should be configured in the Plugin Properties:

- Set the userGroup property to the user group used for the subsetting the users accessing this plugin.
- Set the excludeUsers property to the comma separated list of system and administration users that are not checked by this plugin. This assumes they will be checked by other authentication providers.
- Optionally, set the debug property to true for non-production debugging of your configuration to send debug log messages to the Web Server logs.

15.4.10.3 Oracle Unified Authenticator (Optional)

Note: Oracle Unified Directory is used in this example but any LDAP based security repository can be used instead using the LDAP Adapter.

If you have administration users that are not defined to your IdP but need to administrate the domain then they should either be defined in the Default Authenticator or an Oracle Unified Authentication Provider. For the latter, see Configuring An Authentication Provider for Oracle Unified Directory. Set the *Control Flag* for this Authenticator Service Provider to SUFFICIENT.

15.4.10.4 Default Authenticator

Typically, implementations would continue to use the inbuilt Default LDAP server (DefaultAuthenticator) for the user identities used by the product to start and stop each component (for example, the system or weblogic account). It is highly recommended that these types of accounts need to be defined in at least one security repository defined in the security realm. Set the Control Flag for this Authenticator Service Provider to SUFFICIENT.

15.4.10.5 Provider Ordering

The most important aspect of configuring multiple authentication providers is to order them in the right order to optimize the login process. Reordering can be performed as outlined in Changing the Order of Authentication Providers with the following guidelines:

- It is highly recommended that Oracle Access Manager Identity Asserter and Custom Authenticator be placed first and second respectively. This will ensure the bulk of the users are authenticated efficiently.

- If the Oracle Unified Authenticator (optional) is used, then it can be placed in the third place to catch administration accounts. It can also be placed in second place, if it is required, but if that is before the Custom Authenticator then the group used for authentication, for example cisusers, must also be defined in Oracle Unified Directory.
- It is recommended that the Default Authenticator be placed last as it is typically only used for the accounts that start and stop the product.

15.4.11 Configure CLIENT-CERT

The last step in the online federation process is to configure the login method used by ORMB. For federation, it is highly recommended to set the login method to CLIENT-CERT which tells the product domain that authentication is coming from an external source. Failure to set this correctly will result in the login screen to be displayed upon connection, which is not desirable in a Single Sign-On based solution.

See the *Oracle Revenue Management and Billing Server Administration Guide* to set this value.

15.5 Federated Web Services

Product based SOAP and REST Web Services are secured and federated using Oracle Web Services Manager (OWSM).

Note: The product supports OAuth 2.0 and uses the "2-legged" Authorization model. See Understanding OAuth2 with Oracle Web Services Manager and Oracle Access Management OAuth Service for more details.

15.5.1 Overview

The architecture of the Web Services federation is similar in nature to the online federation with the following important differences:

- Oracle HTTP Server and WebGate are used in a similar role but delegate, via exclusion, to Oracle Web Services Manager rather than to the IdP and Oracle Access Manager.
- The architecture has an authorization server to determine the validity of the user. An authorized user is issued an access token that is used for transaction security. In this document, Oracle Identity Federation is used but any OAuth 2.0 compliant authorization server can be used.
- Inbound Web Services and REST use the Oracle Web Service Manager integration to implement OAuth2 with support inline and at the container level.

15.5.2 Process Flow

The flow of the Web Services transaction is as follows:

- The Web Service call is redirected to the Authorization Server to be granted access to the requested resources. Optionally, depending on the policy used access certificates can be used.
- If the Web Service identity is valid on the Authorization Server an Access Token is issued that is used for relevant calls for the Web Services.
- Trust is established between the Authorization Server and Product Server to ensure proper security protocols are maintained between the servers. For example:

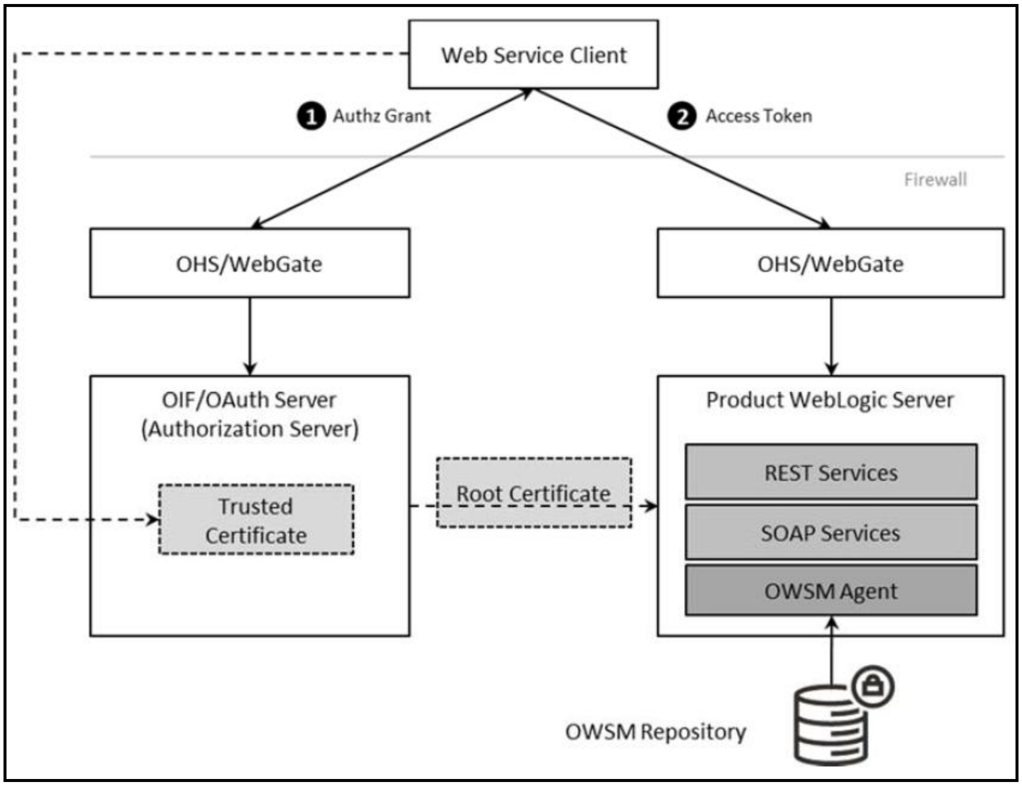


Figure 8: Federated Web Services Process Flow

15.5.3 Setup OAuth Service

The first step in the configuration process is to setup the OAuth component for Oracle Identity Federation in the Authorization Server. See Managing the Oracle Access Management OAuth Service for instructions on this process.

15.5.4 Configure WebGate for SOAP/REST Communications

The online WebGate configuration needs to be altered with the following changes for securing SOAP and REST based services:

- Alter the Application domain within the Authorization Server to add the SOAP and REST URLs as separate resources.

Protocol	Resource URL
SOAP	/<context>/webservices/*
REST	/<context>/rest/ouaf/**

Where <context> is the value in WEB_CONTEXT_ROOT variable configured in the ENVIRON.INI.

- Ensure the *Protection Level* is set to Excluded to delegate the security to Oracle Web Services Manager for these resource URL's. This will disable the *Authentication Policy* and *Authorization Policy* for the resource URL's above.

15.5.5 Create OAuth Client

In the online federation, an identity domain was created for use with online users. For integration with Web Services, a new identity domain must be created and configured specifically for Web Services. See Configuring OAuth Services Settings for detailed information on how to create a new domain and generate a new client to interface to the domain.

When creating the new domain, the following recommendations apply to the configuration:

- Ensure the *Refresh Token Enabled* is set to true.
- Ensure *Lifecycle Enabled* is set to true.
- Ensure the *endpoints* are unique across domains. For example, ouafoauthservice.
- Set the *Attributes* to match your site settings:

Attribute	Comments
jwt.CryptoScheme	To a valid scheme used in your implementation. For example, RS256.
createdByDefault	Set to true.
jwt.cert.alias	Set to oauthkey. This alias will be added later to the keystore.

- In the OAuth Web Service Client for the above domain, specify the default identity, in the *Client ID*, to be used for Web Services. This user must be defined in the ORMB application for authorization. Generate the *Client Secret* for the identity. Ensure *Shown in Clear Text* is set for the identity. Under *Grant Types* under *Privileges*, ensure Client Credentials is enabled to make sure that the token is generated correctly for the Web Services while using a JWT token . See Understanding OAuth2 with Oracle Web Services Manager for additional advice.

15.5.6 Using Keystores and Credentials

As keystores and credentials are typically used, they must match each component of the architecture so that communication can proceed. This subsection will outline the steps to generate, import and configure the various identities in keystores to provide secure communications.

15.5.6.1 Setup Oracle Web Service Manager Client

The Oracle Web Services Manager must define and generate keys to be used in the architecture:

- **Define the OWSM Credential to the Credential Store Framework (CSF)** - Add the Client identifier and the Secret generated in the OAuth Web Service Client, using the WLST command below or using Fusion Middleware Control console:

```
createCred('oracle.wsm.security', key=<key>', user='<user>', password='<secret>',
desc='<description>')
```

  Where:

Command Line Option	Description
<key>	The key used for the credential. This must match either

Command Line Option	Description
	oauth2.client.csf.key or the override specified in the F1-OAUTH Extended Lookup. For example, f1.oauth2.client.credentials.
<user>	The product user specified on the OAuth Web Service Client
<secret>	The value generated for secret on the OAuth Web Service Client
<description>	A short description for the key. This can assist in finding the key in the domain.

- **Generate the Client JWT Bearer Key Pair** - Using keytool generate the orakey key pair and certificate into a temporary JKS keystore. For example:

```
keytool -genkeypair -noprompt -keyalg RSA -keystore /tmp/orakey-keystore.jks -
storepass '<pw>' -alias orakey -keypass '<pw>' -validity 7200 -dname "<cn>"
```

Where:

Command Line Option	Description
<pw>	The password to use for the store. These values will be used in the OPSS Keystore Service (KSS) once imported.
<cn>	CN String for certificate

Points to Note:

This example is for non-production only. Alter the command for alternative certificates.

In non-production, the WebLogic demo certificate can be used. It is recommended to use a valid third-party certificate for production. See Using Third-Party CA Signed Certificates for more information.

- **Define the OPSS Keystore Service (KSS) and Import Keys** – Using Fusion Middleware Control or WLST create the KSS and import the orakey pair. For example:

```
svc = getOpssService(name='KeyStoreService')
svc.createKeyStore(appStripe='owsm', name='keystore', password='<ksspw>')
svc.importKeyStore(appStripe='owsm', name='keystore', password='<pw>',
aliases='orakey', keypasswords='<pw>', type='JKS', permission=false,
filepath='/tmp/orakey-keystore.jks')
```

Where:

Command Line Option	Description
<ksspw>	Keystore Service Password
<pw>	The password used for generated file to be imported.

- Export the certificate for use on the Authorization Server using the following WLST command. For example:

```
svc.exportKeyStoreCertificate(appStripe='owsm', name='keystore', password='',
alias='orakey', keypassword='<ksspw>', type='Certificate', filepath='<file>')
```

Where:

Command Line Option	Description
<ksspw>	Keystore Service Password
<file>	Fully qualified directory and name for exported file. For example: /tmp/orakey.cert.

15.5.6.2 Setup Authorization Server

On the OAuth Server the following must be performed:

- **Import the Certificate** – Import the orakey certificate from the client into the OAuth server using the keytool command. For example:

```
cd $DOMAIN_HOME/config/fmwconfig
keytool -importcert -file <file> -trustcacerts -alias orakey -keystore default-keystore.jks
```

Where:

Command Line Option	Description
<file>	Fully qualified directory and name for file to be imported. For example: /tmp/orakey.cert.

- **Generate the JKS Key Pair** – Generate the JKS Key Pair to ensure that the tokens used are signed appropriately using the keytool utility. For example:

```
keytool -genkeypair -noprompt -keyalg RSA -sigalg SHA1withRSA -keystore default-keystore.jks -storepass '<store_password>' -alias oauthkey -keypass '<key_password>' -validity 3600 -dname "<cn>"
```

Where:

Command Line Option	Description
<store_password>	The password for the key store
<key_password>	The password assigned to the key
<cn>	CN String for key

- **Update OAuth CSF Credentials** – The CSF credentials within Oracle Web Services Manager must be updated to include the private key using the Fusion Middleware Control or a set of WLST commands. For example:

```
updateCred(map="oracle.wsm.security", key="keystore-csf-key", user="owsm", password="<store_password>", desc="Keystore key")
updateCred(map="oracle.wsm.security", key="sign-csf-key", user="oauthkey", password="<key_password>", desc="Signing key")
updateCred(map="oracle.wsm.security", key="enc-csf-key", user="oauthkey", password="<key_password>", desc="Encryption key")
```

Where:

Command Line Option	Description
<store_password>	The password for the key store
<key_password>	The password assigned to the key

- **Export Certificate for use in product** – The final step within the Authorization Server is to export the certificate for use in the product using the keytool utility. For example:

  ```
  keytool -exportcert -keystore default-keystore.jks -storetype JKS -storepass
  <store_password> -alias oauthkey -file <file>
  ```

  Where:

Command Line Option	Description
<store_password>	The password for the key store
<file>	Fully qualified directory and name for export file

15.5.6.3 Import Certificate into Product

On the product server the certificate and trust from the Authorization Server must be imported so that a trust is established between the servers. To achieve this the following steps are recommended to be performed on the product application server:

- Transfer the exported certificate file from the Authorization Server.
- Convert the format of the certificate file into PEM format. For example:

  ```
  openssl x509 -inform der -in <file> -out <pemfile>
  ```

  Where:

Command Line Option	Description
<file>	Fully qualified directory and name for exported certificate file
<pemfile>	Fully qualified directory and name for generated PEM file

- Import the PEM certificate into the OPSS keystore used by the product using Fusion Middleware Control or WLST commands. For example:

  ```
  svc = getOpssService(name='KeyStoreService')
  svc.importKeyStoreCertificate(appStripe='owsm', name='keystore', password='',
  alias='oauthkey', keypassword='', type='TrustedCertificate', filepath='<file>')
  ```

  Where:

Command Line Option	Description
<file>	Fully qualified directory and name for PEM file

15.5.7 Enable OAuth on Product

To use OAuth within the product the following is recommended to be configured:

- Using the configureEnv.sh –a command set the following settings:

Attribute	Recommended Setting
CSRF Protection for REST Services	false
OWSM Protection for REST Services	true

Note: Execute the initialSetup.sh utility to reflect the changes.

- The following Oracle Web Service Management Policies must be attached with Oracle Web Services Manager or within the product configuration for both REST and SOAP:

Protocol	Oracle Web Services Manager Policy
SSL	oracle/multi_token_over_ssl_rest_service_policy
Non-SSL	oracle/multi_token_rest_service_policy

15.5.8 Use Oracle Web Service Manager Policies

The above Web Services policies must be configured within the product to configure OAuth2 support. This may be done at the individual service level (when part of your solution is federated) or globally for all services.

15.5.8.1 Individual Web Service Policies

To attach the token policies to individual Inbound Web Services the following process is recommended to be performed:

- If it does not already exist, create a Web Service Annotation of type *Annotation for OWS Security Policy*. In the uri *Parameter Name*, specify the appropriate policy (SSL or non-SSL outlined in the previous section.
- Attach the Web Service Annotation to the relevant services.
- Deploy/Redeploy the Inbound Web Services to implement the policy. See the *Oracle Revenue Management and Billing Server Administration Guide* for more information.

15.5.8.2 Global Web Service Policies

Note: It is possible to override the global setting on individual Inbound Web Services by specifying additional annotations.

If security policies are to apply globally across all services, then the following process is recommended:

- Create or alter the F1_EMAILCFG *Feature* of Feature Type *External Messages* and specify the *Default Security Type* option with the relevant security policy.
- Deploy/Redeploy the Inbound Web Services to implement the policies across services. See the *Oracle Revenue Management and Billing Server Administration Guide* for more information.

15.6 Federated Outbound Messages

Note: The Authorization Server setup and Client Setup are identical to the configuration of Inbound Web Services. See the Process Flow section for initial setup instructions.

Outbound Messages allows product transactions to send information out of the product via technology connectors in synchronous and asynchronous modes.

15.6.1 Overview

As with the Inbound Web Services the architecture of the Federated security involves issuing a token from an authorized user from the Authorization Server to the External System:

- An Outbound Message is created by the business process of a certain type and with the relevant payload.
- An External System definition with the product decides how this information is sent to the External System (the technology and mode).
- A Message Sender is configured to use the relevant policy in its context parameters to send to the external system. The policies supported are configured on an Extended Lookup to filter the policies available.
- When sending the information out, the Authorization Server issues a token to an authorized transaction to use in the transaction to the external system.
- Trust is established between the Authorization Server and Resource Server for the external system to accept valid transactions. This certificate must be exported from the Authorization Server and imported into the technology used by the Resource Server.

For example:

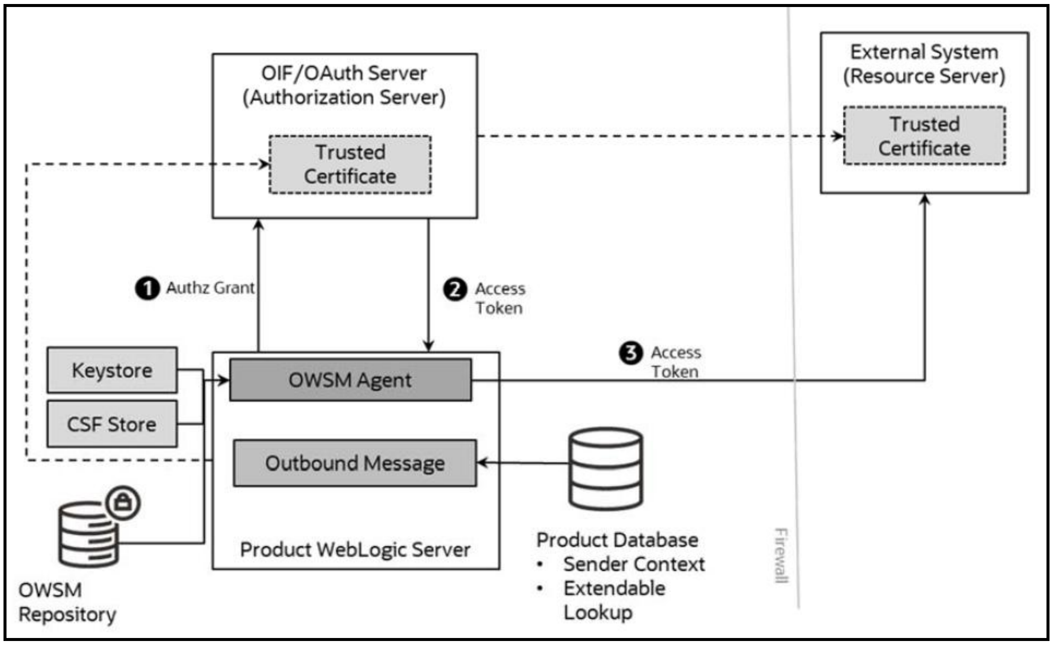


Figure 9: Federated Outbound Messages

15.6.2 OAuth Policies

To use this facility the following policies are recommended for use from Oracle Web Service Manager and the product:

- oracle/oauth2_config_client_policy
- oracle/http_oauth2_token_client_policy

These two policies are compatible with the oracle/multi_token_rest_service_policy used for Inbound Web Services (REST and SOAP).

15.6.3 Extendable Lookup Configuration

Note: These values are shipped with the product metadata and the policy configuration values set for the policy used should see the *Oracle Web Services Manager* documentation for a description of the valid values.

The following Extended Lookups are provided to be used:

Extended Lookup	Recommendations
F1-ValidPolicies	Two policies exist (F1-OWSM-CLIENT and F1-OWSM-TOKEN). These are delivered with the parameter settings.
F1-SetOfPolicies	This Extended Lookup is altered to set the parameter values for the valid policies above as a Policy Set. An extended lookup value is recommended to be added for each external system interfaced.

The following recommendations apply to the configuration of the above policies:

- For any CSF key parameters, the keys need to be added to the CSF as outlined in the Setup Oracle Web Service Manager Client.
- The URI parameters may be hardcoded or use substitution variables as outlined in the *Oracle Revenue Management and Billing Server Administration Guide*. If substitution variables are used, they should be configured in the substitutionVariable.xml configuration file. For example:

```
<uriVariable>
    <name>F1_TOKEN_URI</name>
    <value>http://<server>:<port>/ms_oauth/oauth2/endpoints/oauthservice/tokens</value>
</uriVariable>
```

- Additional parameters may be set according the documentation for the Client and Token.

15.6.4 Message Sender Configuration

The final step in the configuration of the use of federation for Outbound Messages is to configure the context of the Message Sender to use the Policy Set that was configured in the earlier step. To use the federation the following content types must be set:

Context Type	Recommendations
Sender Security Type	This must be set to OWSM.
OWSM Policy Set	Set to the Policy Set configured (e.g., F1-OAUTH)

16. Securing JNDI Access

By default, the JNDI used for the Oracle WebLogic Domain is open to read access by any valid user in the domain. This behavior may not be appropriate as dictated by the security policies at your site. It is possible to set the domain permissions, after installation, to minimize access to the JNDI.

Note: The instructions in this section use the JNDI facilities described in the Access policies for JNDI resources section and Resource Types You Secure Using Roles and Policies of the Oracle WebLogic documentation. See that section before configuring the security of the JNDI.

This section contains the following topics:

- Securing Product Access
- Providing Additional Access to the JNDI

16.1 Securing Product Access

The JNDI registers all the resources used in the Oracle WebLogic domain for the product. For the product to operate the following is recommended:

- Administration Users should be part of an Admin role. Additional roles are supplied with Oracle WebLogic.
- Product users are in group designated by the WEB_PRINCIPAL_NAME and WEB_ROLE_NAME settings in the ENVIRON.INI.
- Optionally, it is possible to create additional groups in your security repository to allocate specific permissions. This is outlined in Providing Additional Access to the JNDI.
- View the JNDI tree for the product servers/clusters in the console and assign the following policies:

Resource	Role/Group
Server Resources	Allocate to Admin role or individual administration accounts. This is required to start/stop and maintain the JNDI resources for the server.
JMX Resources	Allocate to Admin role or individual administration accounts. This is required to monitor the server from the console, Fusion Middleware Control and/or Oracle Enterprise Manager using the JMX interface. If the Oracle Management Pack is for Oracle Revenue Management and Billing, this may also need to allocate to the Product group/role or individual users if the credentials used for the connection are not associated with any users in the Admin group.
JDBC Resources	Allocated to both Admin role/individual administration users and Product group/role to allow access to JDBC connection pools.
EJB Resources	Allocate to Product group/role to allow access to Business Application Server.
JMS Resources	Allocate to Product group/role to allow access for MDB, Outbound Message via JMS or JMS Real Time Adapter access.

- It is also recommended to set the weblogic.jdbc.remoteEnabled to false in the JAVA_OPTIONS and WLS_JDBC_REMOTE_ENABLED variables in the setDomainEnv.sh utility provided with Oracle WebLogic or by ese. For example:

  -Dweblogic.jdbc.remoteEnabled=false
- If the SSL protocol is used it is recommended to set the RMI JDBC Setting to Secure on the Product Server/Cluster Advanced Settings.
- Save the JNDI changes

16.2 Providing Additional Access to the JNDI

If third-party access is required for access to the JNDI of the product, then the following is recommended:

- Setup the role or group in your security repository. In a configuration where multiple security repositories exist, this identity should only exist in one repository.
- Allocate the relevant permission as outlined in the Access Policies for JNDI Resources section and Resource Types You Secure Using Roles and Policies of the Oracle WebLogic documentation.
- The configuration of this additional access should be appended to the existing configuration.

17. Object Erasure Support

To support data privacy concerns, the product now supports erasure functionality for master objects. This allows for removal of Personally Identifiable Information (PII) from the product whilst respecting business rules.

Note: This capability is restricted to be used with master data. Transaction data is supported for erasure using the Information Lifecycle Management (ILM).

17.1 Configuration Of Object Erasure

The Object Erasure functionality provides an ability to define the following for erasure:

- An Object Erasure Maintenance Object (F1-OBJERSCH) that can be used to map the reassure of the object. This maintenance object should be used as a basis for the Business Object to describe the storage of the Object Erasure information for individual objects.
- A maintenance object algorithm to Maintain Object Erasure Schedule to define the rules and retention for the object including any obfuscation rules.
- A generic Batch Control (F1-OESMN) that can be used to implement the erasure or obfuscation rules in batch.

See **The Approach to Implementing Object Erasure** section of the online documentation for details of the process for configuring Object Erasure.

18. Key Ring Support

Cryptography keys may be used to provide a signature or credentials to a request so that the system recognizes that the request comes from a trusted party. Keys may also be used to encrypt or decrypt files shared between two parties.

The Key Ring object is provided to reference the keys that are used over time for a given business use case. Only one key or key pair may be active at any given time. This section includes information about the functionality provided to support different key ring classes for particular use cases:

- Signature Keys
- File Signing Keys
- OAuth Keys

This section contains the following topics:

- Maintaining Key Rings
- Using Key Rings

18.1 Maintaining Key Rings

For information about maintaining key rings, see the *Oracle Utilities Application Framework Administrative Guide*.

18.1.1 Generating Keys

Once the Key Ring is defined, it must have at least one activated key pair. To generate a key pair, use the **Generate Key** button.

Once generated, the key ring will appear in the **Key Pairs** zone with the appropriate fingerprint. To activate the key pair, use the **Activate** button to enable the key. It is recommended to only have one pair active for each key ring at most at any time. It is possible to support multiple, but this is not good security practice. Use the **View** under the **Public Key** column to view and pass on the public part of the key.

Note: The private key is not visible from the product in order to be in line with the security standards.

18.2 Using Key Rings

Key rings can be used within numerous objects within the ORMB application. See the documentation for those objects on how to connect key rings. Once connected, the object will appear in the **Key Ring References** zone.

19. Redaction Rules

Oracle Revenue Management and Billing supports configurable redaction rules that allow exports using Content Migration Assistant (CMA) and Generalized Data Export (GDE) to scramble information as necessary for privacy purposes. This capability is not used outside of Content Migration Assistant and Generalized Data Export.

For information about configuring and using redaction rules, see the *Oracle Utilities Application Framework Administrative Guide*.

20. Cookies Used by ORMB

ORMB uses several cookies for managing behavior of the service. All cookies except ORA_OUAF_Login_language are session cookies that are stored in memory and are removed when the browser or session is closed.

20.1 Cookies Used by Network

The following table provides information that may assist you while configuring cookie controls at the network level.

Cookie	Usage
ORA_OUAF_Login_language	This is a persistent cookie that is used to store the user's preferred language for the provided login screen. For more information, see the *Question: Does Weblogic CCB Cookies Contain Any Trace Information? (Document ID: 2443627.1)* article *on* My Oracle Support.
JSESSIONID	The JSESSIONID cookie, which expires when the user's browsing session ends, helps the server to manage user sessions. It is a standard container cookie. While not accessible to scripts, this cookie can be deleted from the client-side. However, the cookie will be re-sent during the next request from the user. This cookie tracks each request from the same browser, ensuring that the same session data is available on the server side. It does not contain any personal data. This cookie can be secured using CSPFilter provided with the solution and can be further protected from manipulation using the `com.oracle.ouaf.web.disableJessionIdSecureCookieHandling` setting. By default, the `com.oracle.ouaf.web.disableJessionIdSecureCookieHandling` is set to false. Depending on the Oracle WebLogic configuration, an additional cookie `WL_AUTHCOOKIE_JSESSIONID` is also created.
ORA_OUAF_Language	This is an internal cookie that stores the active language used by the browser. This may change if the user switches languages during the session. The cookie is used by the server to render the screen using the language correctly.

Note: All cookies are secured using the `SameSite` attribute in the `webLogic.xml` file. This is controlled by the `com.oracle.ouaf.web.allowCORS` configuration setting.

20.2 Cookies Used by Cloud Services

The following table lists and describes additional cookies used by Oracle Revenue Management and Billing Cloud Services.

Cookie	Usage
ORA_OUAF_Language_Dir	This is an internal cookie that stores the active language direction used by the browser. This may change if the user switches languages during the session. The cookie is used by the server to render the screen using the language correctly. By default, it is set to ltr for most implementations and rtl for Arabic implementation.
ORA_OUAF_Locale_Info	This is an internal cookie that stores the locale information used for sorting. This may change if the user switches languages during the session.
ORA_OUAF_SESSION_EXP, ORA_OUAF_SERVER_TIME, and ORA_OUAF_CLIENT_TM_OFFSET	This is a set of internal cookies that are used to determine inactivity timeouts for user sessions. It is used by the server to inform the user of the session timeout.
ORA_OUAF_START_LOGIN	This is an internal cookie used by the login screen for the session timeout. Its value is determined from the server configuration.

21. Garbling Customer Information

Oracle Revenue Management and Billing enables you to garble the personal information of a customer, such as the customer name, address, email ID, social security number or any other identifier, birth date, gender, and so on. On receiving the data erasure request from a customer, the data controller will garble the personal information of the customer by masking the data with random characters and then cease the access to the customer records.

To garble the information of one or more persons, you need to do the following:

1. Define the **Garble Data (C1-GRBL)** characteristic for each person whose data needs to be garbled and set its value to **Y**.
2. Execute the **Garble Person Information (C1-GRBPI)** batch. For more information about this batch, see *Oracle Revenue Management and Billing Batch Guide*.

On executing the **Garble Person Information (C1-GRBPI)** batch, the system identifies the persons where the **Garble Data (C1-GRBL)** characteristic is set to **Y** and the **Garbling Completed (C1-GBCM)** characteristic is either not defined or set to **N**. Note that the system considers the **Garble Data (C1-GRBL)** characteristic value which is effective on the batch business date.

Once the persons are identified, the data in the following fields is masked with random characters across all records in the system:

- ACCOUNT_NBR
- ACCT_NBR
- ADDRESS1
- ADDRESS1_UPR
- ADDRESS2
- ADDRESS3
- ADDRESS4
- BIRTH_DT
- BROKER_NAME
- CITY
- COUNTRY
- COUNTY
- CUSTOMER_NAME
- EMAILID
- ENTITY_NAME
- ENTITY_NAME1
- ENTITY_NAME2
- ENTITY_NAME3
- GEO_CODE
- HOUSE_TYPE

- IN_CITY_LIMIT
- NAME1
- OVRD_MAIL_NAME1
- OVRD_MAIL_NAME2
- OVRD_MAIL_NAME3
- PER_ID_NBR
- PHONE
- POSTAL
- POSTAL_UPR
- STATE

In other words, the personal and account related information of a customer is garbled across the system. In addition, this batch allows you to garble the required characteristics and identifiers of a person and account.

Note: An account may have one or more persons linked to it. The information of an account is garbled only when the data erasure request is received for the main customer of the account.

Once the personal and account related information of a customer is garbled, the access group of the person is changed to **C1GB** (which does not have any data access role). This ensures that the data of the person and his or her accounts cannot be accessed from any screens in the system. If an approval transaction contains personal and account related information of a customer, you cannot even view the approval transaction in the system.

Note: The access is forbidden for performing any transaction for the person and account only from the user interface and web services and not through the background process.

You can also garble the data of a person using the **C1-GRBPI** inbound web service. However, you can garble the data of only one person at a time using this inbound web service.